Measuring Cybercrime

Bijmans, H.L.J.

DOI
10.4233/uuid:feb2893c-1a79-4767-84d2-d23b41abc889

Publication date
2025

Document Version
Final published version

Citation (APA)
Bijmans, H. L. J. (2025). *Measuring Cybercrime*. [Dissertation (TU Delft), Delft University of Technology]. TU Delft OPEN Publishing. https://doi.org/10.4233/uuid:feb2893c-1a79-4767-84d2-d23b41abc889

MEASURING CYBERCRIME

Hugo L.J. Bijmans

MEASURING CYBERCRIME

MEASURING CYBERCRIME

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
30 september 2025 om 15:00 uur

door

Hugo L.J. BIJMANS

Master of Science in Computer Science, Delft University of Technology
geboren te Voorburg, Nederland.

Dit proefschrift is goedgekeurd door de promotores:

Prof. dr. M.J.G. van Eeten
Dr. R.S. van Wegberg

Samenstelling promotiecommissie:

Rector Magnificus	Voorzitter
Prof. dr. M.J.G. van Eeten	Delft University of Technology, promotor
Dr. R.S. van Wegberg	Delft University of Technology, copromotor
Onafhankelijke leden:	
Prof. dr. M.E. Warnier	Delft University of Technology
Prof. dr. G. Smaragdakis	Delft University of Technology
Prof. dr. A. Hutchings	University of Cambridge
Prof. dr. N. Christin	Carnegie Mellon University
Dr. L. Allodi	Eindhoven University of Technology

Dit onderzoek is mede mogelijk gemaakt door de Nederlandse Organisatie voor toegepast-natuurwetenschappelijk onderzoek, TNO.

Trefwoorden: Cybercrime, Internet measurements, Cryptojacking, Phishing, Law Enforcement

Drukwerk: Ridderprint

Omslag: Joey Roberts

Een digitale versie van dit proefschrift is beschikbaar via
http://repository.tudelft.nl/

Komt goed.

CONTENTS

ACRONYMS

APWG Anti-Phishing Working Group. 105, 106, 109–111, 155, 160, 175

AS Autonomous System. 63, 65, 100, 117, 118, 132, 135, 136

BPH Bulletproof hosting. 117, 118

C&C Command and Control. xiv, xviii, 2, 10, 77, 114, 127, 133

CDF Cumulative Distribution Function. 74, 103

CMS Content Management System. 23, 24, 35, 48, 55

CPU Central Processing Unit. 18, 20, 23, 25, 33, 34, 52, 78

CSAM Child Sexual Abuse Material. 115, 116, 127, 128, 133, 134, 137–139, 175, 179, 180

CSS Cascading Style Sheets. 93, 105

CT Certificate Transparency. 158

DDoS Distributed Denial of Service. 5, 64, 127, 128, 133, 142, 144, 146, 149–153, 175, 179

DNS Domain Name System. 3, 41, 48, 149, 151, 153, 156, 159, 161, 176, 181

FQDN Fully Qualified Domain Name. 97, 98, 100, 101, 104, 105

HTML Hypertext Markup Language. 25, 29, 55, 89, 91

HTTP Hypertext Transfer Protocol. 20, 52, 55, 57, 59, 61, 69–71, 79, 81, 87, 109, 150, 158, 178

HTTPS Hypertext Transfer Protocol Secure. 5, 57, 87, 90, 109, 158, 160

IP Internet Protocol. 3, 27, 32, 41, 48, 59, 61–66, 69, 70, 75, 77, 81, 90, 91, 94, 96, 100, 105, 110, 117, 118, 120, 121, 123, 125, 131, 132, 135, 136, 138, 149–151, 156, 158–161, 164, 165, 174

ISP Internet Service Provider. xv, 13, 62–64, 69, 75, 150, 151, 176

LEA Law Enforcement Agency. xvi, 15, 84, 101, 110–112, 141–144, 153, 165–167, 169–171, 175–177, 182, 183

MITM Man-in-the-middle (attack). xiv, xviii, 10, 48, 51, 53, 54, 56, 58, 71, 72, 78–80, 109, 156, 158, 174

NTD Notice-and-takedown. 6, 110

PII Personally Identifiable Information. 120, 121, 166

PoC Proof-of-concept. 90

RAT Remote Access Trojan. xvi, xx, 14, 141, 142, 146, 161, 164–166, 169, 170, 176

RDN Registered Domain Name. 97, 100

SSH Secure Socket Shell. 55, 69, 77, 81

TLD Top Level Domain. 19, 27, 42, 43, 45–49, 90, 91, 98–100, 158, 174

TLS Transport Layer Security. xv, xix, 3, 5, 14, 83–85, 87, 90, 94, 96, 100, 103, 105, 110, 111, 158, 174

TTP Tactics, Techniques, and Procedures. xiii, xv, xvii–xx, 1, 6, 7, 14, 62, 83–85, 141, 171, 176, 183

URL Uniform Resource Locator. 3, 5, 27, 39, 57, 96–98, 108, 109, 158, 160, 165, 168

VPS Virtual Private Server. 87, 94, 100, 132, 158

SUMMARY

The introduction of the Internet in the early 1990s enabled emailing, Web surfing, and videoconferencing but also opened doors for criminals to abuse these new opportunities. Scamming Internet users, compromising computer systems, or even robbing online bank accounts are practices that we call *cybercrime* nowadays. In this dissertation, we use the definition of Gordon & Ford [95] that refers to *"any crime that is facilitated or committed using a computer, network, or hardware device"* and as most cybercrime is driven by monetary gain [78, 115], we limit ourselves to crimes committed with monetary motivations and exclude the actions of nation-state actors or hacktivists. We are not the first to study cybercrime, and much is already known about the tactics, techniques, and procedures (TTP) of modern cybercriminals. Yet, the scale of cybercrime is ever-expanding, impacting many innocent citizens and resulting in millions of dollars in losses for the global society [227]. Law enforcement agencies and policymakers worldwide try to disrupt these miscreants but often lack a clear picture of the crimes they are fighting. Scholars have tried to assess the scale of various cybercrimes, either through large-scale Internet measurements [124, 241, 263] or through qualitative criminology research [143, 150, 206]. Estimations of the size of cybercrime, however, rely often on reported incidents, court cases, or police reports [31]. Yet, the majority of cybercrime incidents are never reported to the police [57, 59], which implies a much higher amount of actual (attempted) cybercrime. The question arises as to how much higher this amount would be and how to measure it to assist in the effective governance of cybercrime.

To structure measurements of cybercrime, we leverage the concept of value chains and cybercriminal resources. Cybercriminals need a variety of products and services to set up their attacks [72], just as criminals involved in illegal drug production and trafficking. A major difference between traditional drug crime and cybercrime is how it can be measured. Traditional crimes rely on physical products that are often traded decentralized and outside anybody's view, making it hard to measure their presence [164]. Insights into the dependencies in such a criminal operation are often obtained by police investigations following arrests, confessions from ex-criminals, or through undercover operations. This is different from cybercrime, which can be measured more easily and safer [164], as the observability of resources used in cybercrime and their centralized collection allows for external measurements and does not require undercover police operations. The possibility of performing these kinds of measurements does not imply that it is trivial. The Internet is immense and constantly changing, making it difficult to detect criminal activity next to the many benign Internet activities and requires robust measurement approaches to structure analyses of cybercrime.

As the word *cybercrime* suggests, measurements of profit-driven cybercrime involve two disciplines of academic research: Internet measurements (*cyber*) and criminology (*crime*). Both are aimed at identifying the TTP that cybercriminals employ [239] but approach their measurements in a totally different way [109]. As a result, both approaches

come with their advantages, disadvantages, and challenges. Internet measurements allow for obtaining a broad picture of a type of cybercrime with many data points but come with dataset selection challenges and often lack in-depth insights into how cybercriminals work. Criminologists often present this level of detail, but their findings are narrowly scoped and gathered from smaller datasets. Despite the large body of work produced by both disciplines, their research efforts have largely remained disjoint, with researchers on one side not benefiting from the advancements proposed by the other [109]. We hypothesize that combining both approaches into socio-technical measurements of cybercrime, hereby performing Internet measurements while taking criminology aspects into account, could result in improved measurements. What innovations are there to be made to perform such measurements? How to perform cybercrime measurements to assist in cybercrime governance?

The rise of new and emerging types of cybercrime requires scientific innovations to measure them. In 2019, browser-based cryptojacking gained popularity among cybercriminals as a means to earn a profit from stolen computing power. Several researchers started designing techniques to detect illicit in-browser cryptomining activity [101, 127, 149, 199, 205, 207], yet little attention was paid to determining the scale of this cybercrime (i.e., assessing its prevalence among Internet users) nor investigating large campaigns of cryptomining websites related to a shared actor. Additionally, no research has been conducted into other means of distribution, such as MITM attacks on compromised Internet infrastructure. Next, we found geographic demarcations to be scarce in Internet measurements due to the global character of the modern-day Internet. Although various previous studies hinted at the importance of spatiality in cybercrime measurements [73, 98, 181], we do not find many studies involving Internet measurements of cybercrime focused on the Netherlands. Besides law enforcement and policymakers, there is an important role in the fight against cybercrime reserved for a variety of organizations and mechanisms operating within the *anti-abuse ecosystem*. Within this ecosystem, some parties report the existence of malicious content to the intermediaries or facilitators hosting it. Although abuse handling has been around for many years, little is known about the internal decision-making processes within such intermediary hosting providers to deal with abuse reports. Past studies have predominantly taken an external point of view and analyzed how abusive content – e.g., phishing sites, spamming activity, and malware C&C servers – are allocated across providers [219, 232]. Such studies, however, lack insights into the internal dynamics of the respective companies, which is something we would like to address. Lastly, we wonder how cybercrime measurements assist in law enforcement efforts to disrupt cybercrime. What kind of information related to cybercrime is essential for designing evidence-based governance strategies? Some academics are convinced that their scientific measurements of cybercrime could assist governance [239, 249], yet we found no scientific proof for these claims.

This dissertation studies the challenges of performing socio-technical measurements of profit-driven cybercrime. We investigate how cybercrime can be measured adequately and how insights gained from these measurements can improve governance. Thus, we formulate the following research question: *"How to perform socio-technical measurements of cybercrime to assist in cybercrime governance?"*. A total of five studies, included as chapters in this dissertation, aim to answer this research question.

Chapter 2 focuses on cryptojacking on websites and estimates the prevalence of this type of cybercrime, as illicit cryptomining has skyrocketed since 2017. Two extensive studies are performed to examine Web-based cryptojacking – one focused on organized cryptomining and one on the prevalence of cryptojacking on the entire Internet. We use a combination of data sources to build a representative dataset for our measurements and leverage previous academic work to crawl numerous domains and search for overlapping resources used in cryptojacking campaigns. We identify the commonly used attack vector for cryptojacking, the most prevalent category of websites affected, and stumble upon large cybercriminal campaigns.

Next, in Chapter 3, we dug deeper into cryptojacking attacks by examining installed malicious cryptojacking scripts on compromised Internet infrastructure, such as ISP and consumer-grade routers. Through a firmware vulnerability in MikroTik routers, cybercriminals could rewrite Web traffic flowing through these routers and inject it with malicious cryptojacking code. Following this attack vector revealed a completely new realm of cryptojacking activity, much larger and more organized than any website-based cryptojacking. Based on NetFlows recorded in a Tier 1 network, semiweekly Internet crawls, and network telescope traffic, we examine the modus operandi of these cybercriminals. We show how these large installations of cryptojacking attacks relate to each other and sketch the life cycles of infected infrastructure. Additionally, we perform campaign analysis to identify different levels of sophistication among adversaries.

In Chapter 4, we switched focus to a different type of cybercrime: phishing, which is the illicit harvesting of user credentials. Although many phishing studies have been published [58, 119, 139, 241, 262], an in-depth analysis of the cybercriminal TTP targeted at Dutch citizens was lacking. For this research, we specifically focused on the first component in the phishing value chain: development. This component involves developing and trading phishing kits used in phishing attacks. We collect many such phishing kits, leverage the use of TLS certificates by phishers, and develop a crawler to uncover phishing domains aimed at customers of the Dutch financial sector. This allowed us to capture the end-to-end life cycle of such phishing attacks and investigate the luring capabilities to trick victims into disclosing their credentials. These insights enabled us to paint a comprehensive picture of the TTP prevalent in the Dutch phishing landscape and present public policy takeaways for anti-phishing initiatives.

Chapter 5 scrutinized the instruments that have been put into place to fight Internet abuse, like cybercrime. These instruments range from legislation to remove copyrighted material to technical instruments such as blocklists to stop spam and rely on industry standards for abuse reporting: reporting abuse to the resource or network owners requesting mitigation. Through a unique collaboration with law enforcement in the Netherlands, we were granted access to the operational back-end of a hosting provider with a reputation for abuse. We examine its abuse tickets and their follow-up actions to investigate what mechanisms in the anti-abuse ecosystem influence anti-abuse actions. We find that notification rates highly depend on the source and the type of abuse report and less on the involved client. Governance instruments like blocklisting, de-peering, and law enforcement inquiries that could directly hurt business continuity affect client notifications, whereas individual abuse reporting is easily ignored.

Chapter 6 combined previous chapters by examining past scientific measurements

of cybercrime. These are typically based on large-scale Internet measurements of resources such as domains, binaries, or attack traffic. We leverage the concept of value chains to structure 38 studies featuring measurements of phishing, booter services, and remote access trojans (RATs). We review how other scholars have set up their measurements of these profit-driven cybercrimes, examine their methods, and assess the used data sources to identify common denominators. Then, we let law enforcement professionals reflect on some of these measurements and jointly identify the unexplored potential for novel measurements that could solve current needs in cybercrime policing. Combining the input from law enforcement with the structured analysis of previous academic measurements allowed the identification of what characteristics of scientific measurements are useful for law enforcement – i.e., assisting LEAs in determining how their scarce resources can be best put to use.

Finally, Chapter 7 presents the main findings of our work, reflects on its societal impact and governance implications, and presents avenues for future work. We find that an important and often overlooked aspect of Internet measurements examining cybercrime is that such measurements should aim to measure *crime*. Crime involves individuals (or groups) with a clear (profit-driven) intent and corresponding decision-making processes that balance risk and reward. Such characteristics of crime are well known to criminologists but are not reflected in the measurements performed by computer scientists. Additionally, we find that Internet measurements of cybercrime often lack a clearly defined geographical demarcation, which is important to LEAs as they operate in conjunction with local jurisdictions. In writing this dissertation, it has become apparent that all the different measurement lenses can assist in designing measurements that apply directly to law enforcement needs, albeit in different ways. Value chains are a useful lens to determine what measurements exactly aim to measure, life cycle analysis has the capability to show the (im)possibilities for governance interventions, and performing campaign analysis has the potential to provide valuable insights into the degree of organization within a studied cybercrime. Insights into the degree of organization and the different types of cybercriminals can assist law enforcement in designing effective governance strategies. Lastly, we find a misalignment between the focus of academic cybercrime measurements and law enforcement needs. We found most measurements to focus on the deployment and execution components of cybercrime, whereas LEAs desire to learn more about development and monetization.

In short, cybercrime measurement approaches that consider the intent behind criminal acts, that are clearly delineated in terms of geographical scope and the type of actor studied, and that incorporate campaign analysis to assess not only the prevalence but also the degree of organization, such measurements can provide valuable support to law enforcement efforts in combating cybercrime. Structuring academic analyses around value chains, life cycles, and campaign analysis creates safeguards for researchers to assess whether they measure what they want to measure.

SAMENVATTING

De introductie van het Internet begin jaren negentig maakte e-mailen, surfen op het Web en videoconferenties mogelijk, maar opende tegelijkertijd de deur voor criminelen om deze nieuwe kansen te misbruiken. Het oplichten van Internetgebruikers, het compromitteren van computersystemen of zelfs het leegroven van online bankrekeningen zijn praktijken die men tegenwoordig *cybercriminaliteit* noemt. In dit proefschrift hanteren we de definitie van Gordon & Ford [95], die luidt als volgt: *"elke misdaad die wordt gefaciliteerd of gepleegd met behulp van een computer, netwerk of hardwareapparaat"*. Omdat de meeste cybercriminaliteit wordt gedreven door financieel gewin [78, 115], beperken we ons tot misdrijven met een financiële motivatie en worden hiermee acties van statelijke actoren of hacktivisten uitgesloten. Er is al veel bekend over de tactieken, technieken en procedures (TTP) van moderne cybercriminelen. Toch blijft cybercriminaliteit toenemen, met gevolgen voor vele onschuldige burgers en een verlies van miljoenen dollars wereldwijd [227]. Opsporingsdiensten en beleidsmakers proberen deze misdadigers aan te pakken, maar hebben vaak geen duidelijk beeld van de misdaden die zij bestrijden. Wetenschappers trachten de omvang van verschillende vormen van cybercriminaliteit te schatten, hetzij via grootschalige metingen van het Internet [124, 241, 263], hetzij via kwalitatief criminologisch onderzoek [143, 150, 206]. Schattingen van de omvang van cybercriminaliteit zijn echter vaak gebaseerd op gerapporteerde incidenten, rechtszaken of politierapporten [31]. In werkelijkheid ligt de hoeveelheid cybercriminaliteit waarschijnlijk veel hoger, omdat de meerderheid van incidenten nooit bij de politie wordt gemeld [57, 59]. De vraag rijst hoeveel hoger deze aantallen liggen en hoe deze kunnen worden gemeten om effectieve bestrijding van cybercriminaliteit te faciliteren.

Om metingen van cybercriminaliteit te structureren, wordt in dit proefschrift de concepten waardeketens en cybercriminele middelen gebruikt. Cybercriminelen hebben een verscheidenheid aan producten en diensten nodig om hun aanvallen op te zetten [72], vergelijkbaar met criminelen die betrokken zijn bij de productie en handel van illegale drugs. Een belangrijk verschil tussen traditionele drugsmisdaad en cybercriminaliteit is echter hoe deze gemeten kan worden. Traditionele misdaad draait om fysieke producten die vaak gedecentraliseerd en buiten het zicht worden verhandeld, wat het meten ervan bemoeilijkt [164]. Inzichten in dergelijke criminele operaties worden vaak verkregen via politieonderzoek na arrestaties, bekentenissen van ex-criminelen of middels undercoveroperaties. Cybercriminaliteit kan daarentegen eenvoudiger en veiliger worden gemeten [164], omdat de gebruikte middelen gemakkelijk te observeren zijn en gecentraliseerd verzameld kunnen worden. Dit betekent echter niet dat dergelijke metingen triviaal zijn om uit te voeren. Het Internet is enorm en voortdurend in verandering, wat het moeilijk maakt om criminele activiteiten te onderscheiden van de vele legitieme Internetactiviteiten. Het vereist robuuste meetmethodologieën om analyses van cybercriminaliteit te structureren.

Zoals de term *cybercriminaliteit* suggereert, omvatten metingen van financieel ge-

motiveerde cybercriminaliteit twee academische onderzoeksdisciplines: Internetmetingen (*cyber*) en criminologie (*criminaliteit*). Beide disciplines zijn gericht op het identificeren van de TTP die cybercriminelen hanteren [239], maar benaderen hun metingen op geheel verschillende manieren [109]. Dit brengt zowel voordelen als nadelen met zich mee. Internetmetingen bieden een breed overzicht van een type cybercriminaliteit met veel datapunten, maar de juiste dataset selecteren is uitdagend en er missen vaak diepgaande inzichten in hoe cybercriminelen opereren. Criminologen leveren vaak wel dit detailniveau, maar hun bevindingen zijn veelal gebaseerd op onderzoek met een kleinere reikwijdte en dus op kleinere datasets. Ondanks de grote hoeveelheid onderzoek binnen beide disciplines, blijven hun onderzoeksinspanningen grotendeels gescheiden, waardoor onderzoekers van de ene discipline niet profiteren van de vooruitgang in de andere [109]. Wij veronderstellen dat het combineren van beide benaderingen in sociotechnische metingen van cybercriminaliteit, door Internetmetingen uit te voeren met inachtneming van criminologische aspecten, kan leiden tot verbeterde meetmethoden. Welke innovaties zijn nodig om dergelijke metingen uit te voeren? Hoe kunnen metingen van cybercriminaliteit bijdragen aan de bestrijding ervan?

De opkomst van nieuwe vormen van cybercriminaliteit vereist wetenschappelijke innovatie om deze te kunnen meten. In 2019 won cryptojacking in de browser aan populariteit onder cybercriminelen als een methode om winst te genereren uit gestolen rekenkracht. Verschillende onderzoekers begonnen technieken te ontwikkelen om illegale cryptominingactiviteiten te detecteren in Web browsers [101, 127, 149, 199, 205, 207]. Er werd echter weinig aandacht besteed aan het vaststellen van de schaal van deze cybercriminaliteit (d.w.z. het bepalen van de prevalentie ervan onder Internetgebruikers) of aan het onderzoeken van grootschalige campagnes van cryptominingwebsites die onder controle staan van eenzelfde actor. Daarnaast is er geen onderzoek verricht naar andere distributiemethoden, zoals MITM-aanvallen op gecompromitteerde Internetinfrastructuur. Verder constateren wij dat geografische afbakeningen in Internetmetingen schaars zijn vanwege het mondiale karakter van het hedendaagse Internet. Hoewel verschillende eerdere studies de relevantie van de ruimtelijke dimensie in cybercriminaliteitsmetingen hebben benadrukt [73, 98, 181], zijn er maar weinig studies die Internetmetingen van cybercriminaliteit in Nederland behandelen. Naast opsporingsdiensten en beleidsmakers spelen diverse organisaties en mechanismen binnen het *anti-abuse ecosysteem* een belangrijke rol in de bestrijding van cybercriminaliteit. Binnen dit ecosysteem rapporteren bepaalde partijen het bestaan van kwaadaardige inhoud aan de intermediairs of facilitators die deze hosten. Hoewel het verwerken van dergelijk abuse materiaal al jarenlang plaatsvindt, is er weinig bekend over de interne besluitvormingsprocessen van dergelijke hostingproviders bij het verwerken van dit soort meldingen. Eerdere studies hebben voornamelijk een externe benadering gehanteerd en geanalyseerd hoe schadelijk materiaal zoals phishingwebsites, spamactiviteiten en malware C&C servers verspreid is over verschillende providers [219, 232]. Dergelijke onderzoeken bieden echter geen inzicht in de interne dynamiek van de betrokken bedrijven, een aspect dat wij in dit proefschrift willen adresseren. Ten slotte zijn wij benieuwd hoe cybercriminaliteitsmetingen kunnen bijdragen aan inspanningen van politiediensten om cybercriminaliteit te verstoren. Welke informatie over cybercriminaliteit is essentieel voor het ontwikkelen van op bewijs gebaseerde governance? Sommige academici zijn ervan overtuigd dat hun wetenschap-

pelijke metingen van cybercriminaliteit bijdragen aan dergelijke governance [239, 249], maar wij hebben geen wetenschappelijk bewijs gevonden dat deze beweringen ondersteunt.

Dit proefschrift bestudeert de uitdagingen van het uitvoeren van socio-technische metingen van financieel gemotiveerde cybercriminaliteit. Wij onderzoeken hoe cybercriminaliteit adequaat gemeten kan worden en hoe inzichten uit deze metingen kunnen bijdragen aan verbeterde governance. Daarom formuleren wij de volgende onderzoeksvraag: *"Hoe kunnen socio-technische metingen van cybercriminaliteit worden uitgevoerd ter ondersteuning van cybercrime governance?"*. Vijf afzonderlijke studies, opgenomen als hoofdstukken in dit proefschrift, trachten deze onderzoeksvraag te beantwoorden.

Hoofdstuk 2 richt zich op cryptojacking op websites en schat de prevalentie van deze vorm van cybercriminaliteit, aangezien dit soort cryptomining sinds 2017 sterk is toegenomen. Twee grootschalige studies worden uitgevoerd om webgebaseerde cryptojacking te onderzoeken. Één gericht op georganiseerde cryptomining en één op de prevalentie van cryptojacking op het gehele Internet. Wij gebruiken een combinatie van databronnen om een representatieve dataset op te bouwen voor onze metingen en maken gebruik van eerdere academische studies om talloze domeinen te geautomatiseerd te bezoeken en te zoeken naar overlappende kenmerken die worden ingezet in cryptojackingcampagnes. Hiermee identificeren wij de meest gebruikte aanvalsvector voor cryptojacking, de meest getroffen categorie websites en stuiten wij op grootschalige cybercriminele campagnes.

Vervolgens gaan we in hoofdstuk 3 dieper in op cryptojacking-aanvallen door geïnstalleerde kwaadaardige cryptojacking-scripts op gecompromitteerde Internetinfrastructuur (zoals routers) te onderzoeken. Door een kwetsbaarheid in de firmware van MikroTik routers konden cybercriminelen het webverkeer dat door deze routers stroomde herschrijven en schadelijke cryptojacking-code injecteren. Het volgen van deze aanvalsvector onthulde een geheel nieuw domein van cryptojacking-activiteit, dat veel groter en georganiseerder bleek te zijn dan cryptojacking via websites. Op basis van NetFlows geregistreerd in een Tier 1-netwerk, semiwekelijkse Internetcrawls en data uit een netwerktelescoop analyseren we de modus operandi van deze cybercriminelen. We laten zien hoe deze grootschalige cryptojacking-installaties met elkaar samenhangen en schetsen de levensloop van deze gecompromitteerde Internetinfrastructuur. Daarnaast voeren wij een campagne analyse uit om verschillende niveaus van verfijning onder aanvallers te onderscheiden.

In hoofdstuk 4 verschuift de focus naar een ander type cybercriminaliteit: phishing, oftewel het illegaal verzamelen van inloggegevens. Hoewel er al veel phishing-onderzoek is gepubliceerd [58, 119, 139, 241, 262], ontbrak een diepgaande analyse van de cybercriminologische TTP gericht op Nederlandse burgers. Voor dit onderzoek richten wij ons specifiek op de eerste component in de waardeketen van phishing aanvallen: de ontwikkeling. Deze fase omvat het ontwikkelen en verhandelen van phishing kits die worden gebruikt bij phishing aanvallen. Wij verzamelen een groot aantal van deze phishing kits, analyseren het gebruik van TLS-certificaten door cybercriminelen en ontwikkelen een crawler om phishing domeinen te identificeren die gericht zijn op de klanten van de Nederlandse financiële sector. Hierdoor konden we de volledige levensloop van phishing aanvallen vastleggen en de technieken analyseren waarmee

slachtoffers worden verleid om hun inloggegevens prijs te geven. Deze inzichten stelden ons in staat een volledig beeld te schetsen van de TTP binnen het Nederlandse phishing landschap en beleidsmatige aanbevelingen te formuleren voor anti-phishing initiatieven.

Hoofdstuk 5 onderzoekt de instrumenten die zijn ingezet om Internet abuse, waaronder cybercriminaliteit, te bestrijden. Deze instrumenten variëren van wetgeving om auteursrechtelijk beschermd materiaal te verwijderen tot technische middelen zoals het opstellen van blocklists om spam te stoppen, en zijn gebaseerd op industriestandaarden voor abuse meldingen: het rapporteren van abuse aan de eigenaars van een netwerk of online bron met het verzoek tot mitigatie. Door een unieke samenwerking met een Nederlandse opsporingsdienst kregen wij toegang tot de operationele back-end van een hosting provider die bekend is vanwege haar hoeveelheden abuse. We analyseren de abuse meldingen en de opvolgende acties om te onderzoeken welke mechanismen binnen het anti-abuse ecosysteem invloed hebben op de bestrijding van abuse. We constateren dat meldingspercentages sterk afhankelijk zijn van de bron en het type abuse melding, en in mindere mate van de betrokken klant. Governance-instrumenten zoals blocklisting, de-peering en verzoeken van opsporingsdiensten welke directe gevolgen kunnen hebben voor de bedrijfscontinuïteit beïnvloeden de doorzetting van meldingen naar klanten, terwijl individuele abuse rapportage eenvoudig kunnen worden genegeerd.

Hoofdstuk 6 brengt de voorgaande hoofdstukken samen door eerdere wetenschappelijke metingen van cybercriminaliteit te analyseren. Deze metingen zijn doorgaans gebaseerd op grootschalige Internetmetingen van domeinen, software of aanvalsverkeer. We benutten het concept van waardeketens om 38 studies te structureren die metingen van phishing, booter-diensten en remote access trojans (RATs) bevatten. We beoordelen hoe andere onderzoekers hun metingen van deze financieel gedreven vormen van cybercriminaliteit hebben opgezet, onderzoeken hun methodologieën en analyseren de gebruikte databronnen om gemeenschappelijke factoren te identificeren. Vervolgens laten we medewerkers van opsporingsdiensten reflecteren op enkele van deze metingen en gezamenlijk verkennen we het onbenutte potentieel voor nieuwe metingen die kunnen inspelen op de huidige behoeften in de opsporing van cybercriminaliteit. Door de input van opsporingsdiensten te combineren met de gestructureerde analyse van eerdere academische metingen, konden we vaststellen welke kenmerken van wetenschappelijke metingen nuttig zijn voor opsporingsdiensten – oftewel, hoe zij hun schaarse middelen het best kunnen inzetten.

Tot slot presenteert hoofdstuk 7 de belangrijkste bevindingen van ons onderzoek, reflecteert het op de maatschappelijke impact en de governance implicaties, en schetst het richtingen voor toekomstig onderzoek. Een belangrijk en vaak vergeten aspect van Internetmetingen naar cybercriminaliteit is dat dergelijke metingen daadwerkelijk gericht moeten zijn op het meten van *criminaliteit*. Criminaliteit omvat individuen (of groepen) met een duidelijke (winstgedreven) intentie en bijbehorende besluitvormingsprocessen waarin risico's en beloningen worden afgewogen. Dergelijke kenmerken van criminaliteit zijn gemeengoed binnen de criminologie, maar worden zelden meegenomen in de metingen die door computerwetenschappers worden uitgevoerd. Daarnaast constateren we dat Internetmetingen van cybercriminaliteit vaak een duidelijke geogra-

fische afbakening missen, terwijl dit voor opsporingsdiensten cruciaal is, aangezien zij opereren binnen lokale jurisdicties. Bij het schrijven van dit proefschrift werd duidelijk dat de verschillende meet methodologieën kunnen bijdragen aan de ontwikkeling van metingen die direct toepasbaar zijn voor opsporingsdiensten, zij het op verschillende manieren. Waardeketens vormen een nuttig kader om te bepalen wat precies wordt gemeten, levenscyclusanalyse tonen de (on)mogelijkheden voor governance-interventies, en campagne analyses bieden waardevolle inzichten in de mate van organisatie binnen een bepaalde vorm van cybercriminaliteit. Inzichten in de mate van organisatie en de verschillende typen cybercriminelen kunnen opsporingsdiensten helpen bij het ontwikkelen van effectieve governance-strategieën. Tot slot constateren we een discrepantie tussen de focus van academische metingen van cybercriminaliteit en de behoeften van wetshandhaving. De meeste metingen richten zich op de implementatie en uitvoering van cybercriminaliteit, terwijl opsporingsdiensten juist meer inzicht willen krijgen in de ontwikkelingsfase en hoe winsten worden gemaakt.

Samenvattend, metingen die rekening houden met de intentie achter criminaliteit, die zowel geografisch als op actorniveau goed zijn afgebakend, en die campagne analyses uitvoeren om niet alleen de prevalentie maar ook de mate van organisatie te onderzoeken, kunnen een waardevolle bijdrage leveren aan de bestrijding van cybercriminaliteit. Het structureren van academische analyses rond waardeketens, levenscycli en campagne analyses biedt onderzoekers methodologische waarborgen om te garanderen dat zij daadwerkelijk meten wat zij beogen te meten.

1

INTRODUCTION

1.1. BACKGROUND

The introduction of the Internet in the early 1990s enabled emailing, Web surfing, and videoconferencing, but also opened doors for criminals to abuse these new opportunities. Scamming Internet users, compromising computer systems, or robbing online bank accounts are practices we call *cybercrime* nowadays. Categorizing these malicious activities as cybercrime is easy, but sharing a complete and well-established definition of it is not. Various definitions exist and are actively used by scientists, governments, and law enforcement agencies. Europol refers to *cyber-dependent* crime as "*any crime that can only be committed using computers, computer networks or other forms of information communication technology (ICT)*" (Europol [66]). According to Akdemir et al. [2], the most commonly used definition in scientific literature was put forward by Gordon & Ford in 2006, who define cybercrime as "*any crime that is facilitated or committed using a computer, network, or hardware device*" (Gordon & Ford [95]). The European Commission follows this broad definition and refers to cybercrime as "*a broad range of different criminal activities where computers and information systems are involved either as a primary tool or as a primary target*" (European Commission [64]). By taking a closer look at these definitions, we observe two subcategories of cybercrimes. First, crimes in which computers are both the instrument and the target (e.g., hacking, botnets, and cryptojacking), which are referred to as cybercrime in a narrow sense by Leukfeldt et al. [143]. Second, crimes in which computers are essential for their execution but not the primary target (e.g., romance scams through social media platforms), which is referred to as cybercrime in a broader sense [143]. In this dissertation, we follow the definition of Gordon & Ford to study cybercrime in both a narrow and broader sense. As most cybercrime is driven by monetary gain [78, 115], we limit ourselves to crimes committed with monetary motivations and exclude the actions of nation-state actors or hacktivists.

We are not the first to study cybercrime, and much is already known about the tactics, techniques, and procedures (TTP) of modern cybercriminals. Yet, the scale of cybercrime is ever-expanding, impacting many innocent citizens and resulting in

millions of dollars in losses for the global society [227]. Law enforcement agencies and policymakers worldwide try to disrupt these miscreants but often lack a clear picture of the crimes they are fighting. Scholars have tried to assess the scale of various cybercrimes, either through large-scale Internet measurements [124, 241, 263], or through qualitative criminology research [143, 150, 206]. Estimations of the size of cybercrime, however, often rely on reported incidents, court cases, or police reports [31]. Yet, the majority of cybercrime incidents are never reported to the police [57, 59], which implies a much higher amount of actual (attempted) cybercrime. The question arises as to how much higher this amount would be and how to measure it to assist in the effective governance of cybercrime.

Before we can answer this question, we require a clear understanding of the inner workings of cybercrime, what steps a cybercriminal undertakes to be successful, and how previous scholars have assessed Internet artifacts to analyze cybercrime. To do that, we elaborate on the use of resources in cybercrime, introduce a variety of lenses that are used to structure their measurements of cybercrime, and discuss the synergy between Internet measurements and criminology.

1.1.1. CYBERCRIMINAL RESOURCES

When examining traditional crime, one can distinguish various steps a criminal (group) undertakes to be successful. Taking illegal drug production and trafficking as an example, specialized equipment to produce these drugs is necessary at first, as well as a process to acquire raw materials, distribute goods, and manage buyers [110]. All of these steps are necessary to operate a successful criminal business and are highly dependent on each other. Illegal drugs cannot be made without raw materials, and profits can not be made without a buyer. Hence, we distinguish dependencies in this criminal operation. Insights into the (inter)dependencies within such criminal operations are essential to law enforcement and policymakers in designing effective interventions [164, 246]. For instance, insights into the acquisition of raw materials resulted in the prohibition of selling and possessing them [110], making it easier for law enforcement agencies to arrest individuals who carry these goods.

This is no different from cybercrime in the online world. Cybercriminals need a variety of products and services to set up their attacks [72], just as criminals involved in illegal drug production and trafficking. No malware infection can happen without a piece of malicious software, communications with infected machines can not be established without a command-and-control (C&C) server infrastructure, and every phishing attack needs a website displaying a bogus login screen ready to capture user credentials. Zooming in on the latter example shows clear similarities to the illegal drug production and trafficking example from the previous paragraph. The first necessity, acquiring a phishing website, is as essential to this crime as acquiring raw materials for illegal drug production. Similarly, a hosting provider and a domain name to host this malicious website can be compared to the specialized equipment essential for illegal drug production. Finally, the monetization of stolen credentials after a successful phishing attack shows great similarity to an ordinary drug deal in which a producer sells its drugs to a buyer. Just like traditional crime, most cybercrime is driven by profit motives [78].

Table 1.1: Examples of the required resources for cybercrime, including both products and services.

Products		Services	
Phishing kits	Malware binaries	Server hosting	Certificate Authorities
Mining scripts	Target lists	Domain registration	SMS gateway services
Domain names	TLS certificates	URL shorteners	Cryptocurrency mixing

A major difference between traditional drug crime and cybercrime is how they can be measured. Traditional crimes rely on physical products that are often traded decentralized and outside anybody's view, making it hard to measure their presence [164]. Insights into the dependencies within such criminal operations are often obtained by police investigations following arrests, confessions from ex-criminals, or through undercover operations. This is different from cybercrime, which can be measured more easily and safely [164], as the observability of resources used in cybercrime and their centralized collection allows for external measurements and does not require undercover police operations. For example, domain names are publicly listed in DNS zone files [107], websites are hosted on a public IP address, and TLS certificates are stored in the freely available Certificate Transparency Logs [92]. Table 1.1 lists more of these resources used in cybercrime. As this table shows, resources can be both products and services. Many of the resources used for illegal drug production and trafficking – ranging from raw materials to anything related to the production or trafficking of illegal drugs – are explicitly prohibited in most countries. However, just like the use of TLS certificates in cybercrime, some resources (e.g., drain cleaner – sodium hydroxide – for the production of GHB) are not criminal, as they are primarily used for benign purposes. The possibility of conducting cybercrime measurements through observation of public resources does not inherently imply that the process is trivial. The Internet is immense and constantly changing, making it difficult to detect criminal activity among the many benign Internet activities, and it requires robust measurement approaches to structure analyses of cybercrime.

1.1.2. MEASUREMENT LENSES

Following the introduction of cybercriminal resources, the question arises as to how to measure the use of such resources within cybercriminal operations. A variety of viewpoints can be taken to study cybercrime and measure such usage, which we define as *measurement lenses*. This section details three such lenses used to examine resources in cybercrime: value chains, life cycles, and campaign analysis.

VALUE CHAINS

The first lens we introduce provides an economic viewpoint on the resources cybercriminals depend on and defines the steps to criminal success as components in a *cybercrime value chain* [248] – illustrated in Figure 1.1 in the yellow rectangles. Each component in the value chain requires a set of resources as an input and a resource as an output, shown in green in Figure 1.1. Components in the value chain are not strictly executed sequentially, as some components can be executed in parallel – e.g., a booter service can deploy its storefront website and perform reconnaissance operations simultaneously. The first

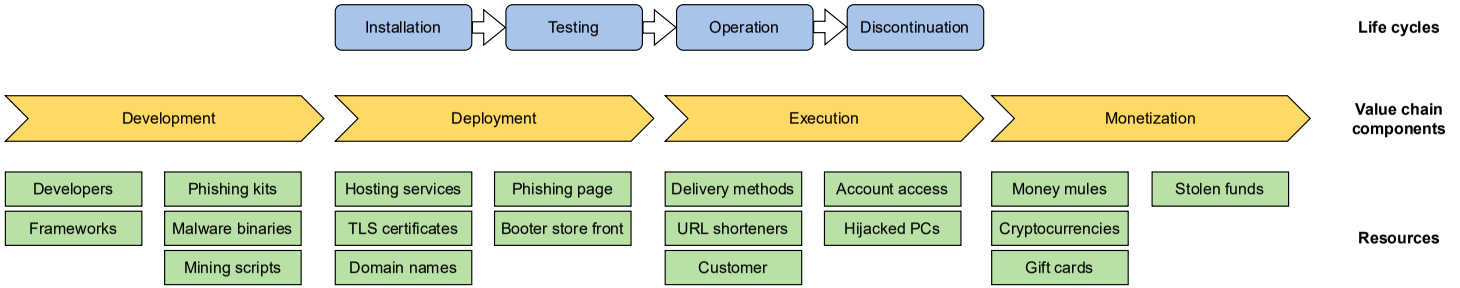


Figure 1.1: A general overview of a cybercriminal value chain, its resources, and a life cycle illustration.

component in a cybercriminal value chain is often focused on (software) development. It requires a skilled developer and results in products like a phishing kit or a malware binary as an output. This, combined with other resources such as hosting services, TLS certificates, or domain names, is used as input for the next component, deployment The next component is execution, which encompasses the resources required to conduct the actual cybercrime. That is, the resulting outputs from both the development and the deployment components, as well as additional resources like, among others, URL shorteners or SMS gateways to get, for example, phishing lures at potential victims. If a cybercriminal operates in a cybercrime-as-a-service model, customers can also be considered an input resource for this component. The resulting account access (in the case of phishing), hijacked PCs (in the case of malware infections), or DDoS attacks (in the case of a booter service) are examples of outputs for this component. The last component in the value chain is monetization since profit-driven cybercriminals eventually have to convert cyber attacks into monetary value. Here, the resulting resources from the execution component, together with money laundering constructs like money mules, cryptocurrencies, or the use of gift cards, allow for gaining profits.

Formerly, cybercriminals self-organized most components [239], but nowadays, as many cybercriminals do not possess the expertise to fulfill every component themselves, they depend on specialists to fulfill components for them, i.e., through outsourcing [14, 249]. Specialized third parties provide resources concerning both products (e.g., software) and services (e.g., hosting services), but are not necessarily criminals themselves. A phishing kit developer is obviously criminal, but the Certificate Authority that supplies TLS certificates for serving a phishing website over HTTPS can not be criminalized. At every component in the value chain, a criminal selects the necessary resources, either by self-organizing or outsourcing them. Some resources can be acquired from multiple suppliers for different prices (e.g., domain names) and can easily be replaced, whereas other resources are scarce and can only be acquired from a single supplier for a set price (e.g., malware binaries). Every resource comes with a price and an accompanying return on investment value, reflected by the durability and the quality of the resource [239]. This 'value add' is an important aspect of determining interdependencies in the underground market, as it is fragile to price increases and thus a target for interventions, according to Thomas et al. [239]. When outsourcing resources to specialized third parties, each specialist will take a cut. Hence, to avoid these cuts or to expand operations, cybercriminals can decide to self-organize components in the value chain themselves. We refer to practice this as *vertical integration* in the value chain [239]. For example, when phishing kit developers deploy their own phishing kit. By doing so, criminals become less dependent on other actors and can potentially increase their profits. However, removing specialized partners from the value chain could also decrease the quality of execution, making criminals more vulnerable to being caught by law enforcement. *Horizontal integration* indicates that a component in the value chain is present in multiple cybercrime value chains. (Bulletproof) hosting providers are a good example of such horizontal integration, with their services being present in almost all criminal business models. Operating a horizontally integrated service generally improves the quality (mostly due to economies of scale), but becoming too large could result in unwanted attention from law enforcement, identifying it as a facilitator of crime [239].

LIFE CYCLES

A different lens to analyze cybercriminal operations covering one or more components in the value chain is through the analysis of *life cycles*, sometimes referred to as end-to-end life cycles [144, 183]. In contrast to value chains – which are structured around the added value of resources in executing a cybercrime – life cycles allow for sequential measurements over time in or over multiple components that reflect the different stages of executing a cybercrime. Measurements of timestamps of key events within the execution of cybercrime are, therefore, the main focus of life cycle analysis and allow for crucial insights into the speed of cybercriminal operations. Past researchers have been leveraging life cycles to study, among others, spam [144], phishing [183], and ransomware [102]. Just as value chains differ per type of cybercrime, so do life cycles differ per study. In the life cycle analysis of spam by Levchenko et al. [144], the life cycle is defined as all steps between the arrival of a spam email and the delivery of the product, ranging from domain names, payment infrastructure, to shipping services. This thus includes the deployment, execution, and monetization components in the value chain. Analysis of ransomware by Huang et al. [102] defines their life cycle measurement from the moment a victim retrieves the ransomware payload to the moment paid Bitcoin ransom is liquidated by ransomware operators – hereby covering the execution and monetization components of the value chain. The blue rectangles in Figure 1.1 depict another life cycle covering the deployment and execution components, during which a criminal installs, tests, operates, and dismantles a malicious website. Given a resource obtained, a cybercriminal needs to install this resource – either a phishing kit, mining script, or malware binary – at some location. When installed successfully, testing is necessary to ensure the correct working of this resource. After testing, the resource is ready for use, waiting for its victims. Discontinuation can be the result of many causes and is initiated by either the cybercriminal itself or by other parties. For example, the discontinuation of a phishing domain could be the result of a notice-and-takedown (NTD) urging the hosting provider to remove the malicious website or the actor itself after being listed on a blocklist.

CAMPAIGN ANALYSIS

The last lens researchers use to analyze cybercriminal operations is by performing *campaign analysis*, which refers to the practice of attributing multiple attacks to the same actor (group) by identifying unique, overlapping resources or techniques [106]. The term was first introduced by researchers examining spam [55] but is nowadays used to cluster all kinds of cybercrime. A cluster of cybercriminal operations attributed to an actor is referred to as a campaign, and tracking such clusters allows researchers to improve their understanding of this actor's TTPs. Campaign analysis has been performed by several scholars in the past [58, 127, 139, 190, 199] but can be found more commonly outside academia at security researchers working for various threat intelligence groups like Google's TAG [91] or Palo Alto's Unit42 [188]. For example, they show commonly used software (resources in the development component), popular means of malicious payload delivery (resources in the execution component), or the use of mixing services (a resource in the monetization component). Such information is pivotal for designing effective defenses or interventions to stop these cybercrimes. For example, when numerous phishing attacks can be attributed to the same actor and this actor uses one specific

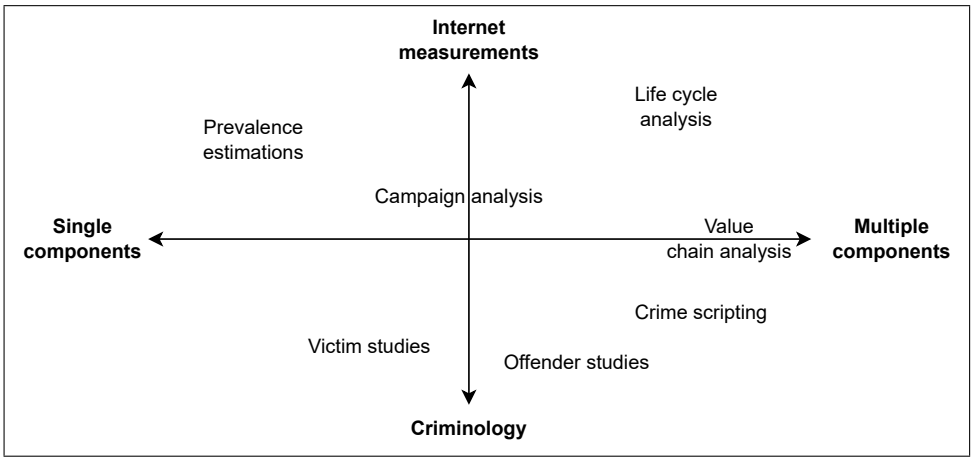


Figure 1.2: The academic disciplines engaged in cybercrime measurement are represented along the Y-axis, while the varying amount of value chain components included in the analyses are depicted on the X-axis.

hosting company, a straightforward intervention for law enforcement would be to demand that this hosting provider share the identity of the criminal behind these websites. However, as cybercriminals can easily switch hosting providers, the question arises as to what the long-term effects of such an intervention would be.

1.1.3. INTERNET MEASUREMENTS & CRIMINOLOGY

As the word *cybercrime* suggests, measurements of profit-driven cybercrime involve two disciplines of academic research: Internet measurements (*cyber*) and criminology (*crime*). Both are aimed at identifying the TTP that cybercriminals employ [239] but approach their measurements in a totally different way [109]. As a result, both approaches come with their advantages and challenges.

INTERNET MEASUREMENTS

Academic work originating from the discipline of Internet measurements, shown on top of the Y-axis in Figure 1.2, is centered around (the creation of novel techniques to) measure the resources used in profit-driven cybercrime or on a specific technical capability or component within the value chain. Authors often deploy their (novel) measurement techniques 'in the wild' [124, 172, 241, 263] to examine (a part of) the Internet to estimate the prevalence of a certain cybercrime. For example, to identify websites that employ cryptojacking, Parra Rodriguez et al. [205] proposed *RAPID*, Liu et al. [149] created *BMDetector* and Hong et al. [101] built *CMTracker*, all published within one year. Each of these studies created robust technology to identify cryptojacking websites. Similarly, in the field of phishing, researchers are constantly crafting new methods to detect phishing websites with ever-rising accuracy. From using simple term frequency algorithms by Zhang et al. [262] back in 2007 to applying advanced ensemble learning in more recent work by Kalabarige et al. [119] in 2022. Such research comes with several advantages. It often requires no data from external parties, such

as difficult-to-acquire police records or access to cybercrime victims or offenders, as researchers are free to design their data-gathering methodologies and can collect the data from the Internet themselves. Automated Internet measurements can run for extensive periods of time, ranging from months to years, allowing longitudinal analyses on large datasets. However, this freedom comes with a caveat: selecting the right dataset. Deploying a measurement system to analyze the entire Internet is practically impossible. Hence, studies will always analyze a subset from which to extrapolate. Disadvantages of performing Internet measurements are the difficulty of creating a representative dataset from which to extrapolate and the lack of data concerning the human impact of cybercrime. For example, multiple researchers [101, 172] assessed the scale of cryptojacking by crawling (parts of) the one million most popular websites listed in the Alexa Top 1M [3]. However, as Scheitle et al. pointed out in their research on the use of such top lists in scientific studies, these top lists "*generally overestimate results compared to the general population by a significant margin, often even an order of magnitude*" (Scheitle et al. [213]). On top of that, we position the global cybercrime measurements performed by security vendors, who have a clear incentive to exaggerate their findings to increase their profits [6]. Additionally, Internet measurements pay no attention to the human beings involved, neither the victims nor the perpetrators. Measuring the amount of online malicious activity does not reveal how many people fell victim, what the average amount of money lost was, or what impact the attack had on someone's well-being. Those questions are typically answered by research originating from the other discipline in Figure 1.2: criminology.

CRIMINOLOGY

In contrast to Internet measurements, criminologists study the human beings involved in cybercrime to provide theories and conceptual frameworks as ways to think and reason about crime and social control. They do that through various quantitative as well as qualitative methodologies, as illustrated at the bottom of the Y-axis in Figure 1.2. There is an ongoing debate within criminology regarding quantitative and qualitative approaches, which is well characterized by Christie [33]. He differentiated between 'near data', which includes information pertaining to a small number of participants while providing rich insights, and 'distant data', such as large datasets stemming from surveys or large amounts of case files, but provides little in-depth understanding. Both types of data can be gathered through direct observations (e.g., direct interactions with criminals) or indirect observations (e.g., written documents like diaries or police reports). Information on cybercrime victims and perpetrators can originate from, among others, police reports [150], victim studies [200], or public surveys [24]. Qualitative academic studies based on police records are characterized by their great level of detail regarding cybercriminal value chains, economic decision-making, and human psychology, but also by the relatively small dataset sizes. For example, a study on phishing by Loggen & Leukfeldt unraveled the crime scripts of phishing networks in the Netherlands [150]. Their work shares detailed insights into the modus operandi of these cybercriminals but is based on only 45 court cases in the Netherlands. As a consequence, this work only covers known criminal cases and arrested offenders. Another study, by Schiks et al., performed an offender study that compared the intellectual capabilities of cyber-

criminal suspects to traditional convicted criminals and the general population [214]. It shared interesting insights into cybercriminals, but is again based on a relatively small dataset of 429 persons, of which 143 were suspects of cybercrime. Both studies examined the behavior of cybercriminals who have been identified after a successful police investigation. However, as only a small portion of cybercrime is reported to the police [59], and an even smaller portion of cases result in an arrest, using solely police reports will likely lead to an underestimation of the scale of cybercrime. To cope with this limitation, some researchers perform quantitative research by setting up large-scale surveys to question citizens directly about their cybercrime experiences. For example, Breen et al. reported on the first large-scale academic survey of consumer cybercrime experiences in the United States [24]. In 2022, over 11k participants filled in a survey, which showed that cybercrime against individual citizens is rare – most types of cybercrime are experienced by less than 1% of the population – and causes a median loss of $100 [24]. A similar study in the Netherlands in 2013 showed that 8.5% of all Internet users fell victim to some type of cybercrime and that most participants deemed financial institutions and website owners responsible for the security of the Internet, rather than the police [57].

Both approaches to measuring cybercrime have their strengths and weaknesses: Internet measurements stemming from Computer Science allow for obtaining a broad picture of a certain cybercrime involving many data points, but come with dataset selection challenges and are not focused on gathering in-depth insights into cybercriminal operations. Criminologists often present this level of detail, but their findings are narrowly scoped and gathered from smaller datasets (e.g., 'near data' research). Despite the large body of work produced by both disciplines, their research efforts have remained mostly disjoint, with researchers on one side not benefiting from the advancements proposed by the other [109]. In line with the few scholars that are already bridging this gap [47, 144, 224, 239], we hypothesize that combining both approaches into socio-technical measurements of cybercrime, hereby performing Internet measurements while taking criminology aspects into account, could result in improved measurements. What innovations are there to be made to perform such measurements? How to perform cybercrime measurements to assist in cybercrime governance? The next section identifies such knowledge gaps in the current state of the art, followed by a specification of the research aim. The latter section also introduces the main research question and its sub-questions.

1.2. RESEARCH GAPS

Given the background on cybercrime outlined in the previous section, the presentation of various measurement lenses to structure the analysis of profit-driven cybercrime, and the potential synergy between criminology and Internet measurements, we identify three gaps in existing scientific literature and detail them in the following paragraphs.

INNOVATIONS IN INTERNET MEASUREMENTS FOR EMERGING CYBERCRIMES

The rise of new and emerging types of cybercrime requires scientific innovations to measure them. In 2019, browser-based cryptojacking gained popularity among cybercrimi-

nals as a means to earn a profit from stolen computing power. Several researchers started designing techniques to detect illicit in-browser cryptomining activity [101, 127, 149, 199, 205, 207], yet little attention was paid to determining the scale of this cybercrime (i.e., assessing its prevalence among Internet users) or investigating large campaigns of cryptomining websites related to a shared actor. Modest explorations into such campaigns have been made by several researchers [63, 127, 199], yet no study systematically examined these campaigns in-depth and identified common denominators or shared origins. Additionally, websites are just one attack vector that can be leveraged to distribute illicit cryptomining scripts. No research had been conducted into other means of distribution, such as MITM attacks on compromised Internet infrastructure. We consider the lack of in-depth cryptojacking campaign analysis, systematic analysis of its prevalence among Internet users, and an investigation of the MITM attack vector as our first research gap we should address.

JURISDICTION-FOCUSED INTERNET MEASUREMENTS OF CYBERCRIME

As explained in § 1.1.3, qualitative studies stemming from criminology often examine cybercrime with a narrow scope, focused on a specific jurisdiction or region. Examples of such include the work of Domenie et al. [57] on the victimization of cybercrime within the Netherlands or a study by Leukfeldt [142] that examined criminals involved in phishing, concentrated within one neighborhood in Amsterdam. Such demarcations are scarce in Internet measurements due to the global character of the modern-day Internet. Jurisdiction-focused measurements – or measurements demarcated by a specific linguistic area – can be executed by fewer researchers simply because fewer researchers have the required knowledge. Additionally, since top-tier academic venues for Internet measurements are keener on global measurements because of their generalizability and perceived applicability to the entire academic population, academic incentives to perform jurisdiction or geographically-focused Internet measurements of cybercrime are lacking. Although various previous studies hinted at the importance of spatiality in cybercrime measurements [73, 98, 181], there is no abundance of studies involving Internet measurements of cybercrime focused on the Netherlands. We identify the low amount of such jurisdiction-focused measurements with actionable results for the global Internet as our second research gap.

HOW HOSTING COMPANIES HANDLE ABUSE TO FIGHT CYBERCRIME

Besides law enforcement and policymakers, there is an important role in the fight against cybercrime reserved for a variety of organizations and mechanisms operating within the *anti-abuse ecosystem*. Within this ecosystem, there are parties reporting the existence of malicious content to the intermediaries or facilitators hosting it. Hosting providers can decide to notify the customer owning this content or to remove the content themselves [116]. Although abuse handling has been around for many years, little is known about the internal decision-making processes within hosting providers to deal with abuse reports. Past studies have predominantly taken an external point of view and analyzed how abusive content – e.g., phishing sites, spamming activity, and malware C&C servers – are allocated across providers [219, 232]. Such studies, however, lack insights into the internal dynamics of the respective companies. Does the provider act

on the abuse report and notify the client? Does it take action against abusive clients? Companies that are perceived as not swiftly responding to abuse reports are often referred to as 'bulletproof' or 'bad' hosters. That is, they are seen as impervious to abuse reports [4]. Some characteristics of bulletproof hosting have been described in both earlier work [86, 180] and industry reports [153]. By calling a company a 'bulletproof' hoster, one assumes intent – i.e., knowingly ignoring abuse reports and enabling abuse. Yet, one cannot measure intent by looking from the outside through external abuse data. This requires an inside view, which is very hard to obtain. Since no existing scientific studies have been able to achieve this inside view of a hosting company managing its abuse, we consider this our third research gap.

SCIENTIFIC MEASUREMENTS TO ASSIST GOVERNANCE

Besides the three already-identified research gaps related to empirical measurements of both cybercrime and its remediation within the anti-abuse ecosystem, we identify a fourth research gap as soon as the other gaps have been resolved. That is, how can these cybercrime measurements assist in disrupting cybercrime? What kind of information related to cybercrime is essential for designing evidence-based governance strategies? Some previous cybercrime measurement studies stated that (their) measurements are of great value for, for example, law enforcement agencies. As illustrated by van Wegberg, "*understanding these interactions would help creating better evidence-based law enforcement strategies*" (van Wegberg [249]), or, as Thomas et al. put it, "*a systematized understanding of underground relationships is crucial to developing effective, long-lasting countermeasures*" (Thomas et al. [239]). These statements illustrate that some academics are convinced that their scientific measurements of cybercrime could assist governance, yet we found no scientific proof for these claims. We define the lack of knowledge regarding the practicality and actionability of scientific cybercrime measurements for governance as the fourth research gap that needs to be resolved.

In short, we identify the following four gaps in state-of-the-art academic research:

1. We lack innovative methods to perform socio-technical measurements examining upcoming cybercrime phenomena.

2. We do not find many jurisdiction-focused Internet measurements of cybercrime that apply directly to cybercrime in the Netherlands.

3. We lack an inside view into the abuse-handling practices of hosting companies to study the effects of mechanisms within the anti-abuse ecosystem.

4. We do not know how information gained from cybercrime measurements can assist in effectively designing governance to fight cybercrime.

We employ five research activities to fill these four gaps. We start by selecting cryptojacking as a cybercrime to perform multiple large-scale socio-technical measurements to estimate its prevalence among Internet users and assess to what extent cryptojacking is part of organized cybercriminal activity by performing campaign analysis. In the

second study, we dive deeper into campaign analysis by researching large cryptojacking campaigns on compromised Internet infrastructure. Thereafter, we shift focus and perform a local measurement of phishing focused on the Netherlands through a combination of value chain and life cycle analysis. Then, to address our third research gap, we perform a socio-technical measurement of a horizontally integrated resource present in many cybercrime value chains: hosting services. From scrutinizing a hosting service provider with a reputation for abuse using ground-truth data, we learn what governance instruments effectively reduce cybercrime and assess the effects of certain mechanisms within the anti-abuse ecosystem. We resolve the fourth identified research gap in our last study by reviewing how previous researchers performed their large-scale Internet measurements of cybercrime and by assessing how applicable these studies are for law enforcement by letting law enforcement professionals reflect on previous academic work.

1.3. RESEARCH AIMS & QUESTIONS

This dissertation studies the challenges of performing socio-technical measurements of profit-driven cybercrime. We investigate how cybercrime can be measured adequately and how insights gained from these measurements can improve governance. Given the research gaps identified in § 1.2 and the research aim stated above, we identify the main research question as follows:

How to perform socio-technical measurements of cybercrime to assist in cybercrime governance?

As shown by this central research question, the focus of this dissertation is to understand and perform measurements of cybercrime to assist in creating better governance to fight cybercrime. We leverage the concepts of cybercriminal value chains, life cycles, and campaign analysis to structure, analyze, and measure different types of cybercrime. Building on the knowledge from both Internet measurements and criminology, we set out to discover how to perform socio-technical cybercrime measurements to assist in creating better-informed governance strategies. Therefore, the scientific contribution of this work lies in the designed and used methods to gather actionable insights and not necessarily in the empirical results gained from the individual studies themselves. Measurements of cybercrime can become outdated quickly as the cybercrime landscape changes rapidly. However, the methods proposed and used to obtain these results remain relevant and can be considered the main contribution of this dissertation. Therefore, we contribute to science and society by showcasing methods to create evidence-based governance to fight cybercrime. This could aid legislators in writing data-driven criminal law, help policymakers formulate evidence-based approaches, and assist law enforcement agencies in guiding their investigations to maximum effect. To answer the research question mentioned above, we follow a mixed-methods approach that leverages Internet measurements and criminology to analyze not just one but several cybercrimes. Within these research activities, we leverage the technical work produced by other scholars and design new techniques ourselves. The five studies that make up this dissertation are explained in further detail in the following paragraphs. Due to the aim of our study, this research demands different methods for our analyses and is multi-

disciplinary at its core. Consequently, we utilize both quantitative and qualitative methods throughout this work. As this is a paper-based dissertation, the research methodology will be individually elaborated upon in each chapter.

1.4. DISSERTATION OUTLINE

This section outlines the remainder of this dissertation. It describes the five studies with their corresponding research questions and presents an overview of the peer-reviewed papers associated with each study and chapter. Table 1.2 lists this overview, including the names of the collaborating researchers.

STUDY 1 – ESTIMATING CRYPTOJACKING PREVALENCE ON THE WEB

Illicit cryptomining on websites has skyrocketed since 2017 and requires innovative methods to study its prevalence. In this first study, we designed a large-scale socio-technical measurement to accurately estimate the prevalence of this new type of cybercrime. Two extensive Internet measurements were performed to examine Web-based cryptojacking – one focused on organized cryptomining and one on the prevalence of cryptojacking on the entire Internet. We use a combination of data sources to build a representative dataset for our measurements and leverage previous academic work to crawl numerous domains and search for overlapping resources used in cryptojacking campaigns. We identify the commonly used attack vector for cryptojacking, the most prevalent category of websites affected, and stumble upon large cybercriminal campaigns.

In short, this study aims to answer the following research question:

> **RQ1:** *What is the prevalence of (organized) cryptojacking on websites, and what tactics, techniques, and procedures are used to deploy this?*

STUDY 2 – EXAMINING CRYPTOJACKING ON COMPROMISED INFRASTRUCTURE

During the first study, it became clear that cryptojacking is not only present on websites but that certain threat actors also installed malicious cryptojacking scripts on compromised Internet infrastructure, such as ISP and consumer-grade routers. Through a firmware vulnerability in MikroTik routers, cybercriminals could rewrite Web traffic flowing through these routers and inject it with malicious cryptojacking code. Following this attack vector revealed a completely new realm of cryptojacking activity, much larger and more organized than any website-based cryptojacking. Based on NetFlows recorded in a Tier 1 network, semiweekly Internet crawls, and network telescope traffic, we examine the modus operandi of these cybercriminals. We show how these large installations of cryptojacking attacks relate to each other and sketch the life cycles of infected infrastructure. Additionally, we perform campaign analysis to identify different levels of sophistication among adversaries.

In short, this study aims to answer the following research question:

> **RQ2:** *How do cybercriminals set up large-scale cryptojacking campaigns on Internet infrastructure?*

STUDY 3 – INVESTIGATING THE DUTCH PHISHING LANDSCAPE

Another cybercrime we study is phishing, which is the illicit harvesting of user credentials. Although many phishing studies have been published [58, 119, 139, 241, 262], an in-depth analysis of the cybercriminal TTP targeted at Dutch citizens was lacking. For this research, we specifically focused on the first component in the phishing value chain: development. This component involves developing and trading phishing kits used in phishing attacks. We collect many such phishing kits, leverage the use of TLS certificates by phishers, and develop a crawler to uncover phishing domains aimed at customers of the Dutch financial sector. This allowed us to capture the end-to-end life cycle of such phishing attacks and investigate the luring capabilities to trick victims into disclosing their credentials. These insights enabled us to paint a comprehensive picture of the TTP prevalent in the Dutch phishing landscape and present public policy takeaways for anti-phishing initiatives.

In short, this study aims to answer the following research question:

> **RQ3:** *What is the role of phishing kits within the Dutch phishing ecosystem?*

STUDY 4 – SCRUTINIZING INTERNET ABUSE HANDLING IN THE WILD

Besides law enforcement actions, various other instruments have been put into place to fight Internet abuse (e.g., cybercrime). These instruments range from legislation to remove copyrighted material to technical instruments such as blocklists to stop spam. They rely on industry standards for abuse reporting: reporting abuse to the resource or network owners requesting mitigation. Through a unique collaboration with law enforcement in the Netherlands, we were granted access to the operational back-end of a hosting provider with a reputation for abuse. We examine its abuse tickets and their follow-up actions to investigate what mechanisms in the anti-abuse ecosystem influence anti-abuse actions. Such insights can, in turn, inform policymakers and law enforcement agencies to align governance repertoire with abuse handling in practice.

In short, this study aims to answer the following research question:

> **RQ4:** *What mechanisms in the anti-abuse ecosystem influence anti-abuse actions?*

STUDY 5 – INTERNET MEASUREMENTS AND CYBERCRIME POLICING

As stated in § 1.2, measuring cybercrime has been the subject of many scientific publications in the past. These quantifications are typically based on large-scale Internet measurements of resources such as domains, binaries, or attack traffic. We leverage the concept of value chains to structure 38 studies featuring measurements of phishing, booter services, and remote access trojans (RATs). We review how other scholars have set up their measurements of these profit-driven cybercrimes, examine their methods, and assess the data sources they used to identify common denominators. Then, we let law enforcement professionals reflect on some of these measurements and jointly identify the unexplored potential for novel measurements that could solve current needs in cybercrime policing. Combining the input from law enforcement with the structured analysis of previous academic measurements allowed the identification of what charac-

teristics of scientific measurements are useful for law enforcement – i.e., assisting LEAs in determining how their scarce resources can be best put to use.

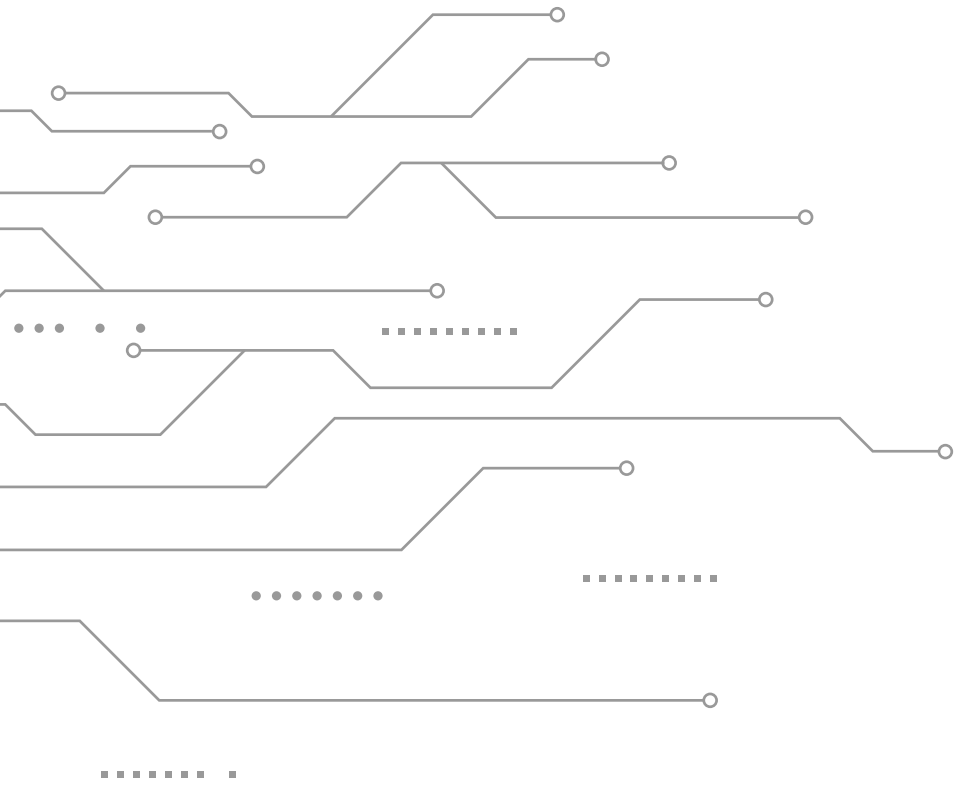
In short, this study aims to answer the following research question:

> **RQ5:** *What are the characteristics of cybercrime measurements that could assist law enforcement in improved cybercrime policing?*

Table 1.2 lists an overview of the chapters in the remainder of this dissertation and the peer-reviewed studies that they are based on. Chapter 7 completes this dissertation with a summary of the main findings, a reflection on the results, and directions for future research.

Table 1.2: Dissertation outline.

Chapter	Publication
Ch. 2	**Bijmans, H.L.J.**, Booij, T.M. & Doerr, C. (2019). "Inadvertently Making Cyber Criminals Rich: A Comprehensive Study of Cryptojacking Campaigns at Internet Scale". In *Proceedings of the 28th USENIX Security Symposium (USENIX Security '19)*. [16]
Ch. 3	**Bijmans, H.L.J.**, Booij, T.M. & Doerr, C. (2019). "Just the Tip of the Iceberg: Internet-Scale Exploitation of Routers for Cryptojacking". In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS 2019)*. [17]
Ch. 4	**Bijmans, H.L.J.**, Booij, T.M., Schwedersky, A., Nedgabat, A. & van Wegberg, R.S. (2021). "Catching Phishers By Their Bait: Investigating the Dutch Phishing Landscape through Phishing Kit Detection". In *Proceedings of the 30th USENIX Security Symposium (USENIX Security '21)*. [15]
Ch. 5	**Bijmans, H.L.J.**, van Eeten, M.J.G. & van Wegberg, R.S. (2025). "Tickets to Hide: Scrutinizing the Anti-Abuse Ecosystem with Internal Abuse Data". *Under submission at NDSS'26*
Ch. 6	**Bijmans, H.L.J.**, van Eeten, M.J.G. & van Wegberg, R.S. (2025). "A Measured Response – On the Nexus of Large-Scale Technical Measurements and Cybercrime Policing". In *Proceedings of the 24th Workshop on the Economics of Information Security (WEIS '25)*. [18]



2

ESTIMATING CRYPTOJACKING PREVALENCE ON THE WEB

Since the release of a browser-based cryptominer by Coinhive in 2017, the easy use of these miners has skyrocketed illicit cryptomining in 2017 and continued in 2018. This method of monetizing websites attracted website owners, as well as criminals, seeking new ways to earn a profit. In this chapter, we perform two large studies into the world of cryptojacking, focused on organized cryptomining and the spread of cryptojacking on the Internet. We have identified 204 cryptojacking campaigns, an order of magnitude more than previous work, which indicates that these campaigns are heavily underestimated by previous studies. We discovered that criminals have chosen third-party software – such as WordPress – as their new method for spreading cryptojacking infections efficiently. With a novel method of using NetFlow data, we estimated the popularity of mining applications, which showed that while Coinhive has a larger installation base, CoinImp WebSocket proxies were digesting significantly more traffic in the second half of 2018. After crawling a random sample of 49M domains, ~20% of the Internet, we conclude that cryptojacking is present on 0.011% of all domains and that adult content is the most prevalent category of websites affected.

This chapter has been published as: **Bijmans, H.L.J.**, Booij, T.M. & Doerr, C. (2019). "Inadvertently Making Cyber Criminals Rich: A Comprehensive Study of Cryptojacking Campaigns at Internet Scale". In *Proceedings of the 28th USENIX Security Symposium (USENIX Security '19)*.

2.1. INTRODUCTION

Unlike traditional currencies, such as the Euro or Dollar, cryptocurrencies are digital assets created as a medium of exchange based on cryptography and a blockchain, which are used to secure both the creation and transactions of units. In 2009, Satoshi Nakamoto released Bitcoin [173], the first ever decentralized cryptocurrency, which made it possible to transfer monetary value to another person by creating a transaction and committing this to the blockchain, a list of blocks secured by cryptographic operations maintained by a peer-to-peer network of miners. These miners secure the blockchain by constantly collecting transaction data from the network and validating it by solving cryptographic challenges based on the previous block, the transaction, and the receiver of the transaction. After validation, the confirmed transaction is inserted into the blockchain again in the form of a validated block. As a reward, the miner gets cryptocurrency. This network guarantees that only the rightful owner of a Bitcoin wallet can make transactions and prevents malicious actors from inserting false information into the blockchain.

Solving these cryptographic challenges as a miner has, however, become so difficult that Bitcoin cannot efficiently be mined anymore on regular PCs. Over the past years, over 4,000 other cryptocurrencies have been created, so-called altcoins. One of them is Monero (XMR), launched in 2014 and nowadays the most popular cryptocurrency in browser-based mining [184]. In contrast to Bitcoin, Monero uses a private blockchain, meaning that while anybody can use it to make transactions, nobody is allowed to view them [236]. It also builds upon a different proof-of-work algorithm to validate its transactions, called CryptoNight, a fork of CryptoNote [209]. This algorithm is designed to be memory-hard and, therefore, requires a large set of bytes in memory to perform frequent read and write operations. Simple consumer-grade CPUs have exactly that memory available at their processor caches, making this kind of mining the most efficient on regular consumer-grade hardware. To speed up the mining process, mining jobs can be distributed among individual miners in a mining pool. In such a pool, miners work together to mine new blocks and share the rewards. Work is distributed among miners in the pool based on the difficulty of the cryptographic challenge. Consequently, powerful machines solve the more difficult puzzles, while low-end machines receive the easier ones. Rewards are shared according to the same principle. Mining pools closely monitor the submissions from their miners and state that they will block any wallet address after receiving evidence that a wallet is used for malware or botnet activities [165].

The introduction of alt-coins that, by design, can be effectively mined on regular PCs also made them an attractive target for cybercriminals. Both the private blockchain and the ASIC-resistant mining algorithm of Monero quickly made Monero one of the preferred choices. In addition to being included in malware [190], there also exist implementations to perform *drive-by mining* or *cryptojacking*, where cryptocurrency is mined in the user's web browser while visiting a website. While originally developed as an alternative mechanism to donate to the upkeep of a website in presence of now ubiquitous ad-blockers, many methods exist to maliciously apply browser-based mining: for example, criminals hack vulnerable websites to install mining scripts [34] or create malicious advertisements with cryptojacking code that are displayed on benign websites [169], but actors have also compromised routers [187] or setup malicious Wi-Fi networks [191] to inject cryptominers into their users' traffic.

Previous studies have performed surveys on the use of cryptominers across the most commonly visited websites and have identified groups of criminals installing cryptominers on a large number of domains for their own profit [127, 199]. It makes sense for a cybercriminal to lure as many users as possible into such mining, which could be accomplished not only by deploying the cryptojacking code into popular websites but also by hacking a large number of websites or injecting a resource such as a common library that is used by a large number of unsuspecting websites. These individual installations are working together in a coordinated campaign, thus significantly increasing the profits of the criminal, but at the same time also indicating an elevated level of knowledge and sophistication of the adversary. The presence and extent of such coordination are, however, largely unknown.

In this chapter, we address this gap and systematically investigate the coordination and collaboration of cryptojackers on the Internet and make the following four contributions:

- We are the first to systematically analyze the relationships between websites that perform cryptomining and the actors behind them. By this campaign analysis, we find the existence of massive installations. In fact, we have identified 3 times as much cryptojacking activity as Rauchberger et al. [199], and the five largest campaigns we detected exceed the *total* size of cryptomining reported in Konoth et al. [127].
- We show that the bulk of organized mining activity is the result of compromised (parts of) third-party software and that comparatively little organized activity is the result of hacked websites or an explicit choice to mine by the website owner.
- Through a survey of 1,136 top-level domains and by comparing the installation base with actual mining traffic on the Internet using NetFlow data, we find that the most prominently installed miner is actually not the one that generates the most mining activity in practice. We also see that applications and attack vectors come and go and that different TLD zones exhibit clear differences in mining application popularity.
- Estimating cryptojacking by solely crawling the Alexa Top 1M results is an overestimation of the size, as we see that cryptojacking activity is almost 6 times higher in that subset compared to the rest of the Internet.

To enable follow-up research, we make our data and software publicly available at `https://www.cyber-threat-intelligence.com/cryptojacking-campaigns`.

2.2. BACKGROUND

WEBASSEMBLY & ASM.JS

To enable faster execution of code inside the browser, Mozilla developed *asm.js*, a technique for translating high-level languages, such as C and C++, into JavaScript to be used by the browser [168]. Multiple validation methods enable the JavaScript engine to compile this code ahead of time and improve execution speed. This technique made it possible to execute code faster inside the browser after its release in 2013.

WebAssembly (*Wasm*) is a more recently released scripting language developed by the World Wide Web consortium in 2017 and is able to compile high-level languages like C, C++, and Rust inside the browser to be used in web applications [251]. It runs in a sandbox within the browser, and it aims to execute as fast as native machine code. Wasm is complementary to JavaScript, as it is controlled by JavaScript code after its compilation. The difference between asm.js and Wasm is the fact that the latter is compiled only once and is started directly at native speed. In contrast, code in asm.js is compiled and optimized at run time, therefore decreasing execution speed. Both techniques are supported by all four major browsers (Chrome, Firefox, Edge and Safari) and have drastically improved the execution speed of applications inside the browser, which made them very attractive for browser-based mining.

WEBSOCKETS & STRATUM

WebSockets is an HTML5 protocol providing two-way communication between the client and a server over a single TCP connection [253]. The protocol enables easy real-time data transfer without refreshing (a part of) the web page. Communication is done over the same TCP ports as the web browser, making it robust to strict firewall rules or other blocking. Developers are free to define the format of messages sent over WebSocket connections. However, there is a protocol specifically designed for cryptomining communications: the Stratum Mining Protocol, a line-based protocol with messages encoded in plain-text JSON-RPC format [226]. Servers communicate with their clients using Stratum to authorize new miners in the pool, distribute jobs based on difficulty and retrieve found hashes from the miners. An example of a WebSocket connection using the Stratum protocol is given in Table 2.1.

BROWSER-BASED MINING

Triggered by the rise of CPU-mineable cryptocurrencies (such as Monero) and the rapid development of useful web standards (e.g., WebAssembly and the Stratum protocol), browser-based cryptomining gained enormous momentum in the autumn of 2017. Coinhive, a German company, created an easy-to-use browser-based mining application as an alternative to advertisements [43, 136]. They provide a JavaScript library, an API, and a WebSocket proxy infrastructure to developers to easily integrate a browser-based miner into their website and let their visitors mine for Monero. 70% of the mined Monero is transferred to the owner of the account, and the remaining 30% is kept by Coinhive [44]. Soon after Coinhive released their miner application, similar ones appeared, such as Cryptoloot [50] and Coin-Have [42]. Nowadays, miner applications come and go, with various capabilities and usage fees, but Coinhive still has a prominent place in the cryptojacking landscape.

OVERVIEW OF A CRYPTOJACKING ATTACK

Although different mining applications exist, all browser-based miners show great similarities. As depicted in Figure 2.1, the user visits the cryptomining website (1) and receives a valid HTTP response (2). The cryptomining website requests a JavaScript file (3), which controls the mining operation. This script first explores the host system, searches for the number of CPU threads available, downloads the WebAssembly mining script for the actual mining operation (4), and distributes it over a number of Web-

WebSocket traffic frames
⇑ {"type":"auth", "params":{"site_key":"<site_key_of_website>", "type":"anonymous","user":null,"goal":0, "version":3000,"coin":"xmr"}}
⇓ {"type":"authed", "params":{"token":"<random_36_characters>", "hashes":0}}
⇓ {"type":"job", "params":{"blob":"<random_152_characters>", "job_id":"<random_28_characters>", "target":"ffffff01", "id":"<random_36_characters>", "algo":"cn","variant":"4","height":1808537}}
⇑ {"type":"submit", "params":{"job_id":"<random_28_characters>", "nonce":"377c32b8", "result":"<found_64_characters_hash>"}}
⇓ {"type":"hash_accepted", "params":{"hashes":128}}
⇓ {"type":"job", "params":{"blob":"<random_152_characters>", "job_id":"<random_28_characters>", "target":"ffffff01", "id":"<random_36_characters>", "algo":"cn","variant":"4","height":1808537}}

Table 2.1: Example of a WebSocket connection using the Stratum Mining Protocol to communicate with a mining pool.

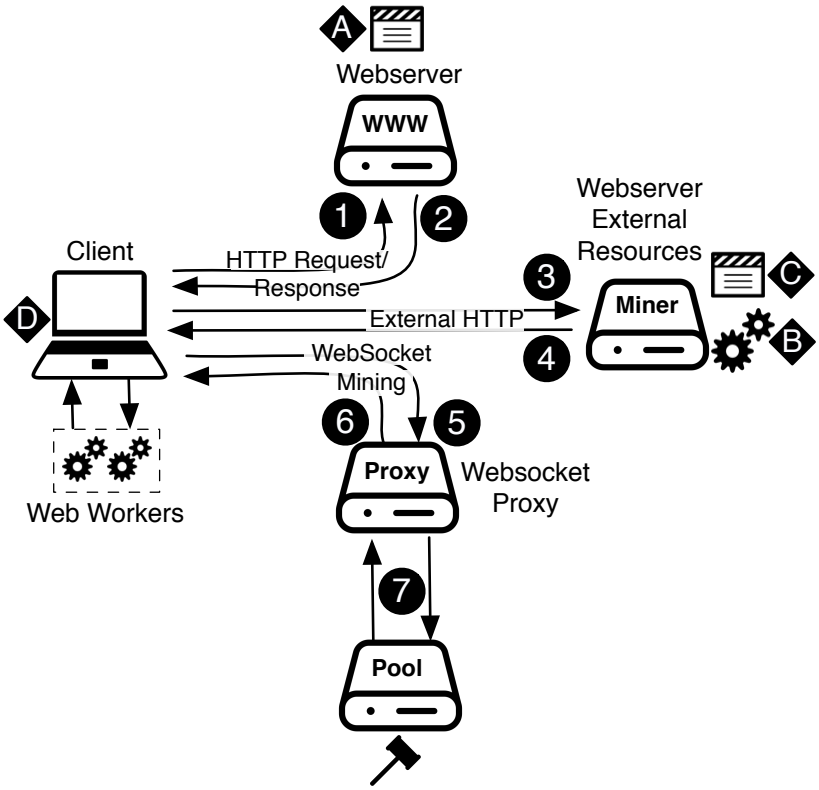


Figure 2.1: Browser-based cryptomining attack.

Workers (a JavaScript instance running in the background, without affecting the page performance). It also creates a WebSocket connection with the mining pool through a proxy (5). The script authenticates itself to the mining pool server (in Stratum format) and, if successful, receives the first job to work on (6). The WebWorkers start working on that job and find hashes that are submitted to the mining pool by the controller script (7).

CAMPAIGN ANALYSIS

Campaign analysis is the field of research focused on discovering clusters of malicious online entities. The term originates from the analysis of large volumes of spam or phishing emails but can also be used in other areas, such as browser-based cryptomining. In this particular case, campaign analysis is focused on finding clusters of the same cryptominers on different domains. Since those miners always include a form of identification to which funds need to be transferred, clustering cryptojacking websites can be done relatively easily. Most mining applications define a *siteKey*, a unique (random) string used to identify the user to which earnings have to be transferred, which can be found in either the source code or the WebSocket traffic. A similar *siteKey* guarantees

that the same account is rewarded for the mining that takes place. Identifying campaigns can also be done by searching for similar WebSocket proxy servers if the website is not using a popular one but instead hosting its own server. We have used these and other techniques to discover campaigns as discussed in § 2.6.1. We have chosen to define a cluster of websites as a campaign once they share identical features more than 5 times. E.g. a cluster of 6 websites with the same *siteKey* or private WebSocket proxy server is considered a campaign.

2.3. ATTACK VECTORS

Mining cryptocurrencies with the computing power of website visitors is not illegal as long as users are asked permission to mine. When a user cannot consent to the mining activities their computer is involved in, it is called *cryptojacking*. Although browser-based cryptomining is a recent phenomenon, a legal precedent on cryptomining without consent already exists. In 2015, a US court settled a case with a developer of Bitcoin-mining software, in which the Attorney General stated that no website should tap into a person's computer processing power and that the user has to be informed about the cryptomining activities that take place on the visited website [100]. However, this is often not the case. In this section, we summarize the attack surface for cryptojacking on the Internet. All attack vectors are marked in Figure 2.1 by their corresponding characters.

WEBSITE OWNER (A)

The owner of a website can add a cryptomining script to their web page without informing their users. This can be done as a replacement for advertisements, which was the case for The Pirate Bay, one of the most popular torrent websites [237]. Only a few days after the Coinhive service was launched, they added a miner to their website, which started mining without user consent, as a replacement for the intrusive advertisements they would normally show. Nowadays, the website shows a disclaimer at the bottom of the homepage, notifying visitors that their CPU will be used for cryptomining. Another major source of website owner initiated cryptojacking is parked domains [63].

COMPROMISED WEBSITES (A)

A cryptomining script can also be present on a web page without the knowledge of the website owner. When a website gets hacked, an attacker is able to inject cryptomining scripts. Now, the attacker receives the rewards for the visitors mining on that website. There are numerous examples of this kind of attack. There have been cryptojacking scripts found on web pages of the Indian government [34], CBS Showtime [157], and many others.

THIRD-PARTY SOFTWARE (B)

Gaining unsolicited access to a large number of domains is a time-consuming operation. As a consequence, attackers have tried different tactics to infect multiple websites at once by infecting third-party software. In the last year, we have seen attacks in which cryptojacking code is injected into popular third-party software, such as JQuery or Google Tag Manager [40]. Drupal, a widely used open-source CMS, was the victim

of a large attack involving more than 100,000 websites [171], and WordPress, a similar CMS, suffered from a weather plugin [255] secretly injecting a cryptojacking script into the website it was installed on.

MALICIOUS ADVERTISEMENTS (C)
Advertisement-supported websites let their advertisements be sold by advertisement networks, such as Google. The downside of this system is that attackers can attach cryptomining scripts to advertisements and distribute them through an advertisement network over a large number of websites. In January 2018, YouTube was a victim of this kind of attack, in which cryptomining scripts were injected into the ads shown on the website [169].

MAN IN THE MIDDLE (D)
The most effective method of gaining a large group of miners for an attacker is by being the man-in-the-middle. In August 2018, 200,000 MikroTik routers were infected by malware, which inserted a Coinhive script into every website the user visits [187]. The bug was patched within a day, but many MikroTik routers are not updated, leaving them still vulnerable. In our research, we are not able to detect these attacks since they are not originating from a website.

2.4. RELATED WORK

Academic research on browser-based cryptomining has only started in 2017 and is, due to the recent developments of the used web standards, very topical. The first explorations into this research field have been performed by Eskandari et al. [63]. In their analysis, the authors queried two large source code datasets for strings known to be part of cryptomining scripts (such as `coinhive.min.js` or `load.jsecoin.com`) and found a large number of domains. This method is only able to detect known mining applications, not the obfuscated or new ones. While calculating the profitability, the authors stumbled upon a Coinhive campaign that ran a miner on over 11,000 parked websites. This study kicked off a number of subsequent investigations, which were all aimed at detecting browser-based cryptomining. Rauchberger et al. [199] created their *MiningHunter*, a crawler able to detect mining scripts even when their malicious activities are obfuscated. The detection method relied on analyzing executed JavaScript code and WebSocket traffic frames. After a successful crawl of the Alexa Top 1M at the beginning of December 2017, they were able to detect 3,178 websites running a cryptominer. 1,210 unique keys were retrieved, and one large campaign involving 1,116 websites infected by a malicious advertisement network was identified. At the same time, Parra Rodriguez et al. [205] worked on *RAPID*, a resource and API-based detection method, which is able to detect browser-based cryptomining and is resistant to JavaScript obfuscation. Their classification was able to classify mining samples with a precision of 96%. Eventually, 656 actively mining websites were found in the Alexa Top 330,550. A similar classification study was performed by Carlin et al. [29], in which they demonstrated that dynamic opcode tracing is extremely effective at detecting cryptomining behavior. Liu et al. [149] proposed a novel approach for detecting browser-based mining applications by creating *BMDetector*, a detection system based on a modified Chrome kernel. Using

this modified kernel, the authors were able to perform JavaScript code block analysis on the compiled JavaScript code, which allowed them to detect heavily obfuscated miner applications as well. Hong et al. [101] built *CMTracker*, a behavior-based detector with two runtime profilers for tracking browser-based cryptomining. The first profiler monitors incoming JavaScript files for known fingerprints, and the second profiler observes the call stack and searches for periodic executions. Their approach was able to detect 868 actively mining websites among the Alexa Top 100K in April 2018. More than half of the keys found were used only once, and they noticed that domains hosting mining scripts were migrating faster than the mining pool domains. The authors also mentioned evasion techniques, such as code obfuscation and payload hiding inside third-party libraries. Periodic execution in mining scripts was also noticed by Wang et al. [250], who created *SEISMIC*, a monitoring service to interrupt browser-based mining scripts based on this finding.

A different view on the subject was given by Papadopoulos et al. [189], who tried to answer the question of whether browser-based cryptomining could be a suitable alternative to advertisements. After crawling a dataset of 200K websites running advertisements or cryptominers, they concluded that advertisements are still more than 5 times more profitable than cryptominers. This will only change once a visitor stays on the same website for more than 5.3 minutes or when Monero becomes more valuable [189]. A broader view of the browser-based cryptomining ecosystem is given by Saad et al. [208], who researched both cryptomining code and user impact. Besides various JavaScript static code analysis clustering methods and battery drainage studies when cryptomining, they did not perform any crawling of the web. This is in great contrast to the work of Rüth et al. [207], who dug deep into browser-based cryptomining by conducting two large web crawls. A first crawl using `zgrab`, which downloaded the first 256 kB of 137M *.com*, *.net*, and *.org* domains, as well as from the Alexa Top 1M websites. Consequently, the resulting HTML file was checked against the NoCoin [84] block list. A second crawl was performed on a subset of 10M websites, with a customized Chrome browser, instructed to dump WebAssembly modules for further inspection. They conclude their work by stating that 0.08% of the probed websites are actively mining [207].

Another large web crawl study is conducted by Konoth et al. [127] as a study for the creation of *MineSweeper*. Again, the Alexa Top 1M (including three internal pages) was crawled, with a crawler extracting information from all loaded JavaScript and HTML files, WebSocket traffic, and requests made while visiting the website. A total of 1,735 websites were found to be actively mining, the majority of them using Coinhive. 20 mining campaigns were discovered in their analysis, of which the largest involved 139 websites. Based on these findings, a novel detection technique was developed, which focused on the aspects all mining scripts have in common: high CPU cache usage and WebAssembly. They developed *MineSweeper*, able to successfully identify mining scripts based on the CPU's L1 and L3 cache usage and cryptomining characteristics in WebAssembly, thus hardening it against miner obfuscation.

As shown by this summary of related work, most attention of academic investigation has been on detecting these browser-based cryptominers. Multiple studies have shown to be able to detect them with high precision [101, 127, 149, 199, 205, 207]. Academic research is less focused on finding campaigns of cryptomining websites, while the online

Table 2.2: All search queries for the PublicWWW database.

Miner	Search term(s)
Coinhive	coinhive.min.js | CoinHive.Anonymous(
JSECoin	load.jsecoin.com
Webmine	webmine.cz
Cryptoloot	/crypta.js | /crlt.js | crlt.anonymous | CryptoLoot.Anonymous
CoinImp	CoinImp.Anonymous hashing.win | hostingcloud.racing
Cryptonoter	minercry.pt/processor.js | cryptonoter
NFWebminer	nfwebminer.com/lib/ | NFMiner(
Deepminer	deepMiner
Monerise	monerise_builder | monerise_payment_address(
Coinhave	minescripts.info
Nebula	CoinNebula.Instance
Mineralt	play.gramombird.com/app.js
Munero	munero.me
Minr	cdn.jquery-uim.download cnt.statistic.date | ad.g-content.bid
Webminerpool	webmr.js
WPMoneroMiner	wp-monero-miner.js
Nerohut	nhm.min.js | nerohut.com/srv
Adless	adless.js
Monero-mining	Perfektstart(
Miscellaneous	function echostat(){var | jquory.js function printju | startMining(pocketgolf.host/start.php async

research community (such as Badpackets [170] or Krebs on Security [136]) is particularly interested in finding those relations. The first explorations into this area have been taken by [127], [63], and [199], but campaigns have not been systematically explored in their research. This paper aims to resolve this gap by focusing on identifying campaigns, methods used in these campaigns, and their evolution. We are also interested in the spread of cryptojacking on the Web, but as previous work is mostly crawling (subsets of) the Alexa Top 1M, we will analyze a broader set of websites online. In this chapter, we will not try to create a new detection method, but we will build upon the work of [127] to perform our crawls.

2.5. METHODOLOGY

In a measurement study like this, suitable datasets and methods are essential for conducting proper research. In this section, we first discuss the datasets used or created, followed by a summary of our crawler implementation.

2.5.1. DATASET CREATION

In our first crawl, we focus on finding campaigns of cryptojacking websites. Previous work of [101, 127, 199, 207] mainly investigated the popular parts of the Internet by crawling the Alexa Top 1M, or subsets of it. But, as pointed out by Scheitle et al., the Alexa Top 1M is not the only list measuring the popular Internet, and the method Alexa uses to create this list raises questions about whether it is the most reliable list to use for research on cryptojacking [213]. To overcome this issue, we have decided to use the union of three top lists on the Internet: the Alexa Top 1M [3], the Cisco Umbrella 1M [39], and the Majestic 1M [154], all using different measurement strategies, to include the popular part of the Internet in our dataset. These last two also include subdomains and domains that do not serve a web page. Therefore, we have only added the domains to the list of URLs to be crawled and omitted the subdomains from the latter two. Since we are interested in finding as many cryptojacking domains as possible for our campaign analysis, we have decided to extend our list even further with a list of websites gathered from querying PublicWWW – a source code search engine – with the keywords listed in Table 2.2. The union of these sets formed the dataset to be crawled and consisted of 1,896,503 websites (unique effective TLDs + 1), as listed in Table 2.3. To estimate the prevalence of cryptojacking on the Internet in general, we will not use a top list as the Alexa Top 1M, because it is not a random sample of the Internet. We, therefore, also download a random sample of ~20% of the websites in 1,136 TLDs. We discuss this crawl in more detail in § 2.7.

OPERATOR NETFLOWS

While the aforementioned datasets provide insights into the landscape of cryptomining installations at a given moment, these data sources do not reveal much about the actual usage of such services. In order to bridge this gap, we analyzed NetFlow traces from the network of a Tier 1 operator from September 2017 until December 2018, which were collected at a 1:8192 sampling ratio. For our analysis, we obtained NetFlow records for all traffic from and to the various WebSocket proxy servers belonging to the mining services. Although NetFlows do not reveal the actual contents of a connection, the used ports and packet sizes can indicate connection types. The identity of the source connecting to the WebSocket proxy is, however, irrelevant and was anonymized to a pseudo-random value by the operator using the CryptoPan algorithm [257]. Additionally, the data access was cleared by the institutional review board. The research team did not obtain direct access to the NetFlow data containing source and destination IP addresses as personally identifiable information, but instead provided a list of IP addresses of cryptomining proxies and mining pools to the data owner, based on which the corresponding flow records were provided with the connection's source IP protected by a salted hash.

2.5.2. CRAWLER IMPLEMENTATION

As mentioned in § 2.4, this research builds upon the work of Konoth et al.[127]. Therefore, we have used their crawler implementation as a starting point for our crawler. The following paragraphs will highlight the major changes and additions made to their work for our research.

Table 2.3: Dataset creation for the campaign-focused crawl.

List	No. of websites	Date (2018)
Alexa Top 1M	1,000,000	Dec 24
Cisco Umbrella 1M	233,145	Dec 24
Majestic 1M	897,767	Dec 24
Custom PublicWWW set	87,051	Nov 23 – Dec 24
Total	**1,896,503**	

Table 2.4: The added miner applications and their keywords.

Miner	Keywords
Nebula	CoinNebula.Instance
WP Monero miner	wp_js_options | wp-monero-miner
Nerohut	nhm.min.js | NHpwd | nhsrv.cf/srv/serve.php?key=
Webminerpool	webmr.js | startMining(
Minero	minero.cc
Adless	adless.js | adless.io
Monero-mining	PerfektStart | perfekt.js
ProjectPoi	ProjectPoi\b | projectpoi.min.js
Papoto	papoto

ADDITION OF NEW MINER APPLICATIONS

The publicly available *Minesweeper* crawler supports 22 different mining applications. Based on previous work and online research, we have added another 9 miner applications to the crawler, in order to also identify the newest miner applications. The added applications and their keywords are listed in Table 2.4. For some of the already supported miner applications, we extended the fingerprints and improved the regular expressions to find *siteKeys*.

ACTIVE MINING DETECTION

We have instructed the crawler to never explicitly consent to any mining operation. Therefore, we define that website to be actively mining without consent when a mining code signature is found, together with a *siteKey*, more than two WebWorkers and a WebSocket connection, or when the Stratum protocol communication or login credentials for a mining pool are found in WebSocket traffic. If one of these conditions holds, we mark the domain as actively cryptojacking.

WEBSOCKET STACK TRACE

The miner application communicates with the mining pool using WebSocket connections. WebSocket traffic was already logged in the crawler, but the initiator of the WebSocket connection was not. By inspecting the stack trace of the WebSocket initiation, we can determine which script was responsible for opening the WebSocket connection

and, therefore, the mining initiator. Using this method, we can easily distinguish between miners started from the main HTML page or those hidden inside other resources. Moreover, similar stack traces are a powerful indicator for campaign analysis since they show what component started the mining application. We have used this method successfully in our campaign analysis to identify attack vectors. Miners hidden inside third-party software such as WordPress are easily noticed in the stack trace, as we will show in § 2.6.1.

CHANGED LOGIC AND EXHAUSTIVE KEY FINDING

Our crawler visits every website twice. First, by using a custom Chrome build, with the `-dump-wasm-module` flag enabled to dump any WebAssembly on the page. If present, these Wasm modules are analyzed for cryptojacking code by the *MineSweeper* application. Second, by using another Chrome build, which visits the website and saves every file it encounters. Instead of visiting 3 internal pages (as Konoth et al. [127] did), we instructed the crawler to visit just one internal page. Besides that, we have implemented a more exhaustive *siteKey* search. The crawler first searches for fingerprints of known miner applications and afterward for the *siteKey* in the following order: WebSocket traffic, the HTML page, and finally, all other HTML and JavaScript resources. A minor addition has been made to automatically decode a base64 encoded *siteKey* of the Mineralt miner [163]. This addition allowed us to retrieve more *siteKeys*, which improves the campaign analysis afterward.

2.5.3. INFRASTRUCTURE

We deployed the crawler in Docker containers on 60 servers within the university network, each running 8 Docker instances in parallel. The crawl started on December 24, 2018, and was completed on January 9, 2019. In total, 1,769,183 websites have been successfully visited in this initial crawl. Afterward, we performed a second crawl using the same infrastructure, which we discuss in § 2.7.

2.6. CURRENT STATE OF CRYPTOJACKING CAMPAIGNS

We have identified 21,022 websites with traces of cryptomining activities, of which 10,100 websites are actively mining without the visitor's explicit consent. Only 648 of these websites are listed in the Alexa Top 1M. 22 different miner applications have been identified among the crawled websites, most of them running at least the Coinhive miner application (71%). Also, 509 websites are deploying multiple miners. For 323 websites, the used miner application could not be detected, which indicates heavily obfuscated or unknown miner applications. The results are summarized in Table 2.5.

Among the identified websites, 204 campaigns have been detected, of which the largest one covers 987 websites. This number of campaigns is a magnitude larger compared to previous work [127, 199]. We have identified the use of third-party software, such as Drupal and WordPress, to be the driving factor behind the largest cryptojacking campaigns.

Table 2.5: Summary of the results of the first crawl.

Crawling period	24/12/2018 – 9/1/2019
# websites crawled	1,769,183 (93%)
# potential cryptojacking websites	21,022
# active cryptojacking websites	10,100
# active miner applications	22
# websites with unknown miners	323
# cryptojacking campaigns identified	204
# websites in largest campaign	987
# websites in Alexa Top 1M	648 (0.065%)
# websites in Cisco Umbrella 1M	109 (0.047%)
# websites in Majestic 1M	506 (0.056%)

MINING WITH CONSENT

There are two mining applications focused on mining solely with visitor consent. First, JSEcoin, a mining service presenting itself as *"The future blockchain & ecosystem for e-commerce and digital advertising"*, allows website owners to let their users mine JSE tokens after explicit opt-in consent [118]. Another consent-focused mining application is AuthedMine, the opt-in version of Coinhive, introduced after adblockers started blocking Coinhive [46]. In our crawl, we have identified 2,477 websites using the JSEcoin miner and 227 websites using AuthedMine. None of the websites using AuthedMine opened a WebSocket connection, which indicates that no mining activity took place. 143 websites using JSEcoin did, however, open a WebSocket connection but never actually started mining. By analyzing the WebSocket traffic, we observed that in most cases, the WebSocket connection initiation was followed by two probes sent back and forth, waiting for the user to opt in. Since these mining applications did not start mining without the consent of the visitor, we have omitted them from our results.

IDENTIFIED DOMAINS IN TOP LISTS

Of the 10,100 domains identified as actively cryptojacking, only 925 were found in one of the three top lists. The Alexa Top 1M contains the most cryptojacking domains (648), meaning that 0.065% of the websites in the Alexa Top 1M are cryptojacking, slightly less than previous work [127, 207]. For both other lists, this number is lower. The addition of the Cisco Umbrella 1M resulted in only 27 additional findings, whereas the addition of the Majestic 1M led to the discovery of 397 new cryptojacking domains. In Figure 2.2, a Venn diagram depicts these differences in subsets. Only a small number of websites are shared among the Alexa Top 1M and the Majestic 1M. Also, note that 9,175 (86%) of the identified websites are not listed in any of these top lists. This finding stresses the necessity of looking further than top lists while performing campaign analysis and studying the prevalence of cryptojacking on the Internet.

CATEGORIZATION OF WEBSITES

We have discovered various sorts of cryptojacking websites on the Internet. By complementing the list of identified domains with website categorization data of Webshrin-

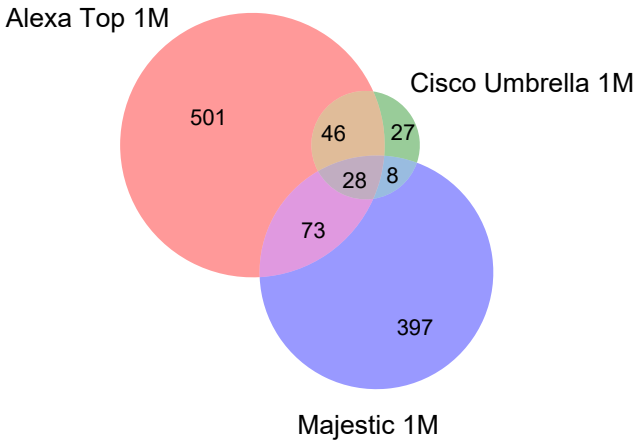


Figure 2.2: Venn-diagram showing the distribution of identified cryptojacking domains over the used top lists.

ker [252], we categorized each cryptojacking website. We confirm previous work by identifying adult content (such as pornography) as the most prevailing category within our dataset, with over 2,000 websites in this category. Illegal content, a category known for being home to abusive web resources, contains a lower number of cryptojacking websites compared to what we expected.

INSTALLATION BASE

Coinhive is still the most popular cryptomining application installed on the identified cryptojacking websites (75%), followed by Cryptoloot (5.3%) and CoinImp (3.2%). However, there are noticeable differences between the complete crawl and the subset of domains in the Alexa Top 1M. Coinhive's share is halved, whereas CoinImp and Cryptoloot installations are doubled in size. Nerohut and Webminerpool miners are relatively more present in the Alexa Top 1M subset, while Mineralt has a similar share in that subset. The bottom two stacked bars in Figure 2.3 show the distribution of miners according to our analysis. We have also discovered services that combine multiple cryptomining applications. The most popular mining combination is the set of Coinhive, Cryptoloot, and Cryptonoter, which are bundled in the implementation of the WordPress Monero Miner plugin [122]. A combination of a Nerohut miner with a Cryptoloot or Webminerpool miner is also regularly encountered. Usually, only one miner starts (due to another script deciding which one to use), but we also encountered domains on which multiple miners were started concurrently.

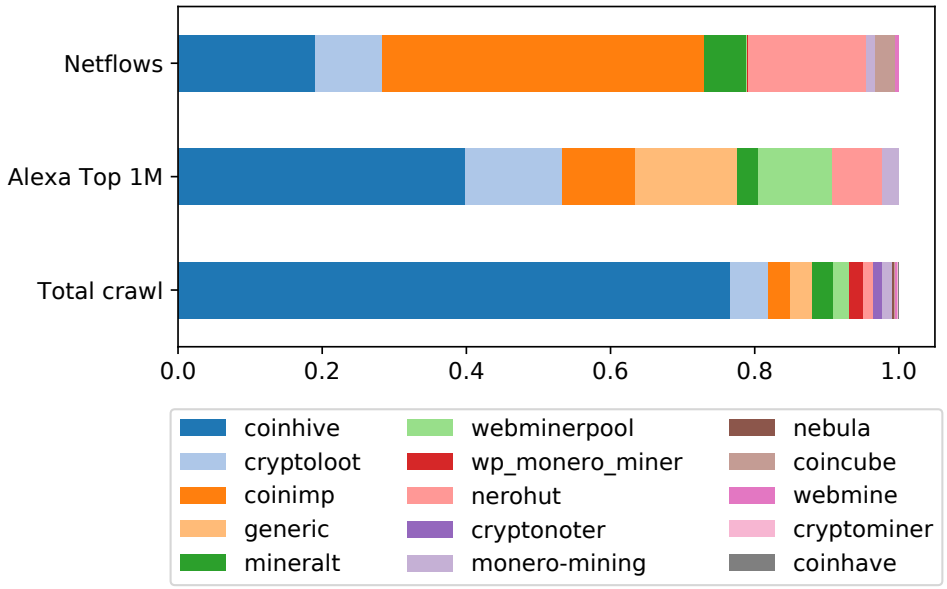


Figure 2.3: Distribution of cryptomining applications based on the total crawl, the Alexa Top 1M, and NetFlows analysis.

ACTUAL MINING ACTIVITY

The distribution of mining applications installed on domains gives an insight into their popularity by actors pursuing cryptomining but not into their actual usage. The amount of actual mining that takes place can, however, be estimated by tracing the connections website visitors make to the mining application's WebSocket proxy, as explained in Figure 2.1. We obtained a trace of connections transported by a Tier 1 network operator in 1:8192 sampling for a period of 14 months and followed the WebSocket proxy server IPs from these mining applications to estimate the traffic to these servers. This gives an insight into how much traffic these WebSocket proxies digest and is, therefore, a more reliable source for popularity measures. The upper stacked bar chart in Figure 2.3 shows the distribution of NetFlows to the WebSocket proxy servers of known mining applications for the month of December. The results show a drastic difference between the installation base and mining traffic. While Coinhive is found on most websites, CoinImp proxy servers handle more than twice as much traffic as the dominant application. WebSocket traffic to the servers of Cryptoloot is similar in size compared to its installation base.

MINING POOL PARTICIPATION

Most mining applications do not disclose the mining pool they are mining for in WebSocket traffic. However, on 135 identified domains, WebSocket traffic did reveal that, as listed in Table 2.6. Most of these websites are participating in the `supportxmr.com` mining pool, which is commonly orchestrated by a Webminerpool or Nerohut mining script. Other pools are less commonly used or were not revealed in WebSocket traffic.

Table 2.6: Mining pools the identified domains are mining in.

Mining pool	Occurrence
supportxmr.com	93
xmrpool.eu	15
greenpool.site	13
minexmr.com	6
xmr.omine.org	4
moneroocean.stream	2
seollar.me	1
xmr.nanopool.org	1

THROTTLING OF APPLICATIONS

Most cryptomining applications allow for a throttle value to be set, which limits the percentage of the CPU the miner can use. It is not necessary to set a throttle value, in this case the miner uses 100% of the available processing power. We have discovered that when a throttle value is set, this is often set to 0.3, meaning that 70% of the processing power can be used by the miner. Setting a throttle to use 70% of the resources seems to be balancing between gaining enough profit and not disturbing the browsing experience too much. In the identified campaigns, the throttle value is mostly set to the same value on all domains. An exception is listed in Table 2.7, in which a campaign involving 180 websites uses two different throttle values.

ATTACK VECTORS ENCOUNTERED

We were able to retrieve the *siteKey* of actively cryptomining websites in 92% of the cases. Most of the gathered *siteKeys* are only used once (78%), and only a small portion (5%) is used on more than 5 different websites. However, the *siteKeys* in this last category are found on 4,663 different websites (46% of the total). The high number of *siteKeys* used only once suggests a large amount of website owner initiated cryptojacking since every domain uses its own key. The fact that almost half of the websites are part of a campaign involving at least 5 websites also indicates different attack vectors. We have manually analyzed the used *siteKeys* in the latter category, and we can conclude that, besides website owner initiated cryptojacking, the use of third-party software is a prevailing attack vector. Third-party applications like WordPress, Drupal, or Magento are often abused to spread cryptojacking injections. These applications play a major part in campaign analysis, as discussed in § 2.6.1.

HIDING TECHNIQUES

With the rise of cryptomining blocking applications such as NoCoin [88] or Minerblock [87], mining scripts are more often hidden to prevent detection. We have encountered a number of hiding techniques in our crawl and distinguished the following levels of obfuscation:

1. *No obfuscation.* The script is loaded in clear text, key and other options are visible to the user.

```
var miner = new CoinHive.Anonymous('key');
miner.start();
```


2. *Limiting CPU usage.* Script is loaded in clear text, and key and other options are visible to the user, but CPU usage is throttled, so detection by the user is less likely.
3. *Renamed variables.* The script is loaded in clear text, but (some) variable names have been changed. These variable names are either replaced by random strings or by completely different words, such as on http://www.2001.com.ve/:

   ```
   startHarryPotter("boddington", "2001");
   ```

4. *Renamed mining script.* The loaded script is still in clear text but hosted on the web server itself instead of fetched from a mining service. The file name is changed to prevent blocklisting, frequently to general names, such as `jquery.js` or `stat.js`.
5. *Hidden inside other scripts.* The miner is appended or inserted into another script. The benign script still functions as normal, but also starts up the mining process.
6. *Obfuscated code.* The loaded scripts are masked by a code obfuscator and contain packed or CharCode code. All application-specific strings are encoded, stored in an array and variable names are replaced by random strings.

   ```
   var _0x5d02=["\x75\x73\x65\x20\x73\x74", ..]
   ```

7. *Obfuscated code and WebSocket traffic.* The loaded script is obfuscated by a code obfuscator, and WebSocket traffic is sent encrypted to the proxy server.
8. *Obfuscated and hidden.* Scripts are hidden inside other files and/or via multiple redirects. Every script is randomly named and obfuscated, and so is the WebSocket traffic. WebAssembly is not retrieved from the server, but included inside the script.

In our crawl, most website owner initiated cryptojacking is not obfuscated, often not even throttling CPU usage. Attacks using third-party software usually hide cryptomining code inside other scripts and apply some obfuscation. We have encountered multiple WordPress themes and Drupal plugins with such a hidden miner. Only 391 websites with encrypted WebSocket have been identified, whereas most websites are using plain text Stratum communication. The highest level of obfuscation is rarely encountered.

2.6.1. CRYPTOJACKING CAMPAIGNS

We have identified 204 cryptojacking campaigns covering 5,733 websites, meaning that 57% of all cryptojacking websites encountered are part of a campaign. We define a cluster of more than 5 websites to be a campaign, as stated in § 2.2. Figure 2.4 shows all the identified cryptojacking domains in a force-directed graph, where domains with similar features attract each other, colored according to the used application. Clear clusters can be distinguished, such as a Monero-Mining campaign shown in pink and a large Mineralt campaign shown in green right above it. Coinhive, the application used the most,

is shown in dark blue with multiple large clusters all over the graph. The circle represents the cryptojacking domains not part of a campaign. In the following paragraphs, we highlight our findings based on different possibilities for identifying campaigns as introduced in § 2.2.

FOUND ON SHARED SITEKEY

We were able to successfully retrieve the *siteKey* of 92% of the actively cryptojacking domains, which enabled us to cluster domains sharing the same *siteKey*. A shared *siteKey* guarantees that the rewards for mining will be transferred to the same account. We have identified 192 cryptojacking campaigns based on the same *siteKey* being installed on more than 5 different websites. As shown in Table 2.7, the largest campaign covers 987 websites, all using WordPress. A variety of plugins and themes include a malicious file named `jquory.js`, which is responsible for starting a Coinhive miner. A similar attack vector is observed in a campaign involving 317 Drupal websites. This campaign is part of the Drupalgeddon 2 and 3 attacks, which took advantage of major remote code execution vulnerabilities in Drupal to inject their malicious scripts [216]. The only large campaign using the Mineralt miner also focused on WordPress, has base64 encoded its *siteKey* inside the `script` tags. This makes them seem different, but match once decoded, since only the throttle value is changed. Not just vulnerabilities in CMS systems are used to spread cryptojacking code. Magento, an e-commerce system, is also involved in a Coinhive mining campaign targeting 175 websites in our crawl. The largest campaign using the compromised websites attack vector involved 376 Chinese websites, which share a miner script injected on the bottom of the page. A provider of The Pirate Bay proxies orchestrates the largest website owner initiated campaign on our list, with 70 proxy domains using the same Cryptoloot miner. These findings indicate that the most successful and largest cryptojacking campaigns are created by abusing third-party software.

FOUND ON SHARED WEBSOCKET PROXY SERVER

Most cryptojacking campaigns are using the infrastructure of popular applications, such as Coinhive, to connect to a mining pool. Thus, clustering domains on these WebSocket proxy servers will not create meaningful clusters. However, when we discard these popular proxy servers, we are able to identify another 12 campaigns, which have not already been identified by shared *siteKeys*. Those are listed in Table 2.8. A Coincube miner campaign involving 27 websites uses `coin-services.info` as a WebSocket proxy server on a variety of ports. This campaign hosts its miner scripts on code repositories such as GitHub and BitBucket, where a number of accounts is created to host the miner files, which are all named `main.js`. On one of the GitHub accounts, even a picture of stacked Ukrainian money can be found [85]. 28 very similar websites, all offering illegal video streams, were found to be using a WebSocket proxy server on `wss://ws**.1q2w3.life/proxy` with, after manual inspection, `seriesf.lv` as the accompanied *siteKey*. This proxy server was also discovered by [127] on 5 websites in their crawl. They estimated that this campaign made a profit of $2,012.90 per month, which is likely to be a lot more since we have found almost 6 times as many domains involved in this campaign. We have discovered that websites using a private WebSocket proxy are more likely to hide their activities by using higher levels of obfuscation.

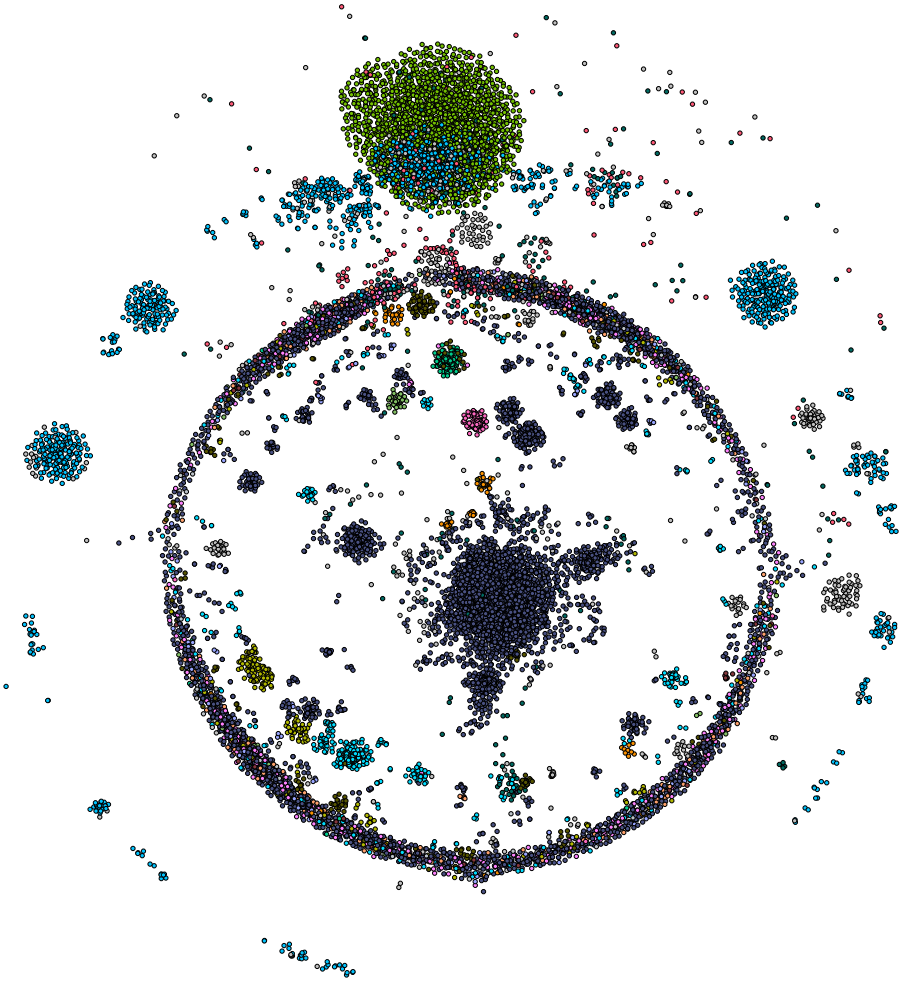


Figure 2.4: Relationships between the identified cryptojacking domains depicted in a force-directed graph.

Table 2.7: Identified campaigns based on a shared *siteKey* (HT = hiding technique encountered).

SiteKey	#	Type	Attack vector	HT
I2OG8vG[..]coQL & hn6hNEm[..]w1hE	987	Coinhive	Third-party software (WordPress)	5
I8rYivhV3ph1iNrKfUjvdqNGfc7iXOEw	376	Coinhive	Compromised websites	2
oHaQn8u[..]EvOS, XoWXAWvi[..]JfGx, no2z8X4[..]w2yK	317	Coinhive	Third-party software (Drupal)	2
TnKJQivLdI92CHM5VDumySeVWinv2yfL	213	Coinhive	Third-party software (WordPress)	1
GcxML3FZ;60;1 & GcxML3FZ;-70;1	180	Mineralt	Third-party software (WordPress)	6
ZjAbjZv[..]9FiZ, PQbIwg9H[..]gfVW	175	Coinhive	Third-party software (Magento & WordPress)	4
w9WpfXZJ9POkztDmNpey3zA1eq3I3Y2p	103	Coinhive	Compromised websites	2
j7Bn4I56Mj7xPR2JrUNQ9Bjt6CeHS3X1	79	Coinhive	Third-party software (WordPress)	2
cb8605f33e66d9d[..]6af74f86e6882899a8	70	Cryptoloot	Website owner initiated (The Pirate Bay)	2
49dVbbCFDuhg9nX[..]K2fkq5Nd55mLNnB4WK	70	Coinhive	Compromised websites	1

Table 2.8: Identified campaigns based on shared WebSocket proxy servers (HT = hiding technique encountered).

WebSocket proxy server	#	Type	Attack vector	HT
wss://ws**.1q2w3.life/proxy	28	Nebula	Website owner initiated	6
wss://coin-services.info:****/proxy	27	Coincube	Compromised websites	6
wss://heist.thefashiontip.com:8182/	24	Webminerpool	Malicious advertisements	5
wss://delagrossemerde.com:8181//	15	Webminerpool	Website owner initiated	8
wss://wss.rand.com.ru:8843/	13	Coinhive	Third-party software (WordPress)	8
ws://185.165.169.108:8181/	8	Webminerpool	Website owner initiated	2
ws://68.183.47.98:8181/	7	Webminerpool	Website owner initiated	2
wss://gtg02.bestsecurepractice.com/proxy2/	6	Unknown	Third-party software (WordPress)	3

Additionally, we have discovered 14 WebSocket proxy servers with very similar addresses on 75 domains (e.g. `nflying.bid`, `flightzy.bid` and `flightsy.bid`). These servers are contacted by the most obfuscated miner encountered in this crawl. The miner code is hidden inside a randomly named file, the miner code is heavily obfuscated, and the WebSocket traffic is sometimes encrypted. Our efforts to reverse engineer the obfuscated miner code have so far been unsuccessful. Therefore, we can not cluster them as being a campaign based on the shared proxy servers, but we have added the signature to our crawler as a separate mining application for the next crawls.

FOUND ON SHARED INITIATOR FILE

In our crawling process, the stack trace of an initialized WebSocket connection is saved for every website. While examining these stack traces, some file names emerged and led to the identification of another 4 cryptojacking campaigns. The oddly named file `gninimorenomv2.js`, responsible for opening WebSocket connections on 24 websites, seemed to be part of a malicious advertisement campaign, which injects cryptojacking scripts into served advertisements. As shown in Table 2.8, this file opens a connection to `wss://heist.thefashiontip.com:8182/` to earn the profits from the displayed mining advertisements. Another campaign was identified by grouping the websites in which `adsmine.js` was responsible for opening a WebSocket connection. These websites turned out to be 17 very similar pornography websites, which indicates that this campaign is website owner initiated. The newly discovered mining application, as described in the previous section, served obfuscated mining scripts to its miners. Although obfuscated, an inspection of the random file names revealed clusters of websites injected with the same randomly named miner, which led to the discovery of another 3 campaigns, all targeting WordPress websites.

FOUND ON SHARED MINING POOL LOGIN

Most miner applications submit their solved hashes to a WebSocket proxy server, which combines the hashes of multiple miners before forwarding them to the actual mining pool. However, we have discovered 238 websites directly submitting their hashes to a mining pool. These websites use only six unique cryptocurrency wallet addresses. The shared wallet addresses guarantee that profits made by cryptojacking are transferred to the exact same wallet. These findings did not lead to the discovery of any new campaigns but did confirm previous findings. E.g., proxy `wss://delagrossemerde.com:8181/` (used by 15 sites) is solely receiving traffic from domains using the same wallet.

The different methods used in this section enabled us to find 204 cryptojacking campaigns. We can conclude that the largest campaigns are using third-party services like WordPress, Drupal or Magento as their method of spreading. Only one campaign using advertisements with injected cryptojacking scripts has been identified, this in contrast to previous work by [127, 199], who reported malicious advertisements as a significant attack vector. Compromised websites or website owner initiated campaigns are generally smaller in size. The obfuscation level used in most campaigns is rather low, heavily obfuscated code is encountered rarely and in more than half of the identified campaigns a miner added in plain text.

2.6.2. AN IN-DEPTH CAMPAIGN SEARCH

The sizes of the campaigns identified in § 2.6.1 depend on the dataset we crawled, so they could have been incomplete. To find more websites belonging to the identified campaigns, we have taken the indicators of compromise for a large number of campaigns and queried PublicWWW for domains matching these IoCs. This resulted in a dataset of 7,892 websites. Combined with the 21,022 potentially cryptojacking websites from the initial crawl, a total of 25,121 URLs was crawled on February 12, 2019, more than a month after the initial crawl. We successfully obtained 24,187 (96%) of them.

Most of the campaigns remained of similar size in this crawl, except for a campaign involving three keys, `ef937f99557277ff62a6fc0e5b3da90ea9550ebcdfac`, `06d93b8-46706f4dca9996baa15d4d207e82d1e86676c` and `dd27d0676efdecb12703623d6864-bbe9f4e7b3f69f2e`. This advanced campaign is targeting domains using Bitrix24, a CRM platform. The most remarkable website within this campaign is the website of the Ministry of Education of Belarus (`https://edu.gov.by/`). The malicious code is hidden as the core loader of Bitrix24 and uses both Nerohut and Cryptoloot to mine with. It has a built-in anti-detection method since it stops mining once a developer tools window is opened. In our initial crawl, we identified only 68 domains belonging to this campaign, which turned out to be 855 in our in-depth search, making this campaign the second-largest campaign we have identified so far. Another campaign, involving key `vPfPDHk89TxmH1arysiJDrutpYGntofP`, is displaying fake loading screens on 86 websites, whereas only 47 of these have been identified in our initial crawl.

All other campaigns remained similar or slightly smaller in size. Except for the two aforementioned campaigns, we conclude that our initial crawl likely identified the correct size of campaigns, given the database of PublicWWW. Their database contains source code snapshots of over 544M websites, which should provide a proper approximation.

2.6.3. EVOLUTION OF CRYPTOJACKING

To study the evolution of cryptojacking on the Internet, data is needed from different moments in time. Fortunately, Konoth et al. [127] shared their crawling results, and Hong et al. [101] shared their list of identified cryptojacking domains, which made it possible for us to crawl these exact same sets of URLs and to analyze whether these domains were still mining. Additionally, we have followed the domains identified in our crawls over a period of 3 months, and analyzed WebSocket proxy traffic over time using operator NetFlows.

COMPARISON WITH PREVIOUS CRAWLS

Konoth et al. [127] crawled from March 12 until 19, 2018, and identified 1,735 potential cryptojacking domains. We crawled their list on January 21, 2019, and obtained 1,725 of them. 85% of the websites are not cryptomining anymore, and only 10% are still using the same application. On 136 websites (7%), the same key was found in both crawls. As Figure 2.5 shows, many websites using a Coinhive miner removed the miner application. Some continued using Coinhive, but also a small shift into less popular mining applications can be observed. Websites already using these miners tend to stick to their choice and are still using the same miner almost a year later. We have also seen a number of

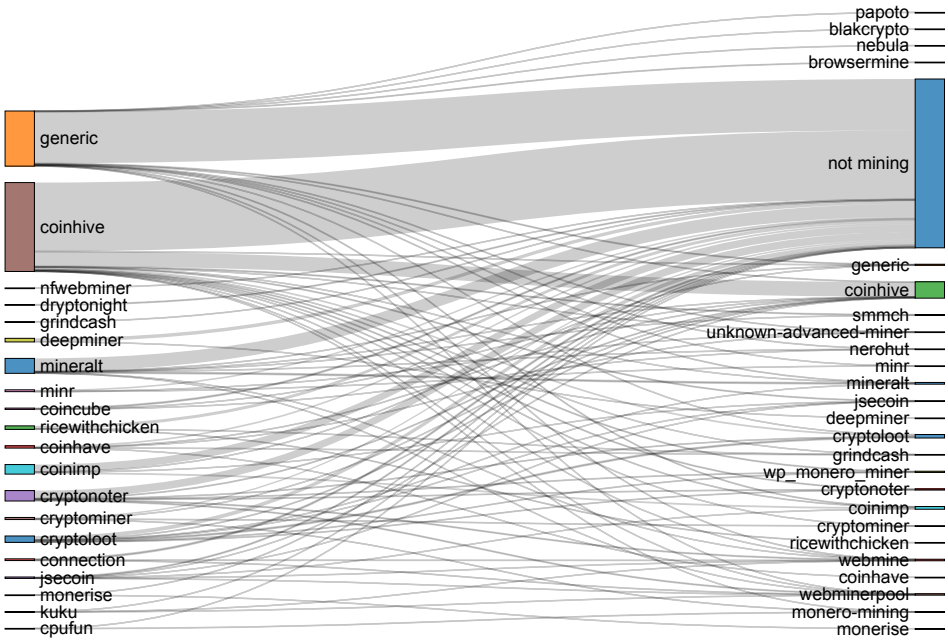


Figure 2.5: Usage evolution between March 2018 and January 2019 in the list of identified domains by [127].

mining applications become extinct, such as Deepminer and NF Webminer. Hong et al. [101] also published the list of identified cryptojacking domains from their crawl in February 2018. A year later, on February 12, 2019, we have crawled this list of 2,770 domains. We obtained 2,435 (88%) of them and only 340 (14%) domains are still actively cryptojacking. Both crawls show that many websites stopped cryptojacking themselves or removed the miner infection. After one year, approximately 85% of the domains are not actively cryptojacking anymore. We have also observed a small portion of domains switching to less popular applications. The low number of 7% of websites still mining with the same *siteKey* indicates the fast changes in the cryptojacking threat landscape.

EVOLUTION OF IDENTIFIED DOMAINS

We have followed all previously identified cryptojacking domains for a period of 3 months (until May 5, 2019) and crawled them initially occasionally but afterward every other day. Within this time period, Coinhive announced the end of its mining application due to decreased Monero prices and hash rate [45]. The announcement was made on February 26, 2019, and stated that mining would not be operating anymore after March 8, 2019 and that the service would be discontinued by the end of April 2019. This led to a drastic change in the cryptojacking landscape, as Coinhive's dominance in actively mining installations collapsed when their mining service was set non-operationally. Mining applications were, however, not massively replaced, which confirms our finding that a large portion of browser-based cryptomining is not initiated by the website owner. Only when the Coinhive mining service was actually

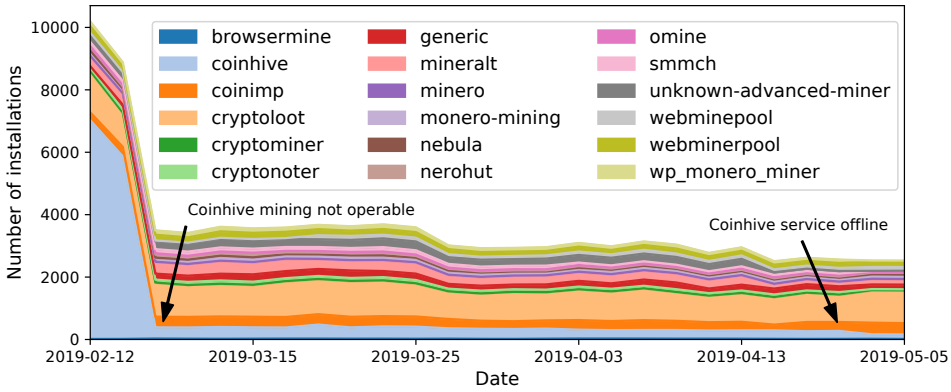


Figure 2.6: Evolution of the cryptojacking domains per type.

discontinued, and errors were shown when requesting the offline Coinhive mining resources did we observe a small increase in Cryptoloot and CoinImp installations.

WEBSOCKET PROXY TRAFFIC OVER TIME

As discussed in § 2.2, most miner applications use a WebSocket proxy server to forward traffic from their miners to the mining pool. Using the aforementioned NetFlow data, we analyzed traffic towards popular WebSocket proxies from September 2017 till December 2018, which gives an insight into the evolution of cryptomining applications usage, as shown in Figure 2.7. We have taken the set of WebSocket proxy IPs the miners connect to as a basis, which we extended by using passive DNS data to discover other WebSocket proxy server IPs used by these applications but hosted on different servers, not encountered during our crawls. The same passive DNS data was used to verify whether these IP addresses were solely used as WebSocket proxy servers. To prevent other traffic to these servers from being in our dataset, we have both set the maximum packet size to 550 kB and verified that only WebSocket traffic was counted towards these servers. For most proxies, this is traffic towards port 80 or 443, and for a few servers using specific ports, this could be different. An example is the WebSocket proxy server of the WP-monero-miner which uses port 8020.

The blue line starting in September 2017 shows how the web-mining ecosystem was monopolized by innovator Coinhive at the start, whereas copycats like Cryptoloot and Webmine started to emerge in October. We see that CoinImp essentially started to eclipse all other miner applications from mid-April 2018 onward regarding mining traffic to the proxies. This is unexpected, given the distribution of installations on websites and previous studies. Some mining proxies only have transient success: a remarkable example is the WP-monero-miner, released shortly after Coinhive in 2017. The application hosted its own mining pool and digested a lot of traffic in January 2018, only to almost disappear again weeks later. Coinhive, the application used by most websites, is a constant factor in the miner landscape with over 4,000 NetFlows a day in mid-2018 (given our 1:8192 sampling, thus 32M connections per day), but not as large

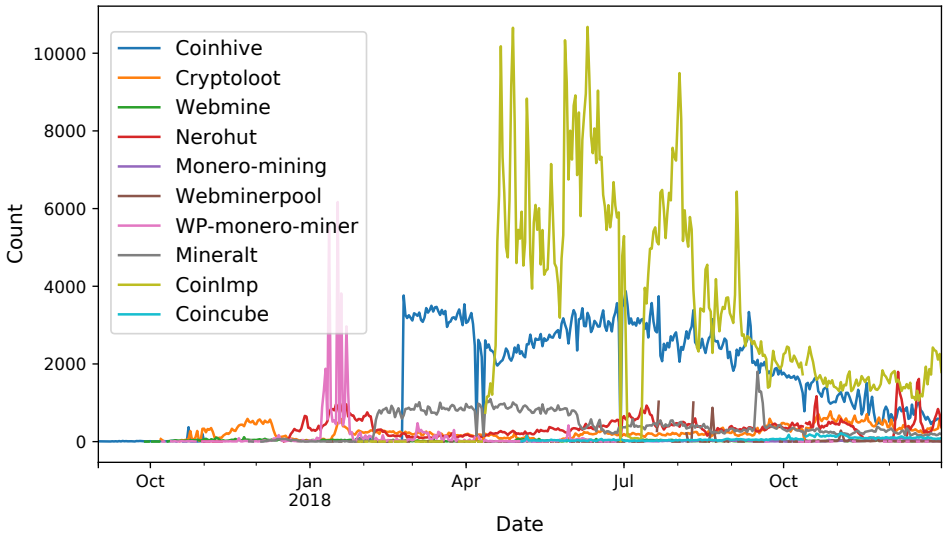


Figure 2.7: Number of NetFlows involving WebSocket proxy servers for popular miners between Sep 2017 and Dec 2018.

as one would expect from its installation base. Additionally, a clear declining trend can be observed in the NetFlow counts for all mining services after the summer of 2018. The last months of NetFlow data show a diverse set of mining applications actively used.

2.7. AN INTERNET-SCALE STUDY ON CRYPTOJACKING

In order to estimate the prevalence of browser-based cryptojacking on the Internet and to indicate any differences between Top Level Domains (TLDs), we have performed another crawl, in which we have crawled ~20% of the websites belonging to each of the 1,136 existing TLDs. We obtained a daily zone transfer for all generic top-level domains (gTLDs) – such as *.top*, *.loan* – from the Internet Corporation for Assigned Names and Numbers (ICANN), as well as a feed of registered country code top-level domains (ccTLDs) – such as *.uk*, *.jp*, or *.ru* – from a security intelligence provider. From these lists, we randomly picked a sample of ~20% of the size of each TLD [56]. Based on the results of the previous crawl, we have added another 5 mining applications to the crawler implementation, as listed in Table 2.9. From January 11 until April 3, 2019, we crawled the random sample, including 48.9M domains. This yielded a total of 125 TB of network traffic.

2.7.1. GENERAL FINDINGS

After crawling a random sample of 48.9M websites in a large number of different top-level domains, we are able to draw conclusions about the prevalence of browser-based cryptojacking on the Internet. We estimate that 0.011% of all domains are actively cryptomining without their visitors' explicit consent, meaning that one in every 9,090 web-

Table 2.9: The added miner applications and their keywords in the latest version of the crawler.

Miner	Keywords
SMMCH	simple-monero-miner-coin-hive smmch-public | smmch-mine.js
Webminepool	webminepool.com/lib/base.js
Unknown miner	proofly.date | flightsy.date | gettate.trade | nflying.win alflying.date | flightzy.date | joytate.date | flightzy.bid zymerget.faith | flightsy.win | zymerget.bid | nflying.bid baseballnow.press | flightsy.bid
Omine	omine.org
Browsermine	browsermine.com.cc | bmcm.pw | bmnr.pw lm-sdfhfad.ml | new BMCM | asdvhsrtsb.ml

sites is cryptojacking. Comparing this number to the statistics of the top lists used in our initial crawl, we conclude that cryptojacking activity is mainly focused on the popular parts of the Internet. In the Alexa Top 1M, 0.065% of the websites was actively cryptojacking, in this random sample only 0.011% of the websites, which is almost 6 times lower. This can be explained by the lucrativeness of cryptojacking, in which a higher popularity means more visitors, yielding more potential miners and thus higher potential profits. Additionally, it shows that researching the prevalence of cryptojacking by crawling the Alexa Top 1M overestimates the problem size. However, the distribution of used applications in our random sample is fairly similar to the distribution in the Alexa Top 1M. The distribution of mining applications in this crawl is listed in Table 2.10.
The categories of domains identified in this crawl are very similar to the initial crawl. As depicted in Figure 2.8, *Adult content* remains the most prevailing category, while other large categories are *Technology* and *Under Construction*, the category involving parked, expired or yet-to-be developed domains. Based on these two very different crawls we can conclude that cryptojacking is indeed more prevailing on domains hosting adult content.

2.7.2. CRYPTOJACKING ON DIFFERENT TLDS

We have crawled domains of roughly ~20% of 1,136 different TLDs in order to analyze the prevalence of cryptojacking. As Table 2.11 shows, cryptojacking activity varies enormously within different TLD zones. The four largest TLDs, *.com*, *.de*, *.net* and *.org* have a similar percentage of cryptojacking websites, but we have discovered almost 6 times as much cryptojacking activity in the Russian TLD. Also, domains in the Brazilian and Spanish zones are more susceptible to cryptojacking, having respectively 4 and 3 times more cryptojacking activity than average. On the contrary, the *.top*, *.us* and *.loan* zones host only a few cryptojacking websites. Our website category analysis showed that adult content is the most prevailing category for cryptojacking activities. This triggered our attention for the *.xxx* domain, which is specially created for adult content, which we, therefore, crawled completely instead of ~20%. Surprisingly, the *.xxx* domain contains only one website actively cryptomining. When comparing used mining applications on

Table 2.10: Distribution of cryptomining applications installations in the Internet scale crawl (sum of percentages is >100%, because of websites using multiple applications).

Type	# of websites	Percentage
Coinhive	2,531	48.767%
Unknown	689	13.276%
CoinImp	513	9.884%
Cryptoloot	504	9.711%
Mineralt	276	5.318%
Nerohut	247	4.760%
Webminerpool	233	4.489%
Unknown-advanced-miner	92	1.773%
SMMCH	80	1.541%
Browsermine	73	1.407%
Webminepool	62	1.195%
WP-Monero-Miner	60	1.156%
Omine	56	1.079%
Monero-mining	55	1.060%
Cryptonoter	50	0.963%
Cryptominer	26	0.501%
Minero	24	0.462%
Nebula	23	0.443%
Webmine	19	0.366%
Coincube	19	0.366%
Project-poi	4	0.077%
Adless	1	0.019%

Table 2.11: Results of the TLD crawl. Listed are the top 10 largest domains, followed by remarkable TLDs.

TLD	Size	Crawled	Cryptojacking
.com	149,937,597	27,555,546 (18.4%)	2,353 (0.009%)
.net	15,008,406	2,741,550 (18.3%)	238 (0.009%)
.de	15,089,860	2,244,139 (14.9%)	254 (0.011%)
.org	11,330,764	2,021,630 (17.8%)	145 (0.007%)
.info	6,524,248	1,309,323 (20.6%)	77 (0.005%)
.ru	5,480,467	998,422 (20.0%)	593 (0.059%)
.nl	5,360,173	880,122 (16.4%)	191 (0.022%)
.top	4,024,497	788,748 (19.6%)	19 (0.002%)
.br	3,813,745	383,910 (10.1%)	185 (0.048%)
.fr	3,449,775	567,887 (16.5%)	133 (0.023%)
.pl	2,621,515	523,497 (20.0%)	81 (0.015%)
.us	2,409,802	472,323 (19.6%)	2 (0.000%)
.loan	2,228,165	445,749 (20.0%)	0 (0.000%)
.es	2,010,710	327,810 (16.3%)	110 (0.036%)
.online	1,105,999	219,447 (19.8%)	67 (0.031%)
.pro	295,201	58,999 (14.2%)	32 (0.054%)
.space	268,846	53,363 (20.0%)	19 (0.036%)
.website	276,063	54,704 (19.8%)	21 (0.038%)
.xxx	93,101	91,877 (98.7%)	1 (0.001%)
Total		**48,948,669**	**5,190 (0.011)%**

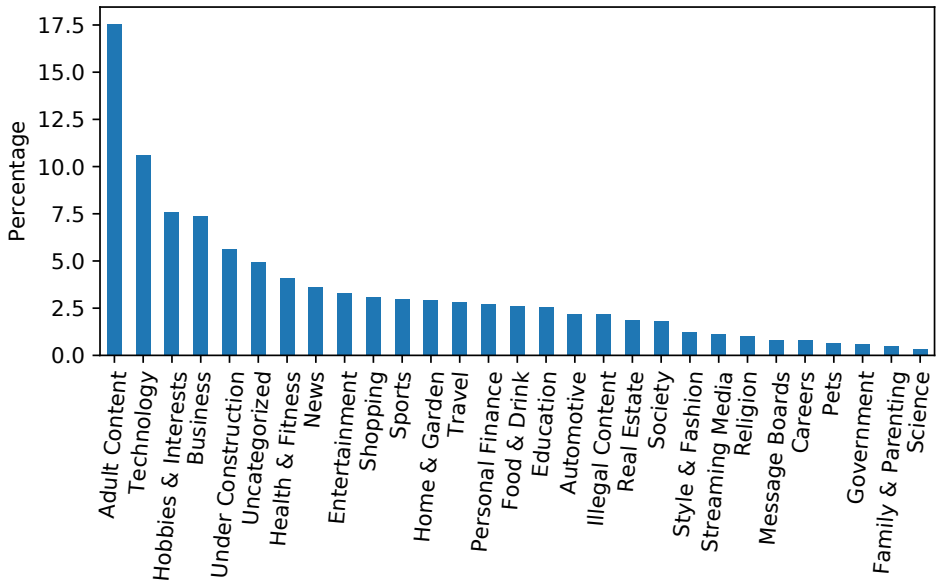


Figure 2.8: Categories of mining domains in the second crawl.

the different TLDs, large differences can be distinguished, as shown in Figure 2.9. Coinhive is the most popular miner in most zones, whereas Cryptoloot is preferred in the Russian zone, and French and Czech websites contain more Nerohut miners. The Russian zone is also the only TLD where browsermine is used regularly. The high number of generic miner applications in the Dutch and Belgian zones is remarkable. A large number of these domains in the *.nl* and *.be* zone are part of a campaign using expired domain names of a Dutch registrar (*Totaaldomein B.V.*) to host porn and unknown cryptominers.

Our results show a different popularity of used mining applications compared to previous work of [207]. They detected Coinhive on 85% to 90% of the *.com*, *.net* and *.org* TLDs, whereas we determine that this market share is significantly lower (~50%). This result proves that a simple solution like the NoCoin block list is unable to detect all miners, and analyses with such techniques result in different outcomes.

2.8. DISCUSSION

Crawling the Internet inevitably comes with its shortcomings. Limitations in the crawler implementation, network used and analysis can produce both false positives and negatives. The latter category can occur, for example, when extreme obfuscation is used, as we have seen in § 2.6. However, we believe that due to our double crawling strategy, based on both WebAssembly and code signatures, this could not have happened very often. Finally, the use of worldwide NetFlow traffic from a Tier 1 network operator allowed us to analyze the popularity of cryptojacking services in a revolutionary way, although BGP policies and a specific PoP and IXP footprint could lead to a bias of certain

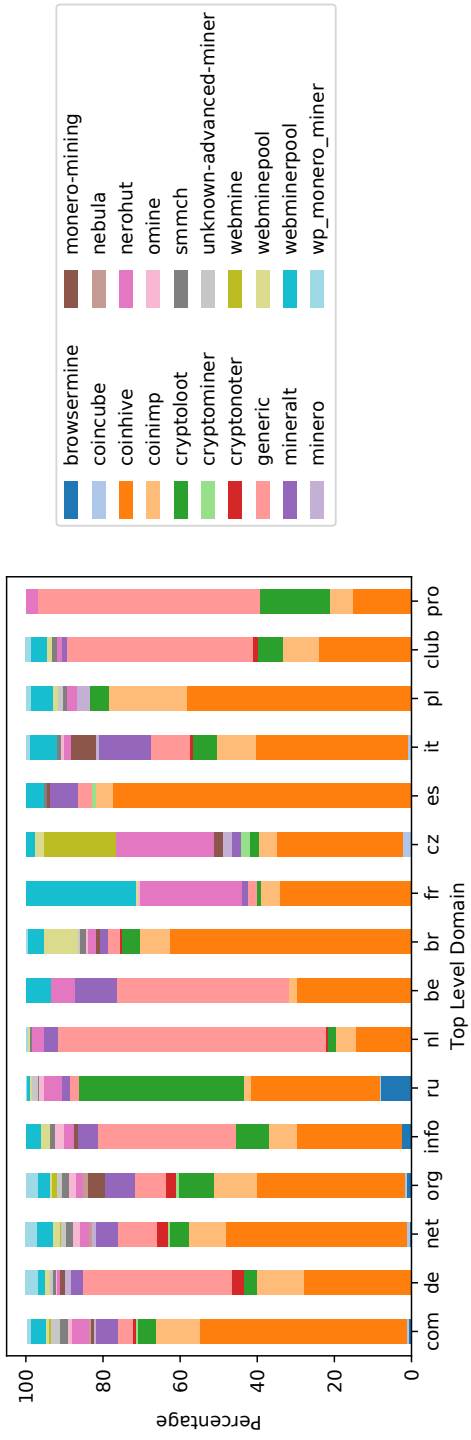


Figure 2.9: The distribution of used mining applications in various TLDs.

autonomous systems just as some discrepancies might arise due to 1:8192 random sampling. Additionally, NetFlow records do not provide information regarding the content of a connection; therefore, the actual contents cannot be determined. However, during our crawls, we could confirm the mining applications to contact the WebSocket proxy servers in question, and passive DNS lookups did not show any other domains pointed to that IP. Furthermore, the NetFlows both revealed no traffic to other ports than those seen from our crawlers and packet sizes resembling those observed in our crawls. Thus, the methodology should provide valid results.

FUTURE WORK

The additional angle provided by the NetFlow data allowed us to study the evolution of cryptojacking over a longer period of time, something that has not been done before. Regular crawls of the Internet, especially of the already identified cryptojacking domains, give more insight into this practice, and it will increase the innovation of defense mechanisms. The most influential defense against cryptojacking will nonetheless be frequent patching, as most cryptominers are installed exploiting known vulnerabilities. CMS providers, such as Drupal or WordPress, have shown agility in patching vulnerabilities, but the responsibility of installing these patches remains with the website owner. Finally, as we have seen a decline in the price of Monero (-85% in 2018), we believe that cryptojacking infections on individual websites will decrease but that cyber criminals will search for other possibilities to exploit cryptojacking at an even larger scale. As we have mentioned in § 2.3, the most effective method of collecting large groups of miners is by launching a MITM attack. Investigating the prevalence of this attack vector for cryptomining is something we preserve for future work.

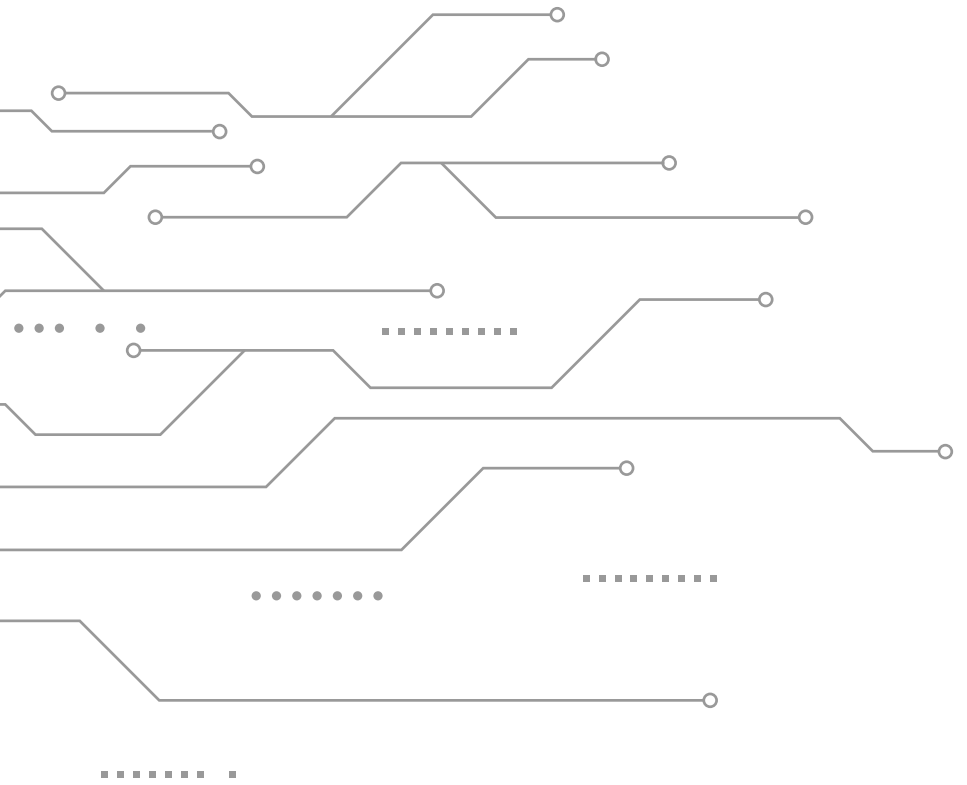
2.9. CONCLUSIONS

In this chapter, we have studied the prevalence of cryptojacking as well as of cryptojacking campaigns on the Internet. We have performed multiple large crawls, each with a different focus. In our first crawl, we analyzed the 1.7M most popular domains to identify organized campaigns. We found 204 campaigns, from which we conclude that the size of cryptojacking campaigns is heavily underestimated by current academic research. Additionally, using solely the Alexa Top 1M shows significantly different results regarding the size of organized activity and infection rate, which we found to be almost 6 times lower in a random sample compared to the Alexa Top 1M, hence overestimating the problem. Third-party software is often used by attackers to spread cryptojacking scripts over a large number of domains. The share of domains serving advertisements injected with cryptojacking scripts is lower compared to previous work, most likely because of stricter monitoring by advertisement networks. We have seen that obfuscation of cryptojacking scripts is definitely present but only occasionally used. Comparing our results with data from previous studies (in both February and March 2018) shows that after a year, only 15% of the websites are still actively mining. This, and our novel way of estimating miner application popularity by analyzing NetFlows, led to the conclusion that the cryptojacking landscape is constantly changing and involves a variety of actors.

A second, Internet-scale crawl involving ~20% of 1,136 TLDs (48.9M websites), which represents a truly random sample of the Internet, allows us to conclude that cryptojack-

ing is present on 0.011% of all domains. Not unexpectedly, this percentage increases in the more popular parts of the Internet because cryptojacking on popular domains is much more lucrative. Both of our crawls have shown that cryptojacking mostly takes place on websites hosting adult content, although the *.xxx* TLD is home to only one cryptojacking website. Based on the applications used within the time span of our analysis, we can conclude that Coinhive was the largest mining application in terms of installation base but that CoinImp's WebSocket proxy servers were digesting much more traffic in 2018. Looking at the different TLDs, we conclude that Russian, Brazilian, and Spanish zones are home to a disproportionate number of cryptojacking domains.

With the discontinuation of Coinhive in March 2019, the landscape of cryptojacking has changed enormously, but based on our results, we are only expecting a further decline in individual cryptojacking activities, given that the Monero value keeps diminishing. However, this only stresses the importance of organized cryptojacking campaigns, as cyber criminals will find new ways to spread their cryptojacking infections to remain profitable. Here, campaign analysis will be an important asset. As adversaries are unlikely to develop a unique approach for each infected website, reusing resources and methods will provide an effective angle to detect and mitigate these activities.



3

EXAMINING CRYPTOJACKING ON COMPROMISED INFRASTRUCTURE

The release of an efficient browser-based cryptominer, as introduced by Coinhive in 2017, has quickly spread throughout the Web either as a new source of revenue for websites or exploited within the context of hacks and malicious advertisements. Several studies have analyzed the Alexa Top 1M and found 380 — 3,200 (0.038% – 0.32%) [29, 101, 127, 199, 205] to be actively mining, yielding an estimated $41,000 per month revenue for the top 10 perpetrators [127]. While placing a cryptominer on a popular website supplies considerable returns from its visitors' Web browsers, it only generates revenue when a client visits the page. Even though popular websites attract millions of visitors, the relatively low number of exploiting websites limits the total revenue that can be made. In this chapter, we report on a new attack vector that drastically overshadows all existing cryptojacking activity discovered to date. A firmware vulnerability in MikroTik routers allowed cybercriminals to rewrite outgoing user traffic and embed cryptomining code in every outgoing Web connection. Thus, every *Web page visited by* any *user behind an infected router would mine to profit the criminals. Based on NetFlows recorded in a Tier 1 network, semiweekly crawls, and telescope traffic, we follow their activities over 10 months and report on the modus operandi and coordinating infrastructure of the perpetrators, which were during this period in control of up to 1.4M routers, approximately 70% of all MikroTik devices deployed worldwide. We observed different levels of sophistication among adversaries, ranging from individual installations to campaigns involving large numbers of routers. Our results show that cryptojacking through MITM attacks is highly lucrative, a factor of 30 more than previous attack vectors.*

This chapter has been published as: **Bijmans, H.L.J.**, Booij, T.M. & Doerr, C. (2019). "Just the Tip of the Iceberg: Internet-Scale Exploitation of Routers for Cryptojacking". In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS 2019).*

3.1. INTRODUCTION

Cryptocurrencies, which started with the release of Bitcoin in 2009 [173], represent monetary value secured by a blockchain. Transactions are permanently stored in an ever-growing list of records, where transaction data can be added by solving a cryptographic challenge. This puzzle depends on the last block, the current transactions, and their recipients. Once solved, a new record gets inserted into the chain, consolidating previously conducted activities' records. Users are incentivized to participate and donate computational resources to the system, as the one solving the puzzle gets a (fraction of a) cryptocurrency unit as a reward.

As the Bitcoin blockchain was designed to increase the difficulty of these challenges continuously, Bitcoin mining is no longer profitable on regular PCs, requiring specialized hardware such as ASICs. As a result, thousands of other cryptocurrencies, so-called alt-coins, have emerged that replace the proof-of-work algorithm of Bitcoin with alternative mechanisms to validate transactions. The Monero cryptocurrency [236], which uses a private blockchain with transactions not publicly visible, relies on the CryptoNight algorithm, a memory-intensive computation of subsequent reads and writes that can be efficiently run using the processor-level cache found in typical consumer-grade CPUs.

The reward that can be gained from these alt-coins has, however, also attracted the attention of cyber criminals, who have distributed cryptomining code through malware or as part of botnet installations [190]. With the recent introduction of a JavaScript miner by Coinhive in 2017, cryptomining code can now be shipped as part of a Web page and efficiently executed by a Web browser, thereby providing an easy, scalable, and low-effort method to roll out cryptomining to a large user population. This has led to new business and revenue models, for example, replacing advertisements by letting website visitors donate computational resources [237].

The relative ease with which website visitors can be recruited for cryptomining has also led to a major surge in illicit cryptomining, so-called cryptojacking or drive-by mining, in which the visitor's resources are hijacked without knowledge and consent. Aside from cryptojacking that is initiated by the website owner without their visitors' consent, criminals also seek to increase their revenue by compromising websites to install mining code [34, 157], as well as hiding miners in third-party software used by Webmasters and thus inadvertently being deployed [40, 255]. Cryptojacking is also spread through exploitable vulnerabilities in content management systems [171], through the distribution of advertisements including malicious code [169], or through malware [190]. Previous work by Konoth et al. [127] estimates that cryptojacking possibly yields monthly revenues of $41,000 for the 10 most successful perpetrators across the websites listed in the Alexa Top 1M.

In this chapter, we will analyze a previously unseen attack vector for cryptojacking, namely, man-in-the-middle attacks launched through compromised consumer and edge routers that inject mining code into every Web page requested by their users. This was made possible by a firmware vulnerability in MikroTik routers discovered in early 2018 [178], which allowed adversaries to change the device configuration and create an outgoing HTTP proxy, and that remained widely unpatched until a year later. By following reconnaissance scans of the perpetrators through a large network

telescope, the detection of compromised routers through semiweekly crawls, and the tracing of connection patterns of adversaries, their supporting infrastructure, and the compromised routers based on NetFlows from a Tier 1 operator, we are able to provide a comprehensive insight into how this vulnerability scaled out into massive cryptojacking campaigns that drastically overshadow previous mining activities. In this work, we make the following four contributions:

- We are first to investigate a new type of attack that exploits Internet infrastructure for cryptomining and show how, over a period of 10 months after the initial discovery of the vulnerability, cybercriminals launch massive campaigns to control 1.4M routers, with a peak of 460,618 simultaneously infected routers.
- We analyze adversarial tactics and unveil the supporting infrastructure used within the campaigns, and are able to show differences between groups in how they locate their victims, compromise routers, and run their infrastructure.
- We demonstrate that previously reported vectors are negligibly small in number of affected users and created revenue compared to the reported MITM attack. We find that this attack yielded monthly revenues estimated to exceed $1,200,000 per month for the top 10 grossing accounts, a factor of 30 larger than previously estimated cryptojacking revenues from hacked websites, malicious advertisements, and website-owner-initiated mining combined.
- We observe high levels of sophistication in three identified campaigns, of which the largest involved 40 mining accounts linked to one single actor.

The remainder of the chapter will be structured as follows: § 3.2 provides an overview of related work. § 3.3 introduces the concept of cryptojacking and previously used modus operandi and describes the vulnerability and its exploitation used for MITM-based cryptojacking on routers. § 3.4 describes the datasets used in this study. § 3.5 presents the techniques, tactics, and procedures in use during the identification, exploitation, monetization, and maintenance of the compromised systems. § 3.6 puts the techniques and sophistication levels of the ecosystem into perspective and quantifies adversarial revenues. Finally, § 3.7 summarizes and concludes our work.

3.2. RELATED WORK

The growing interest in cryptojacking by cyber criminals was followed by an interest of the academic world to research this new phenomenon. Only shortly after the release of the Coinhive miner in September 2017, Eskandari et al. made the first explorations into the field by searching source code databases *Censys.io* and *PublicWWW* for strings known to be part of cryptomining libraries [63]. Due to the possibilities of JavaScript obfuscation and other hiding techniques, other research that followed soon focused on the detection of these mining applications. Rauchberger et al. built *Mininghunter*, a crawler instructed to analyze both source code and WebSocket traffic of the visited pages [199]. A crawl of the Alexa Top 1M resulted in the identification of 3,178 cryptojacking websites and the discovery of a number of campaigns. Other Web crawling studies performed by

Parra Rodriguez & Posegga [205] and Carlin et al. [29] used machine learning techniques to determine active mining on a Web page while crawling, and both reached high precision scores. The *CMTracker* made by Hong et al. [101] also crawls the Web but detects cryptojacking behavior based upon periodic executions in WebAssembly modules. This robust detection method was able to identify 868 actively mining websites in the Alexa Top 100K. Wang et al. performed a similar study by learning a support vector machine (SVM) on the characteristics of WebAssembly modules and concluded that analyzing WebAssembly modules is a very efficient and robust detection method [250].

The latest Web crawling studies involve the work of Rüth et al., who crawled the three largest top-level domains (*.com*, *.net* and *.org*) as well as the Alexa Top 1M to estimate the prevalence of browser-based cryptomining (0.08% of the probed websites were actively mining) [207], Konoth et al., responsible for creating another crawler which identified 1,735 actively mining websites and performed campaign analysis to gain knowledge about the ecosystem [127], and Kharraz et al. performing a similar study but identified the actively cryptomining websites using machine learning techniques [124]. They also dedicated a section to campaign analysis, in which the authors identified 35 campaigns involving a total of 386 websites. The largest study to date across 55M websites discovered that the prevalence of cryptojacking significantly varied by top-level domain zone and the popularity of websites, and that about half of all cryptojacking activity is organized and part of a campaign [16].

Initial evidence begins to suggest that the exploitation of websites for browser-based mining through their visitors might not generate the main source of revenue. Papadopoulos et al. concluded that advertisements are still over five times more profitable than browser-based cryptomining [189]. A longitudinal study performed by Pastrana et al. revealed that in the cryptojacking ecosystem, only a small number of cybercriminals are making large profits, and those making profits had mined 4.3% of all Monero in circulation. [190].

Previous studies have primarily focused on either the detection or the estimation of the compromised website attack vector. While cryptojacking as part of a man-in-the-middle (MITM) attack – for example, through a malicious WiFi network – is mentioned as being feasible by Eskandari et al. [63], this particular attack vector has never been researched before by the academic community. As an MITM attack will affect all traffic that crosses a particular device, the potential number of victims and, with it, potential revenue is, however, much higher. This chapter will thus address this gap, and show how and to what extent cryptojacking is deployed in the wild through the attack on Internet infrastructure.

3.3. BACKGROUND

The introduction of memory-bound cryptocurrencies like Monero allowed for new methods of cryptomining, one of them being browser-based cryptomining. These cryptocurrencies, together with new Web technologies such as WebAssembly (native speed code execution within the browser sandbox), WebWorkers (separate JavaScript instances), HTML5 WebSockets (simple multiplex TCP connection), and the Stratum Mining Protocol (JSON-RPC formatted mining pool communications) paved the way for the creation of an efficient browser-based cryptominer by Coinhive in 2017. Their

miner, and most other mining applications, work as follows: the user visits a cryptojacking website, which includes (a reference to) a cryptojacking script. This script explores the host system, downloads a highly optimized WebAssembly module for mining, and spawns a number of WebWorkers to run this module. Consequently, it sets up a connection with a mining pool through a proxy server operated by the service, authenticating using a *siteKey* or (Monero) wallet address, which is essentially the account of the adversary. For readability, we will refer only to the first six characters of a *siteKey* in this chapter. The mining pool distributes a job to work on. The WebWorkers start mining and find hashes that are submitted to the mining pool. When the browser window is closed, all mining activity stops.

3.3.1. PAST MODUS OPERANDI OF CRYPTOJACKING

As stated by a New Jersey Attorney General in 2015 [100], mining cryptocurrencies with the computing power of others is not considered illegal when a clear notification of such activities is shown, and the possibility of opting out exists. However, most cryptojacking cases lack these and are therefore considered illegal. There have been cryptojacking scripts found on malware-infected PCs [190], but since the release of the Coinhive miner, cryptojacking in the form of browser-based mining gained enormous popularity. There is a large number of websites running a cryptominer to increase their revenues, such as The Pirate Bay [237], but cryptojacking has also occurred on websites where the owner did not initiate it. Website compromises, such as government pages of the Indian government [34] in 2018, have led to cryptojacking infections, but cybercriminals are constantly searching for more efficient methods to deploy their miners. To spread infection over a large number of websites, attackers abused third-party software (such as infecting WordPress plugins [255] or exploiting Drupal CMS vulnerabilities [216]) with cryptojacking scripts as well as injected advertisements with mining code and served them through ad networks to websites unaware of any infection [169].

3.3.2. PERVASIVE CRYPTOJACKING THROUGH MAN-IN-THE-MIDDLE ATTACKS

As mentioned in the previous section, cryptomining code is included as part of the served HTML page, which requires the website owner to explicitly install a cryptominer or inadvertently embed it due to a compromised component. It is, however, also possible to modify the request in transit by modifying the HTML as a man-in-the-middle.

In the attack reported in this chapter, adversaries compromised the routers' operating system and reconfigured the system, causing requests from clients to any website to be rewritten and channeled to an internal HTTP proxy server running on the device. With the initial compromise of the router, the perpetrator installs a script to change the firewall rules of the device, opening telnet and SSH to the Internet if not already exposed, and introduces a firewall rule to redirect outgoing requests on port 80 to a proxy port. Finally, it deploys an HTML page sent by the proxy to each outgoing connection. While different groups of actors followed slightly different techniques, tactics, and procedures, as we will show in § 3.5, it meant, as shown in Figure 3.1 from the perspective of the user, any outgoing connection to port 80 was redirected to the proxy on port 80 or 8080 (1). This served a Web page based on a common template, shown in Listing 3.1 for a con-

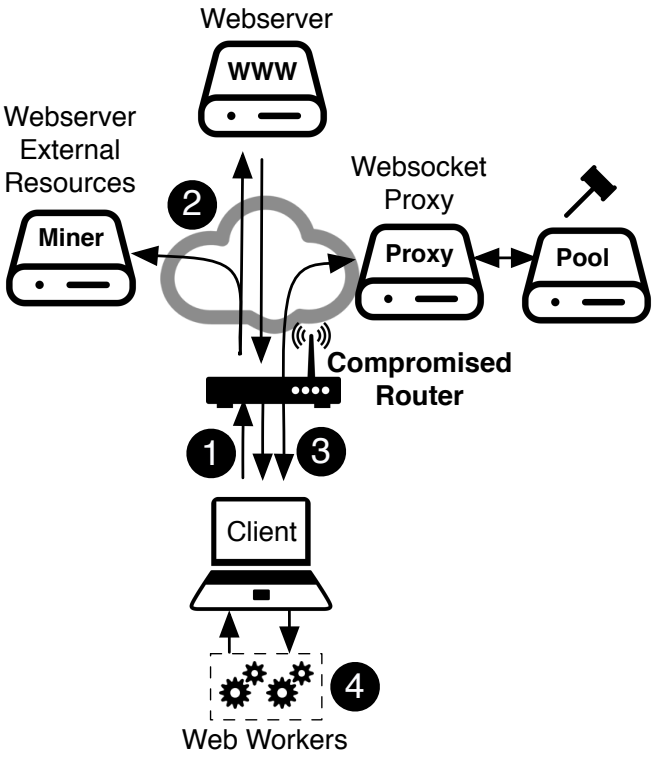


Figure 3.1: Through an MITM attack on routers, adversaries performed cryptojacking on websites visited by users.

```
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html;charset=windows
      -1251">
  <title>"http://www.facebook.com/"</title>
  <script src="https://coinhive.com/lib/coinhive.min.js"></script>
  <script>
    var miner = new CoinHive.Anonymous('<mining key>',{throttle: 0.1});
    miner.start(CoinHive.FORCE_EXCLUSIVE_TAB);
  </script>
 </head>
 <frameset>
  <frame src="http://www.facebook.com/"></frame>
 </frameset>
</html>
```

Listing 3.1: HTML returned by the proxy of an infected router, with a Coinhive miner and the actual page in an iframe.

nection to `facebook.com`. This led the client's Web browser to fetch two Web resources: the outer frame containing a JavaScript that loaded cryptomining code (2), and within the frame, the actual website the user intended to visit was displayed (2). The client's Web browser would set up a WebSocket connection to a WebSocket proxy or to mining pool in order to retrieve instructions (3), and spin up WebWorkers to mine for a specific *siteKey* (4).

From the perspective of the perpetrator, this design has a number of advantages. First, as the iframe opens the original page, the user will, at first sight, not notice anything wrong as the requested Web page loads within the borderless iframe. Second, as the interaction with the loaded website functions normally, the victim will remain on the Web page for an extended period of time, thus increasing the time the miner will run in the background. Third, as clicks on the embedded page do not reload the outer frame, the cryptominer keeps running during navigation on the visited Web page, thus maximizing mining cycles.

SUSCEPTIBILITY OF HTTP(S) CONNECTIONS

While the browser address would show a connection to the router instead of the requested URL, the hijack, from a usability perspective, is both comparatively frictionless and effective. The original URL is displayed as the title of the page, and experimentation on recent versions of both mobile and desktop browsers showed that websites can even be loaded via HTTPS within the iframe without triggering a warning by the browser. In this case, the HTTP proxy loads an unencrypted HTTP page with an iframe showing the secured HTTPS contents. Thus, unless the rewritten URL raises suspicion with the user, we can expect the activity to go by relatively unnoticed.

3.3.3. VULNERABILITY CVE-2018-14847

The exploited vulnerability in this attack is CVE-2018-14847 and affected MikroTik RouterOS through version 6.42, allowing *"unauthenticated remote attackers to read*

arbitrary files and remote authenticated attackers to write arbitrary files due to a directory traversal vulnerability in the WinBox interface." [178]. Of special significance to the attack is that MikroTik uses RouterOS across their entire product line, making the vulnerability applicable to a large number of both consumer and carrier-grade routers. As we will see later, the vulnerability of carrier-grade devices explains the magnitude of cryptomining activity that could be realized in this attack.

WinBox is a small Win32 binary that allows for the administration of RouterOS using a graphical user interface. The functionalities of the WinBox interface are almost identical to the console functions, but some advanced and critical system configuration changes – like changing the MAC address – cannot be made from the WinBox GUI. Several WinBox commands did not require authentication, e.g., an attacker could open files for reading while being unauthenticated, while another allows an attacker to write files to disk given some authentication [235]. By sending a carefully crafted package to the WinBox service on port 8291 exploiting one of these commands, the attacker would retrieve the user credential store `user.dat`, and using these credentials, drop files to disk to enable a developer backdoor [235]. Triggered if a specific file, `/pckg/option` or `flash/nova/etc/devel-login`, is present on the system, the developer mode sets up a root BusyBox shell accessible over port 22 (SSH) or 23 (Telnet) giving complete control over the device.

3.4. DATASETS

The study was made possible through a combination of three datasets, each covering a different angle of the reported malicious activity. First, we used traces from a large network telescope to trace adversarial scanning activity. Second, we relied on a periodic crawl for the proxy status page by Censys [61] and used Shodan [218] to discover which routers were infected. Third, we use NetFlow data to visualize the communication patterns between the infected routers and the remaining Internet to identify their staging hosts and quantify the volume and revenue of this large-scale exploitation.

Figure 3.2 shows a timeline of the main phases of the cryptojacking exploitation of MikroTik routers, together with the timeline and purpose of the used datasets. While the vulnerability was discovered in April 2018, the MikroTik routers were only exploited for MITM cryptojacking from the middle of July onwards until January 2019, when the bulk of the ecosystem was cleaned up. We use telescope traffic and operator NetFlows already months prior to the abuse from January 2018 onwards, to observe prior knowledge of the vulnerability and any preparation activities by the adversaries as discussed in [19, 83], to trace the activities of actors in finding these devices as well as to identify their installation, maintenance, and monetization strategies. After the monetization through MITM cryptojacking emerged in July, we then followed the state of the compromised devices through the public lists Censys and Shodan until the general wind-down of these campaigns. Details about each dataset are presented below. Table 3.1 lists all datasets collected and provides links to where they are used for analysis in § 3.5 and 3.6.

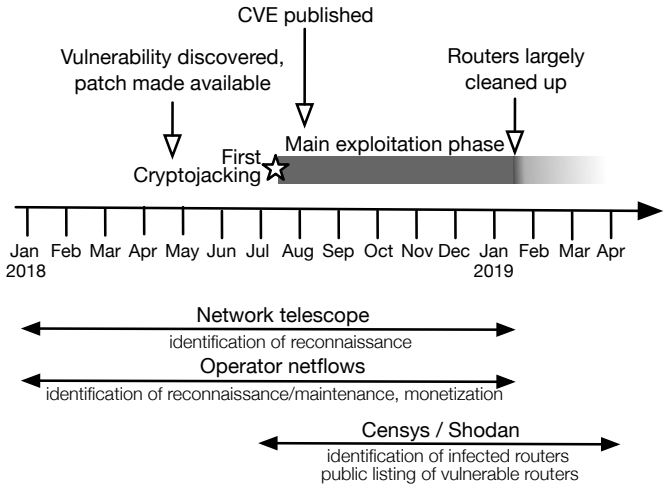


Figure 3.2: Timeline of the cryptojacking campaigns and the used datasets.

3.4.1. NETWORK TELESCOPE

In order to exploit routers using the WinBox vulnerability, the attacker must first know where vulnerable routers are located. This identification and localization could be done in one of two ways: either the adversary scans the Internet for open ports or banners that would identify the devices or obtains a list of devices.

To discover which adversaries are actively scanning the Internet for devices with the WinBox vulnerability, we rely on a large network telescope of three partially populated /16 networks, through which a total of approximately 130K dark IP addresses are monitored. In order to discover whether TCP port 8291 is open and to send a payload triggering CVE-2018-14847, adversaries first need to complete a TCP handshake. This ensures that perpetrators cannot spoof their source IP. Otherwise, the handshake couldn't finish and reveal the location of the adversary or a potential proxy. The telescope collected approximately 21.7 TB between January 2018 and January 2019, out of which only a small part of 1.6 GB was probed on port 8291. The size of the used telescope provides tight approximations of network activity estimations, as shown in [21].

3.4.2. ACTIVE SCANS OF CENSYS & SHODAN

In addition to § 3.3, the exploitation through the rewriting proxy was unusual as it unnecessarily exposed the Web page to the Internet instead of just presenting it to the users on the inside. Since RouterOS allows both port 80 and port 8080 to be used by an HTTP proxy, an Internet-wide survey of these ports made it possible to discover which MikroTik routers are currently infected as they are serving the proxy page, and based on the embedded *siteKey* track who currently "owns" the device.

Table 3.1: Summary of the datasets and their usage in this analysis.

Dataset	Time frame	Size	Usage in analysis
Telescope	Jan 2018 – Jan 2019	1.6 GB	Adversarial identification through port scanning (3.5.1)
Censys	Jul 2018 – Apr 2019	43 GB	Adversarial targeted scanning (3.5.1), Infections and re-infections (3.5.2), System architecture (3.5.3), Monetization configuration (3.5.4), Revenue and ecosystem (3.6)
Shodan	Jul 2018 – Apr 2019	236 GB	Adversarial use of public datasets (3.5.1), System architecture (3.5.3)
NetFlows	Jan 2018 – Jan 2019	3.2 TB	Characterization of port scanning (3.5.1), System architecture (3.5.3), Evolution of monetization (3.5.4), Maintenance patterns (3.5.5), Revenue and ecosystem (3.6)

Table 3.2: Regular expressions used to detect mining code in the Censys datasets.

Miner type	Regular expression
Coinhive	`new CoinHive\.Anonymous | authedmine.com/lib/ coinhive.com/lib/coinhive.min.js | coinhive | cnhv\.co`
Cryptoloot	`CRLT\.anonymous | webmine.pro/lib/crlt.js | cryptoloot verifier.live/lib/crypta.js | crypta`
Coinimp	`coinimp | new CoinImp.Anonymous new Client.Anonymous | scrip | srcips | priv\.su freecontent.data | freecontent.date | hashing\.win hostingcloud.science | freecontent.stream`
Omine	`omine\b | omineID`
Webminer	`coinwebmining.com | cwm\.js | serv1swork mining711 | gazanew`
Mineralt	`ecart\.html\?bdata= | amo\.js\" | mepirtedic\.com gramombird\.com | tulip18\.com | mineralt\.io dinorslick | istlandoll\.com | feesocrald besstahete\.info | nexioniect\.com | pampopholf\.com`
Coinhave	`minescripts\.info`
Coinpot	`coinpot | wait\.php`
Monero-mining	`perfekt`
Webminepool	`webminepool\.com/lib/base\.js | WMP\.Anonymous`
Obfuscated	`147\.135\.234\.198 | 91\.134\.24\.238 unescape | pastebin`

CENSYS

To trace infections and their evolution, we thus rely on Censys [61], which scans and archives the responses of all IPv4 addresses on a number of common ports, among them 8080 and 80. As the vulnerability was exploited for cryptomining in July 2018, we retrieved these Internet surveys twice a week from July 2018 until the end of the study in April 2019. We identify a router as a MikroTik system if the proxy header was set to `MikroTik HttpProxy` and mark it as infected when it contains scripts or code for cryptomining. The regular expressions used for this detection step are listed in Table 3.2 and resulted in a dataset of 43 GB. This yielded a total of 1,452,550 unique IPs belonging to an infected router at some point during the study.

SHODAN

A second service that scans devices for open ports is Shodan [218]. The service additionally extracts banners to link them with known vulnerabilities, enabling easy searches for specific devices and credentials. Given the Internet surveys of Censys, we queried Shodan's databases and recorded when a particular IP that could be identified as compromised due to the HTTP proxy page, including a cryptomining script, appeared in Shodan's database. Therefore, we queried the host information endpoint of Shodan's API with the history flag enabled and searched for the timestamp when Shodan encountered the open proxy ports for the first time in their crawls and listed them with the annotation `mikrotik` or `routeros` in their public search results. For the 1.4M routers, this dataset of historical open ports and services comprised 236 GB of records.

3.4.3. OPERATOR NETFLOWS

While the aforementioned datasets provide insights into vulnerable devices and which routers are exploited at a given moment, these data sources do not reveal anything about the scale of the operation and how concretely the infrastructure is managed and controlled. In order to fill this gap, we analyzed NetFlows from the network of a Tier 1 operator between January 2018 and January 2019, which were collected at a 1:8192 sampling ratio at each of their edge routers. For the quantification of traffic volumes in § 3.6, the flow aggregates were scaled up by this sampling ratio.

ANONYMIZATION

While the IP addresses of vulnerable MikroTik devices are public knowledge as they appear in both Censys and Shodan, we need to ensure the privacy of users and their traffic during our study. For our analysis, we obtained NetFlow records for all connections from or to the 1.4M infected MikroTik routers in a tuple consisting of time, source and destination addresses and ports, as well as packet size, which allowed us to investigate when and how the routers made connections. The identity of the other endpoint is, however, irrelevant and was anonymized to a pseudo-random value. For this, the operator applied the CryptoPan algorithm [257] to the remote points of the NetFlows, which does prefix-preserving deterministic randomization of IPv4 addresses based on AES as a source of randomness. The algorithm was proven to be semantically secure by Xu et al. [257], and the key to the data randomization remained with the Tier 1 operator. The procedure was developed in collaboration with and approved by the operator's relevant departments.

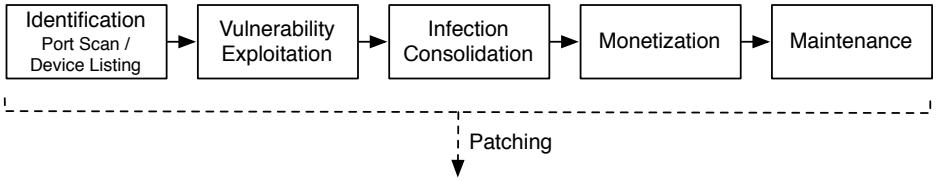


Figure 3.3: Life cycle of the vulnerable routers.

This protocol will thus allow an analysis of whether devices connecting and controlling the vulnerable routers are located, for example, in the same /24 network, but not which one. We can furthermore investigate whether there are specific, anonymized IP addresses that connect to multiple vulnerable or infected routers to do exploitation or quantify the number of hijacked flows due to source/destination port combinations but cannot tell the identity of these devices nor the destinations visited by the victims. In order to help the presentation of the results and elaboration on certain strategies and patterns, the subsequent discussion will include anonymized IP addresses, however these do not allow any inferences on networks except that addresses in the same netblock – for example a /24 – were also in the same subnet in the original trace. Whenever we use an anonymized IP address in the text, it will be printed in *italic*, while the publicly known and thus unanonymized IP address of an infected router will be shown in regular font.

3.5. ADVERSARIAL TECHNIQUES, TACTICS & PROCEDURES

In this section, we analyze the techniques, tactics, and procedures (TTP) adversaries use in the exploitation of 1.4M MikroTik routers and their subsequent abuse. We will split this discussion based on the stages in the life cycle of a router infection as shown in Figure 3.3. This life cycle begins with the identification of candidate victims, the exploitation of the vulnerability, and the methods used to gain a foothold and consolidate the infection. After a device is compromised, actors will install tools to monetize the exploited routers and perform maintenance, until the infected system is removed from the pool due to decommissioning or patching. As we will see in this section, each of the individual steps can be accomplished in a variety of ways, and we find adversaries using different techniques.

3.5.1. IDENTIFICATION

In order to gain a foothold on a machine, adversaries first need to know where exploitable devices are located. This also holds for vulnerable MikroTik routers, of which, according to market surveys, approximately 2M units were installed worldwide [215]. Routers are usually deployed in one of three ways on the Internet: (a) they are either provided by the Internet Service Provider (ISP) to the customer who uses the device to connect to the ISP 's network, (b) they are bought, deployed and operated by the customer to connect to the Internet, or (c) they are part of the network infrastructure of the ISP. As RouterOS was used across the entire MikroTik product line, we see vulnerable devices of all three types in practice.

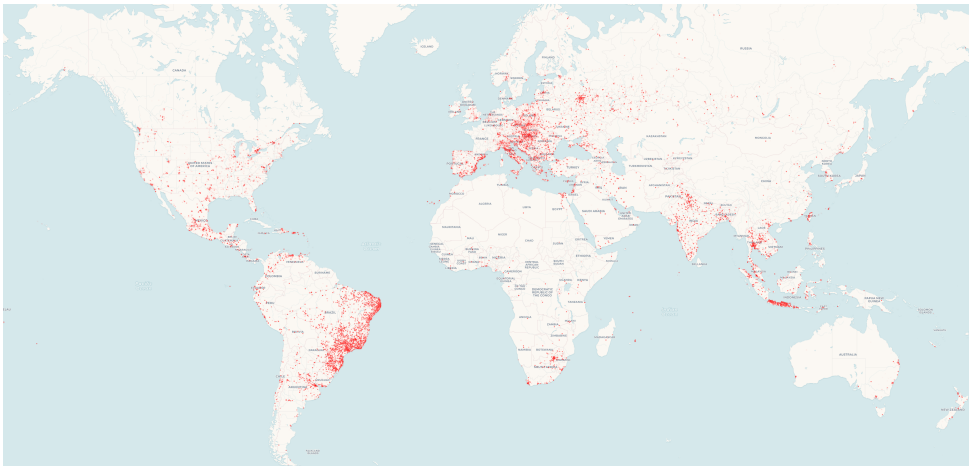


Figure 3.4: Geographical location of the MikroTik routers compromised during the study period.

Table 3.3: Top 10 most affected Autonomous Systems (AS).

AS	Count (%)	AS	Count (%)
Telekomunikasi Indonesia	55,082 (3.8%)	Cat Telecom	12,883 (0.9%)
Telefonica Brasil S.A.	33,589 (2.3%)	Rostelcom-AS	11,352 (0.8%)
TCI	21,357 (1.5%)	TOT-NET	11,136 (0.8%)
PTC-Yemennet	13,585 (0.9%)	UKRTELNET	10,993 (0.8%)
BSNL-NIB	13,046 (0.9%)	IR-THR-PTE	9,248 (0.6%)

Figure 3.4 shows a heatmap of all MikroTik routers that were exploited at least once during the study period, mapped to a geographic location by using the MaxMind GeoIP database [156]. The devices are very prevalent in select parts of the world, especially Brazil or Indonesia, where such a device responded at 29%, and 35% of all publicly accessible IP addresses of the largest operators in these countries, thereby indicating that these devices were provided by the ISP to the customers. Table 3.3 lists the number of compromised MikroTik routers for the 10 most affected autonomous systems and their share of the overall infected population. We can see that 136,659 exploited MikroTik routers could be linked back to the 5 most compromised ISPs. The heatmap also shows sparse deployments throughout the world, with clusters appearing in densely populated areas, proportionally to the number of IP addresses located in an area, suggesting that these routers were owned and operated by end customers.

DISCOVERY USING PORT SCANNING

To localize potential victims, adversaries could make use of port scanning to test remote IPs whether they have TCP port 8291, the port associated with the WinBox vulnerability, open. This reconnaissance could be done at different levels of granularity and sophistication: on the low end, attackers could blindly trawl through the entire Internet in a

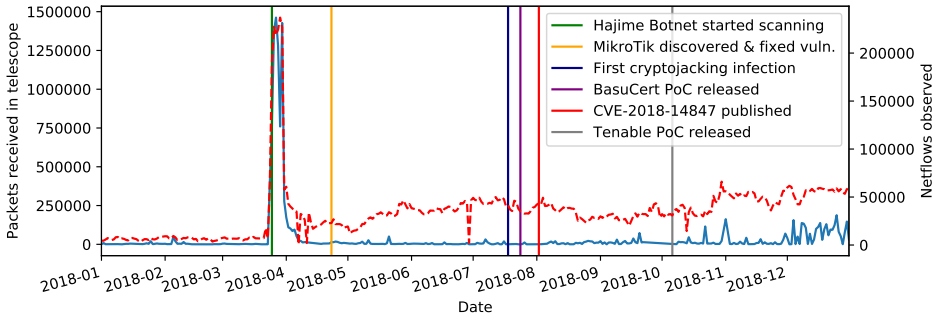


Figure 3.5: Packets received on port 8291 in our network telescope (in solid blue) and NetFlows observed (in dashed red).

horizontal port scan to discover any potential victim, albeit at the disadvantage of creating much noise and potentially being identified and blocklisted. A sophisticated scanner could, however, do some prior background research and determine in which networks large MikroTik installations exist as a result of these devices being used within an ISP's network or being given out to its customers.

We can differentiate between these types of strategies using the data provided by the network telescope and the general flow statistics of the Tier 1 operator. Figure 3.5 shows the absolute number of packets directed against port 8291 in our telescope, as well as traffic carried by the operator during 2018, aggregated by day. The vertical lines show important milestones in the lifespan and news coverage of the exploited vulnerability. On March 24, the average daily traffic towards TCP port 8291 exploded by 6 orders of magnitude as the Hajime botnet executed a short but concentrated horizontal scan for the port across the Internet [177]. On April 23, the vulnerability was discovered and patched by MikroTik, and the resulting news coverage only led to a very minor continuous increase in scanning traffic. This is interesting, as, for example, the media reporting around the Memcached DDoS vulnerability in early 2018 led to a major influx of actors and probing activity [83]. Starting in mid-July, the first cryptojacking installations started to appear in the wild, followed by a public proof-of-concept for the exploit. Finally, at the beginning of August, the CVE report was published in the National Vulnerability Database [178].

As we can see from the graph, the general characteristics of the telescope and NetFlow traffic resemble each other. Both record the same sudden increase in network traffic due to the Hajime botnet at the same moment and with a similar magnitude, demonstrating that the botnet initiated an unspecific worldwide trawl for the vulnerability. While after this burst, the telescope traffic returns to business as usual, aside from selected worldwide scans, we see in the NetFlow data that geographically targeted scans – not targeting our network telescope – immediately followed and continued to run until the end of the observation period. As the number of infections started to rise in December 2019, we observed increased worldwide scanning activity as both our telescope and NetFlow data reported more connections toward port 8291.

Out of a total of 1.7M IP addresses that probed the three /16 network ranges in our

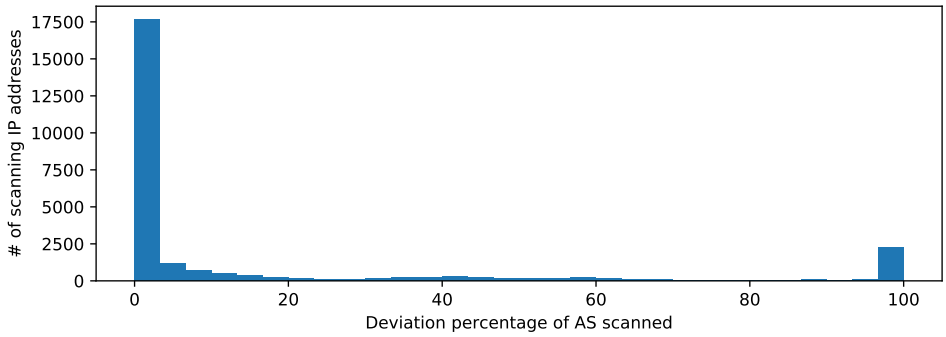


Figure 3.6: Histogram of the specificity of scans for port 8291.

telescope as well as the rest of the Internet during the late March burst, only 124K IPs continued to probe specific parts of the Internet for router vulnerabilities. This seems to indicate that the scanners used the data collected from previous tests (as our passive monitors would not respond to 8291), or that additional knowledge – such as the popularity of MikroTik in specific parts of the world – is used to steer the search. In order to determine the specificity of these scanners, we compared the traffic distributions of the Tier 1 operator towards all autonomous systems (AS) with the traffic distribution for the anonymized scanning source IP addresses. This relative comparison accounted for the fact that the operator would not be part of an exact random sampling of all worldwide traffic flows but that due to BGP policies and specific IXP and PoP presences, certain autonomous systems would be preferred. From this relative comparison, we can determine whether sources showed specific preferences for select networks or scanned the Internet non-discriminately. Figure 3.6 shows a summary of all scanners as a histogram of the scanners' deviation from the expected non-discriminatory baseline. As we can see in the graph, there exist three basic behaviors: the bulk – which is also visible in our telescope – targets the entire Internet unspecifically, a smaller but significantly sized group that specializes and concentrates the scan on a specific AS, while a small portion of adversaries scan a large but apparently curated list of destinations.

LOCALIZATION USING PUBLIC DATASETS

In addition to actively scanning and probing IPs on the Internet to test whether they are running RouterOS and are potentially exploitable, attackers could try to get a premade list of device IPs to connect to potential targets directly, for example, by searching on Shodan. To determine whether the attacker uses such services to locate vulnerable routers, we consider the moment Censys retrieved a proxy page from a router with a mining *siteKey* on port 8080 or 80, which means that at this moment the device was compromised. If, at that moment, the router was not yet listed in Shodan, the perpetrator must have found the vulnerable router by independently scanning for it. If, prior to the Censys publication, there was already a record in Shodan, the attacker could have obtained knowledge from this service.

When we track this relationship for every *siteKey* on the date it first appeared on the

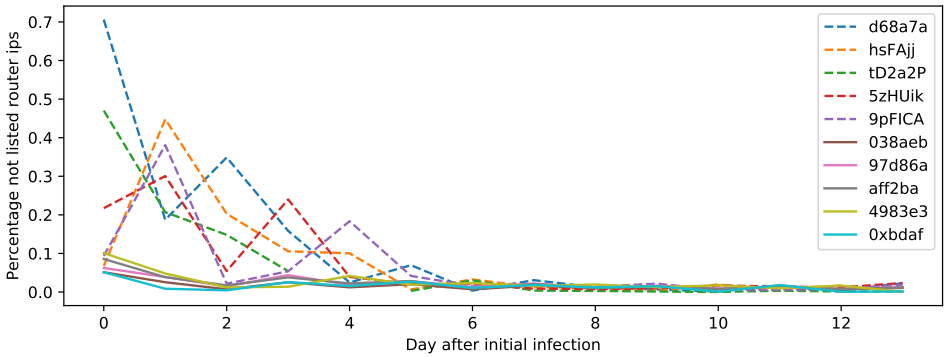


Figure 3.7: Percentage of infected unlisted routers per key.

1.4M routers, we find that 54% of the cases a new *siteKey* is installed on routers that were already listed in Shodan, whereas 29% of the new installations were derived from independent scanning. In the rest of the cases, too few routers were compromised with the same *siteKey* to significantly categorize them. Figure 3.7 shows the percentage of unlisted routers used by actors within the first 14 days of their activity. We clearly see two regimes. Innovators and early adopters such as *d68a7a* and *hsFAjj* which are shown as dashed lines (for *siteKey* emergence see Figure 3.11) all perform their own discovery, and start off with a high number of new, unlisted routers. This percentage drops over time, as the compromised devices are then included in Shodan. The long-lasting campaign *tD2a2P* starts out with 42% unlisted routers on its first and keeps adding unknown devices to its installed base for the months to come. On the other hand, we find a large number of campaigns that primarily feed off public lists to populate their setups. One of the most profitable campaigns, *6a9929*, had at its peak 13,815 routers infected simultaneously, almost exclusively drawn from public lists. As we will see in § 3.6, the degree of innovation is not a proxy for the amount of revenue these campaigns make – innovation does not always seem to pay off. To summarize this phase in the life cycle, there is a wide variance of MikroTik devices located all over the world, we have seen a steadily increasing interest in scanning port 8291 throughout 2018, and half of the newly installed *siteKeys* are installed on a router that was already listed in Shodan, whereas only 29% of all new installations was the result of independent scanning by the adversary.

3.5.2. VULNERABILITY EXPLOITATION

With the vulnerable routers identified, adversaries can trigger the vulnerability by sending a simple payload, as discussed in § 3.3.3. While the activities of the perpetrators on the devices cannot be inferred using our datasets, we can investigate patterns of adversaries to infect devices, and how infected devices are taken over.

INFECTIONS AND REINFECTIONS

From the previous discussion, we have seen in the NetFlow data and our telescope that a large number of IPs in our telescope and in the NetFlow data scanned for port 8291 and

that actors additionally used records such as Shodan to find exploitable targets. Once a device, however, appears in Shodan, it could already be infected due to a proxy service running on port 80 or 8080. This naturally raises the question of whether and how reinfections occur, in other words, whether actors are grabbing compromised devices from others or are updating *siteKeys* on routers they already "own".

Figure 3.8 depicts the transition behavior of the 1.4M routers between *siteKeys*, filtered to only include edges if more than 500 devices are taken over from the original "owner" by a particular new actor. The size of the circle is the number of routers that transition away from this *siteKey*, the thickness of the arrow and the color of a circle are the number of routers that are newly infected with a particular key. Since we can not retrieve the persona behind a *siteKey*, we assume in this section that every *siteKey* is a different persona. However, as we reveal in § 3.6.2, we have strong suspicions that this is not the case. In Figure 3.8, we see two types of transition behaviors First, we see *siteKeys* which draw their installation base from pools of already infected routers. An example of this is *4983e3*, which relies on lists of infected devices and then reinfects them with a new *siteKey*, something we could already infer for this *siteKey* from Figure 3.7. Second, we see *siteKeys* on routers being replaced in a specific sequence by another account. When the routers of different keys before and after the update share properties, for example, they are accessed by the same person or share infrastructure components, we can conclude that these are examples of an actor performing key rotation. The sequence from *iWDUFD* to *ByMzv3* to *aff2ba* to *ef18c8* shows an example of such a transition, which is visible as all routers in a node leave towards the same destination as can be seen in the identical color of node and arrow. A special form of these updates also appears in the application of obfuscation techniques. For example, the routers infected by the *siteKey 4983e3* were at some point being updated to the iframe code shown in Listing 3.2. The obfuscated JavaScript expression *0xbdaf('0x0')*, however, decodes to *4983e3*, so here, the transition is an evolution in technique rather than a progression of accounts. Account rotations, however, do not necessarily occur in chains, an example being the cluster on the right, where the routers originally mining for *hsFAjj* transition towards *SK_LCx* and *oDcuak*, and where a little over 15K routers shift back and forth between the receiving *siteKeys*. While the reinfection graph only displays the largest transitions for readability, there is a lot of change happening, especially in the long tail of the distribution. Overall, 55% of all routers are infected with more than one key, and 15% of all MikroTik devices had 5 or more *siteKeys* in 2018.

3.5.3. INFECTION CONSOLIDATION

After the adversary has been able to obtain the system credentials and activate the developer backdoor, root access is used to establish a foothold on the device. As described in § 3.3, the firewall configuration is changed, the proxy is activated, and additional files are downloaded to the system. We defer a discussion on monetization to the next section and discuss the infrastructure used to perform the scanning, logins, and loading of additional components.

```
<script src="https://xmr.omine.org/assets/v7.js"></script>
<script>OMINEId(\"4983e34ef01b4b579725b3a228e59e79\",\"-1\");
throttleMiner=10; </script>

<script src="https://xmr.omine.org/assets/v7.js"></script>
<script>var _0xdafb=['\\x34\\x39\\x38\\x33\\x65\\x33\\x34\\x65\\x66\\
    x30\\x31
\\x62\\x34\\x62\\x35\\x37\\x39\\x37\\x32\\x35\\x62\\x33\\x61\\x32\\x32
    \\x38
\\x65\\x35\\x39\\x65\\x37\\x39'];
OMINEId(_0xbdaf('0x0'),'\\x2d\\x31');throttleMiner=0xa;</script>
```

Listing 3.2: The original Omine infection on top, the obfuscated variant listed on the bottom.

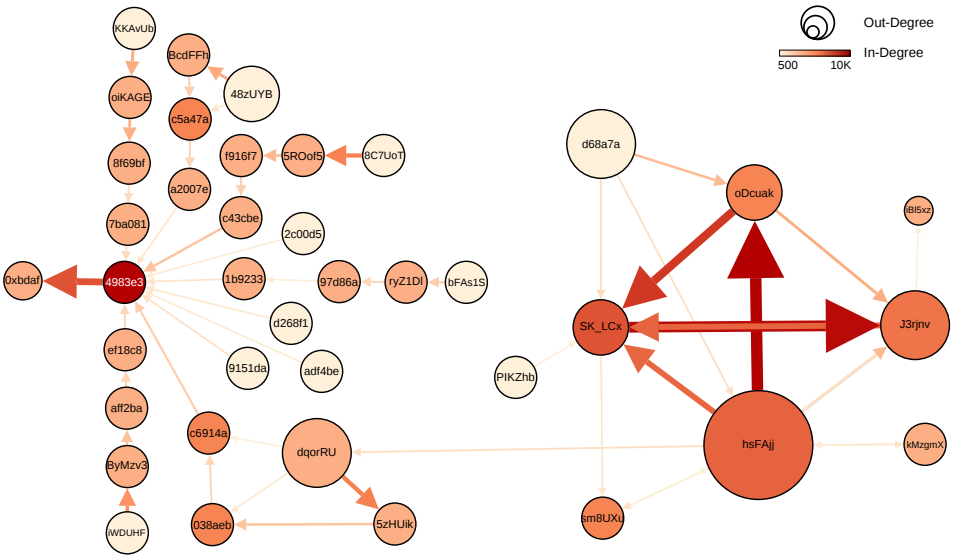


Figure 3.8: Reinfections of compromised devices with different keys with >500 overlapping IPs.

NODE TO NODE RECONNAISSANCE

Based on Censys and Shodan data, we obtained a list of infected devices over time and could, in the NetFlows, thus trace which anonymized IP addresses would connect to the WinBox service on vulnerable and infected routers. While the bulk of these connections came from a variety of anonymized IPs, 6.5% of the flows towards port 8291 were sent from infected MikroTik routers to other MikroTik routers. We observed 948 infected routers which were systematically scanning their local subnet for additional vulnerable routers on port 8291. While based on NetFlows it is not clear whether these infected routers only enumerate vulnerable hosts or also perform the compromise itself, we find this additional structural component noteworthy. Interestingly, this behavior was only implemented in geographic regions where MikroTik routers seemed to be rolled out structurally by ISPs, as we observed this behavior specifically in Brazil.

INFRASTRUCTURE

In August 2018, the first router infections spread throughout Brazil and were under the control of a sophisticated adversary. After successfully locating vulnerable MikroTik devices, it exploited the WinBox vulnerability and injected both a miner into the HTTP proxy page and installed a script that would fetch new updates and commands from a *staging server* on port 2008 every 30 seconds. These updates could involve changes in the miner service or a new *siteKey*. Using our NetFlow data, we have identified six of these staging servers in the subnet of *211.164.222.**, which confirms the research of [228]. We have identified that these staging servers are active from 26 July to 21 September 2018, and these servers have connected to 220 distinct infected routers during this period. The most prominent *siteKey* involved in making these connections was *hsFAjj*. However, our NetFlow data also shows that devices infected by *SK_LCx* and *oDcuak* begin to make contact with these servers towards the end of this period, suggesting a link between these *siteKeys*.

Based on the connection patterns of the compromised routers and the maintenance activities (which we discuss in § 3.5.5), we can deduct the system architecture as depicted in Figure 3.9. While a handful of infected routers are performing scanning and infections within the same prefix, compromised routers remain unconnected among themselves. They only have two flows in common: the connection on port 2008 to a handful of staging or Command & Control (C&C) servers, as well as SSH flows on port 22 from a shared origin. When a router is taken over, the new perpetrator does not seem to always aim to eradicate a previous infection after having replaced the proxy template and *siteKey*. In fact, we find numerous examples where the routers taken over by a different *siteKey* keep beaconing to staging servers associated with an unrelated actor, who shows no other commonalities or features with the new owner.

3.5.4. MONETIZATION

With the vulnerability triggered and a foothold on the routers established, the adversaries moved to the exploitation of the routers for monetary gain. Over the course of the study period, we observed the evolution of two monetization strategies. First, the routers are used as a (free) proxy service, and second, cryptomining code is injected into users' web browsing sessions.

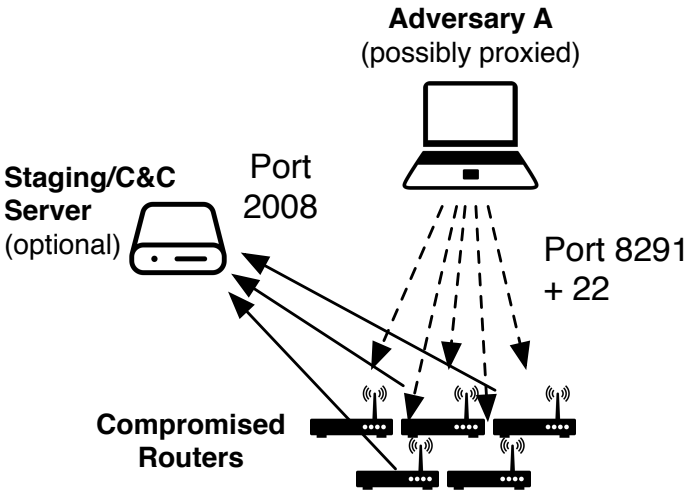


Figure 3.9: Schematic overview of the system architecture.

HTTP PROXIES

The first use case of the compromised MikroTik routers was the establishment of HTTP proxies. Here, traffic from a Web browser to a Web server is tunneled through the HTTP proxy, thus masking the IP address of the client towards the server. HTTP proxies are used as a basic variant of a VPN service, although application-protocol specific and with limited authentication options, if at all implemented. Starting from July 9, 2018, the first MikroTik routers were repurposed as HTTP proxies, which we identified from the emergence of large incoming traffic towards specific high TCP ports, namely 36551, 53281, and 58833. This use case remained, however, relatively rare, with only 3,216 of the total 1.4M infected routers being abused in this way. Interestingly, the usage as an HTTP proxy did not seem to serve a monetary gain, as within 3 days 95% of the routers for which these unusual spikes appeared were posted to free public proxy lists [198], and allowed a connection without user credentials. This usage was only relatively short-lived, as most were disabled within 40 days, at which point SOCKS proxies were spun up at TCP 4145.

SOCKS PROXIES

In contrast to HTTP proxies, SOCKS proxies work at the transport layer and forward traffic transparently with regard to the application layer protocol. This allows this proxy type to be used in combination with any application, thus extending the monetization potential. Shortly after the emergence of this new use case, the HTTP proxies on the MikroTik routers were replaced by SOCKS proxies, and 1,530 MikroTik routers remained in use as SOCKS proxies even until the end of our study. Further characterization of the NetFlows is not possible, as the application traffic itself would be forwarded inside the tunnel, and the router would rewrite the outgoing flow to an ephemeral source port. However, we do find that the exploitation as a SOCKS proxy was under the control of a

Table 3.4: Router "ownership" based on cryptomining *siteKey* and corresponding SOCKS proxy activity.

SiteKey	*hsFAjj*	*J3rjnv*	*SK_LCx*	*oDcuak*	*d68a7a*
% of all SOCKS traffic	53.6%	29.1%	7.8%	3.4%	1.2%

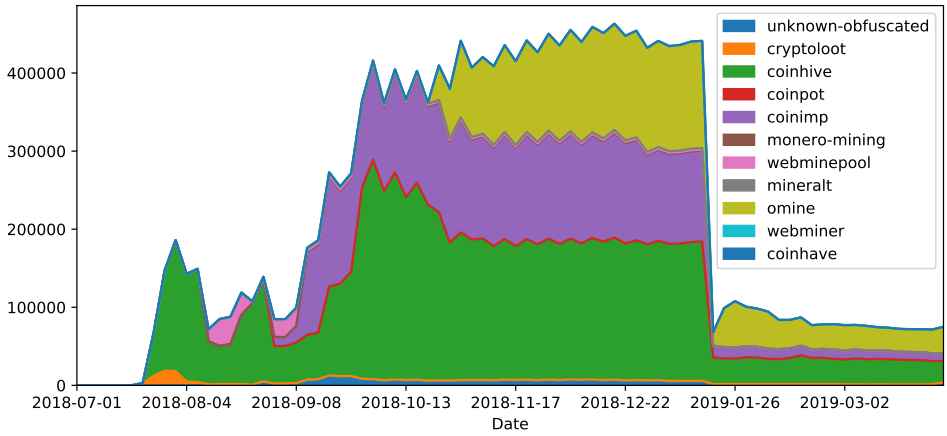


Figure 3.10: Evolution of cryptominers over time per service.

few and not deployed pervasively. We can conclude this, as the use of SOCKS proxies was never encountered alone, but only in combination with a cryptomining infection. As we discussed in the previous section, adversaries were routinely reinfecting devices and by changing the cryptomining *siteKeys* effectively snatching the devices away from their competitors. With the infection script reconfiguring the device, including firewall and proxy settings, we can thus assess that the "ownership" with respect to an active cryptomining would also indicate who had control over the SOCKS proxy at that point in time. As we discuss in the next section, we identified a total of 140 cryptomining keys on the 1.4M MikroTik routers, but as shown in Table 3.4, only five *siteKeys* were in use on a router whenever the device was proxying traffic. Their impact is, however, huge: more than 95% of all MikroTik SOCKS activity that originates from MikroTik routers is the result of proxies operated by these five *siteKeys*, with *hsFAjj* being one of the early adopters of MITM-based cryptomining. The small number of *siteKeys* related to SOCKS proxy activity suggests a relation between those *siteKeys*, as others do not exhibit this behavior.

CRYPTOJACKING PROXIES

While the usage as HTTP proxies was not commercialized and only a few actors repurposed a limited number of devices as SOCKS proxies, a large number of actors engaged in cryptojacking user connections, and a total of 140 different cryptomining *siteKeys* being installed on the routers during the study, with a maximum of 106 different *siteKeys* being active at the same time.

Table 3.5: Top 10 largest MITM cryptojacking campaigns identified.

SiteKey	Service	Total infected	Max. concurrently	Date first seen
hsFAjj	Coinhive	223,844	167,182	Jul 21, 20218
4983e3	Omine	117,502	64,539	Nov 3, 2018
f6c7f3	Omine	102,241	36,059	Jan 23, 2019
tD2a2P	Coinhive	71,513	61,835	Aug 22, 2018
oDcuak	Coinhive	55,437	47,310	Aug 1, 2018
48zUYB	Coinhive	52,181	26,122	Sep 12, 2018
dqorRU	Coinhive	50,566	27,808	Sep 15, 2018
9pFICA	Coinhive	50,376	25,928	Sep 15, 2018
BOvlp3	Coinhive	49,640	22,921	Sep 15, 2018
8C7UoT	Coinhive	47,981	24,773	Sep 15, 2018
Total		1,452,550	460,618	

MINING SERVICES

Figure 3.10 depicts the number of infected routers over time, categorized by mining application used. As we see in the figure, MITM-based mining started out based on Coinhive, which was at that time the obvious choice to be introduced in the MITM vector as it was the first service for cryptomining and already widely deployed in website-based mining [16, 127, 199]. Starting in mid-September, this homogeneity shattered with first the emergence of CoinImp and, later on, Omine, all taking on approximately equal market shares, which led to a peak of cryptojacking activity on 19 December 2018, as 460,618 routers were actively cryptojacking concurrently. This activity continued relatively unchanged until 26 January 2019, when mining activity suddenly disappeared from the bulk of infected routers. The distribution of miner applications between Coinhive, CoinImp, and Omine remained relatively unchanged.

Interestingly, *siteKeys* found as related in previous analyses, do not necessarily use the same mining application, possibly to avert risks from accounts becoming frozen by an individual cryptomining service. Despite risk being shared across mining services, several actors also spread out their activities across multiple *siteKeys*, as can be inferred when the same maintenance hosts connect to routers with multiple *siteKeys* (as we will show in § 3.5.5). These movements between mining services and the market shares of CoinImp and Omine might also be explainable based on fees: while Omine charges a 2% fee and CoinImp is entirely free, Coinhive takes a 30% cut.

EVOLUTION OF SITEKEYS

Figure 3.11 shows the evolution of *siteKeys* installed on MikroTik routers between July 2018 and April 2019, ordered by the time they were first encountered on a router. The size of the circle indicates how many routers this *siteKey* was installed on a given day. We can see that MITM-based cryptomining was pioneered by three *siteKeys*: first, *d68a7a* who emerged first but besides a small peak remained only a minor player. Second, *hsFAjj*, who followed one week later, temporarily controlled 70% of all infected routers and introduced new strategies for controlling and otherwise monetizing the routers, remain-

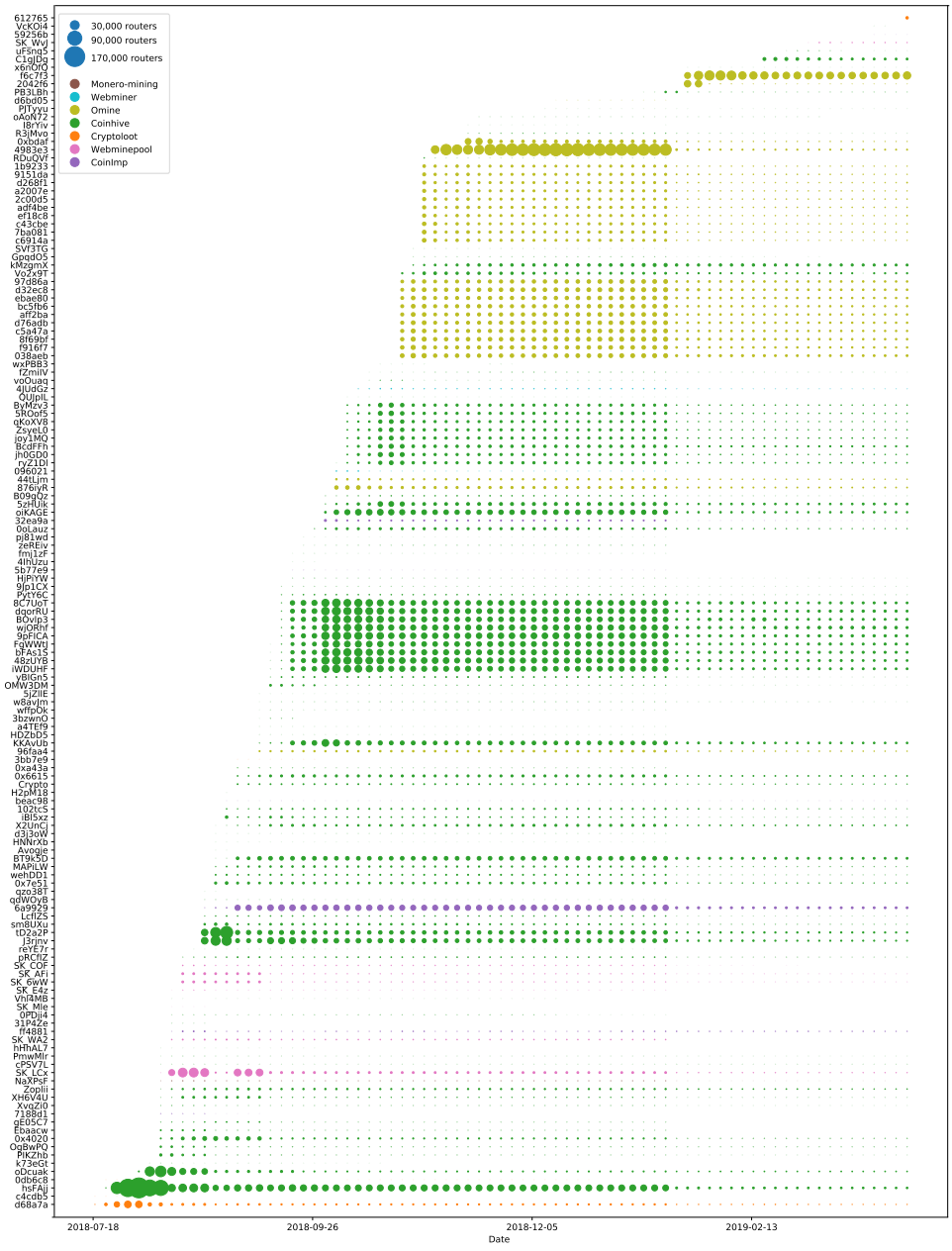


Figure 3.11: Evolution of detected *siteKeys* over time, colored per mining service.

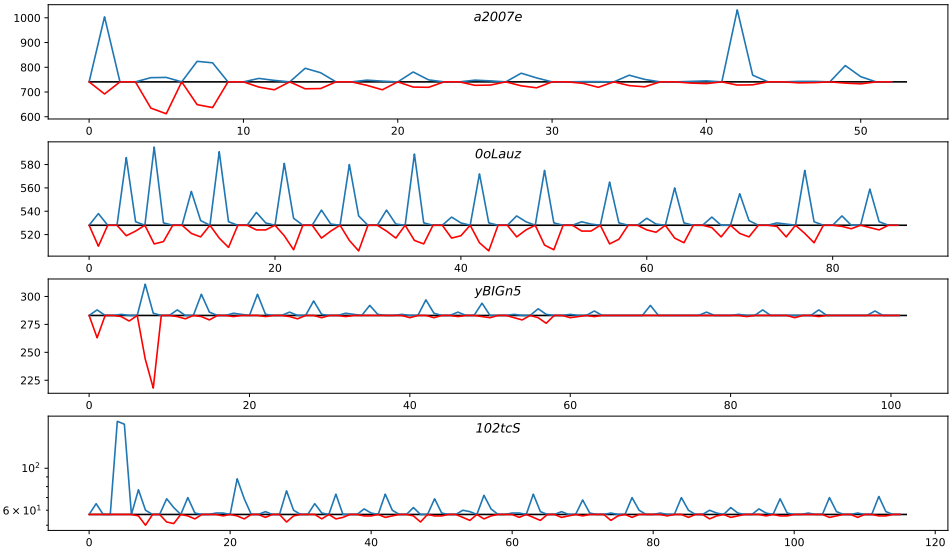


Figure 3.12: Additions/deletions over time per *siteKey*.

ing a steady force until the general decline. Third, *oDcuak*, like the first mover (*d68a7a*), experiences a small surge followed by a steady but comparatively low-volume activity.

Approximately one week after these first movers, a large number of new *siteKeys* started to appear, frequently co-emerging in groups that stayed relatively similar in size. Four sequential blocks of 10 *siteKeys* can be clearly observed, two of them using Coinhive and the other two using Omine as their mining service. While all other *siteKeys* never reach the same size as *hsFajj*'s initial deployment (167,182 infections), each of them is able to hold control over up to 64,539 routers at a time. While we find a total of 1.4M routers to be vulnerable and, at some point, infected, the perpetrators are never rolling their cryptojacking infections out to all potential victims simultaneously. Instead, we see a constant flux, with new routers being infected so that the mining deployments stay consistent in size. This is necessary, because once infected, most of the routers are patched quickly, as shown in Figure 3.13. This figure shows the cumulative density function (CDF) of the number of days a router is infected on a logarithmic scale. We see that 50% of the devices are patched within 18 days after compromise, whereas 30% of the devices remain active for more than 50 days, urging actors to constantly replace disappearing routers to maintain their installation base.

This is best observed when we look at the *siteKeys* in Figure 3.11 that remain relatively constant over time. Four of these *siteKeys* are depicted in Figure 3.12 with respect to daily additions and removals from the pool, indicated in blue and red respectively, starting from the day the *siteKey* first became active. This behavior, as well as the sets of *siteKeys* that appear together, might indicate a strategy to offset risk. If a particular *siteKey* gets blocked by a mining service, others will still generate profits. The same might hold for the deployment size in general, where an all-out operation from becoming too greedy could

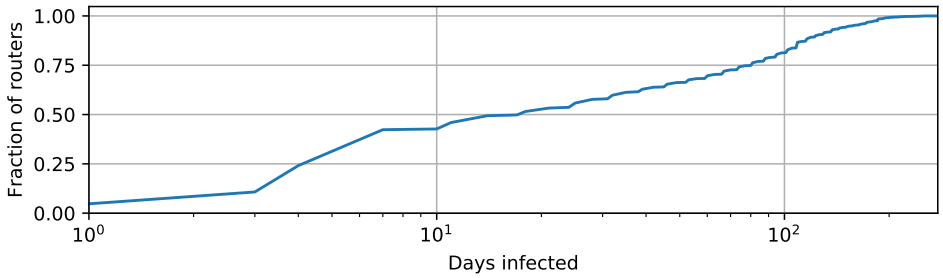


Figure 3.13: CDF of the infection duration per IP address.

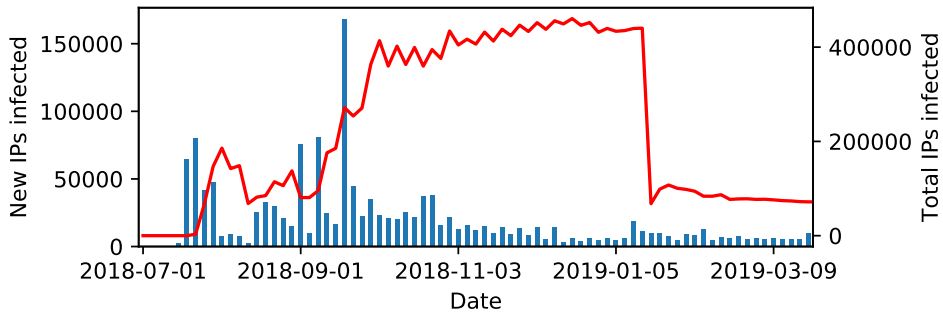


Figure 3.14: New and total of IP addresses infected per day.

lead to increased press coverage and faster cleanup of the vulnerability than maintaining a smaller infection size and thus lower profile. This diversification, however, stops from December 2018 onwards, where we see that most actors no longer replenish routers lost. This might be explained by Monero's significant drop in value, which decreased by 60% from early November until a month later.

A SUDDEN DROP IN MINING ACTIVITY

Indeed, we observe a steady decline of new devices that are added to the pool from November 2018 onwards, as shown in the bottom of Figure 3.14, which leads to a flattening out of the overall installation base. As we have already seen in Figure 3.10, the ecosystem of router-based cryptomining drastically changes in late January. Most apparent is the major drop in participating devices, approximately 87% of all infected routers disappear, which affects the installation base of all *siteKeys* across all autonomous systems and countries. While such a large and universal movement would indicate some external trigger or cause, we could not find any evidence of a coordinated cleanup action, for example, by an ISP or a grey hat hacker (aside from one who has taken credit for patching 100,000 routers in November 2018 [37]). Additionally, we contacted Censys whether they had made any changes to their crawling strategy, but that was not the case. After the sudden cut, we also see a rotation of remaining actors towards new *siteKeys*,

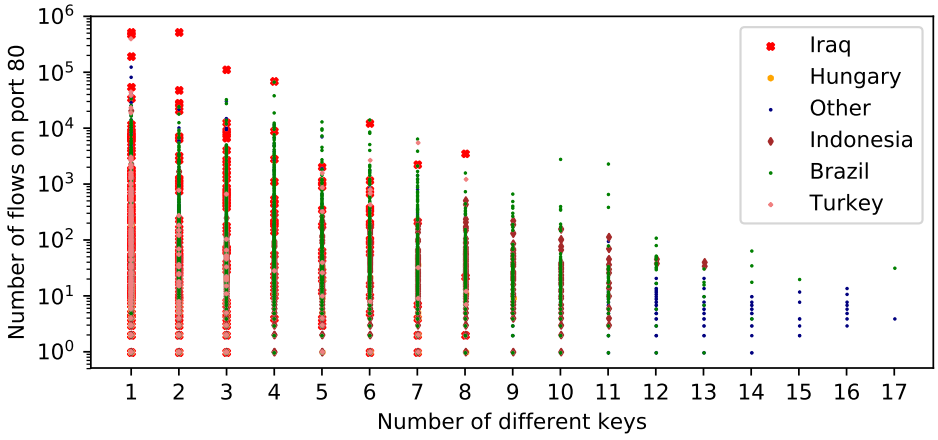


Figure 3.15: Relation between the number of flows to port 80 and the number of keys per router.

where the new *siteKey f6c73* partially takes over the efforts of *4983e3*, however only a few continue to re-establish their activities and forego previous practices, whereas *f6c73* is responsible for most new infections.

GEOGRAPHICAL FOCUS

Based on the heatmap in Figure 3.4 and the large deployment of MikroTik devices in certain autonomous systems as shown in Table 3.3, we have seen that a number of countries seemed prime candidates when looking for MikroTik devices, which would logically mean that advanced adversaries should focus their activities there. As RouterOS is used in both consumer devices and carrier-grade routers, we would naturally expect some devices to be more lucrative than others, immediately posing the question of whether reinfection of devices – in other words "stealing" routers – would primarily occur in popular areas and target those devices where a lot of money could be made.

Figure 3.15 shows the number of *siteKeys* as a function of the amount of NetFlows on port 80 this router processed during its infection. Counterintuitively, there is no trend that high-value targets are more fought over than low-value ones. Especially routers with much traffic tend to stick with just a low number of *siteKeys*. This is surprising, as a cryptomining operation on a large router would clearly affect more people, lead to more complaints, and thus logically faster patching. The lack of a fight for high-grossing routers can, however, partially be explained based on the location of the routers, indicated by the color of the data point. While routers in Indonesia and Brazil – the hotspots of the infection – cover the entire spectrum and are changing keys considerably, the most stable infections – and the highest-grossing ones for that matter as we will see in § 3.6 – are in countries that do not appear anywhere near the top in MikroTik installation counts, for instance, 6 out of the 10 most grossing routers are located in Iraq. This means that actors targeting niche markets accomplished much more valuable deployments, as these routers mined longer for them.

3.5.5. MAINTENANCE

When we look at the life cycle of a malware infection, for example, a botnet, after the initial exploitation, the compromised device remains in contact with the perpetrator or a C&C server to download additional components or receive new configuration. While we would expect a similar behavior for malware targeting routers, we saw little evidence for post-compromise maintenance operations.

CONFIGURATION ACCESS AND PERIODIC UPDATES

As a *siteKey* is directly linked to a particular actor, we analyzed whether any connections were made between an endpoint and the group of routers that were at a certain moment compromised by the same *siteKey*. Using the association rules methodology described by Agrawal & Srikant [1], we have searched for maintenance patterns where specific *siteKeys* have a large probability to coincide with a specific anonymized IP address or port number, as maintenance would likely be performed from a set of C&C servers or from the attacker's PC. As connections to port 22 (SSH) and 23 (Telnet) in NetFlows are also caused by prevalent port scanning, we differentiate between port scanning and active SSH sessions in NetFlows based on the packet size and only include connections with a confidence c and support s of at least 40% among our router/*siteKey* set. In other words, we require that at least 40% of the infected routers had been contacted by a common origin while being significantly present in the data.

We have observed maintenance connections on port 22 (SSH), which was only pursued by the actor(s) responsible for routers infected with one of three *siteKeys oDcuak*, *SK_LCx* and *hsFAjj*, while other strains and actors do not seem to deploy such coordinated access. Surprisingly, routers with any of these *siteKeys* were in contact with the same remote host at a given moment in time, strongly suggesting that the *siteKeys* were actually related to the same persona. In addition, when a new IP address appeared to make contact with the compromised devices, routers with all three *siteKeys* were always contacted by the same source. For example, routers with these *siteKeys* made SSH connections to *236.197.108.8* between 3 and 20 August 2018, while between 11 and 14 August 2018, these routers were contacted by *236.247.130.64*.

Each of these IPs seemed to employ automation, contacting routers either at midnight or during the timeframe of 16–19 hours. Besides these IPs, almost no evidence of scripted interactions between a controlling source and the infected routers has been found, which would be evident from a large number of connections being made at the same time, or sequentially within a short time period. In total, we observed 5 IPs making such common connections over time, matching our earlier observation about the link between the aforementioned three *siteKeys* as discussed in § 3.5.4.

3.6. DISCUSSION

The analysis of the tactics, techniques, and procedures of the actors demonstrated different levels of sophistication. This section translates flow volume into a revenue estimation per campaign, and finally, we review the previous findings and use them to describe the ecosystem of cryptojackers and their differences in sophistication.

3.6.1. QUANTIFICATION OF REVENUE

Previous sections already suggested that MITM-based cryptomining operates at an entirely different scale than previously reported attack vectors. This is due to three reasons:

1. The volume of compromised entities is much higher. Instead of a few thousand websites [29, 101, 199, 205], here a total of 1.4M infected routers is involved. Instead of mining on the web browsers of the users who visit one of the select infected websites, the MITM attack vector through routers would greatly amplify earnings, as cryptomining is introduced into *any* web page visited by *any* user connected behind an infected router.

2. MikroTik uses the vulnerable RouterOS on consumer-grade and carrier-grade devices. A carrier-grade router will likely serve significant user populations and, thus, within a short time, amass large volumes of revenue.

3. While 30% of all website-based cryptomining is removed within 15 days [101], we find that 30% of the MITM-based mining remains active for more than 50 days. Although routers are often patched quickly, the pool of vulnerable devices is so large that it barely affects the installation base.

In this section, we extend the previous results towards a quantification of adversarial revenue per *siteKey* using this new attack vector. Unfortunately, as we have shown in § 3.5.4, most mining is deployed through a cryptojacking service, such as Coinhive or Omine. This prevents us from performing a similar analysis to that of Huang et al. [103], who queried the (Bitcoin) mining pools directly to estimate the profits of a campaign. In our analysis, only 3 of the 140 discovered *siteKeys* were mining directly in a mining pool. However, we can leverage our datasets and use the same method established by Konoth et al. [127] to conduct a quantification for a direct comparison with website-based mining but make some adjustments for the shifted attack vector. For their analysis, Konoth et al. built a three-step estimation model:

- *Estimation of monthly visitors and visit duration:* They estimate visitor count and the average time spent for the 1,705 cryptomining websites they detected to be cryptojacking based on visitor statistics from SimilarWeb [220].

- *Average computing power of visitors in hash rate per second:* Cryptocurrency is mined during the visit to the website. They measured the hash rate of two desktop CPUs and 16 mobile devices and determined an average rate of 40.5 and 14.56 per second, respectively. Afterward, information on MineCryptoNight [162] is used to convert that to XMR/s.

- *Current value of cryptocurrency:* The overall mining power of the visitors is then mapped to and monetized in Monero cryptocurrency, which was valued at $\$253/XMR$ at that time. Based on this value, the top 10 grossing actors generated an overall revenue of some $41,000 per month.

In the following analysis, we are following the same equation:

$$\text{traffic [\# flows]} \times \text{avg. time [s]} \times \text{mining rate [XMR/s]} \times \text{value [\$/XMR]} = \text{profit [\$]}$$

Table 3.6: Cryptojacking revenue estimation parameters.

Parameter	Methodology in [127]	This study
Number of visitors	SimilarWeb estimations	# of NetFlows on port 80
Average hashing rate	SimilarWeb estimations	desktop / mobile: 25 H/s
Monero market value	$ 253 as of May '18	$ 253 for equal comparison
Time on website	SimilarWeb estimations	Average, 1st / 3rd quartile

but adjust them for the specific attack vector observed. First, our NetFlow traces allow for an extrapolation of the actual number of HTTP connections on port 80, and we attribute the count of flows to the revenues of a *siteKey* installed on the proxy page at that time. While the embedded miners also work for iframed-HTTPS connections, we did not find evidence that this attack was pursued in the wild. This will thus be a lower bound on the amount of traffic.

Second, Konoth et al. estimated average visiting times for each of their 1,705 detected websites using SimilarWeb data, but the MITM attack works across all pages of the Internet. As the actual end point of the outgoing connection has been anonymized for privacy, we can approximate the average visiting time as we query the average visiting duration of websites listed in the Alexa Top 10k – the 10,000 most popular websites – on SimilarWeb. The average visiting time for these websites is 293 seconds. For our calculation, we will work with three values for visit duration to provide a range of the revenues made by the attackers: the average visiting time, as well as the first and third quartiles of visit durations. Yet, the highly conservative estimation based on the first quartile already highlights the magnitude of this new attack vector. Table 3.6 compares the parameters used in [127] compared to our study.

Third, Konoth et al. [127] also used SimilarWeb data to estimate the hashing rate for both mobile and desktop visitors, being 14.56 and 40.5, respectively. We estimated the hashing rate based on the desktop/mobile device ratio found across the Internet as a whole, which is listed in [62] as 0.58, resulting in a weighted hash rate of 25 H/s. Since we want to compare the profitability of MITM-based to website-based cryptojacking, we could either compare the amount of Monero mined or translate the Monero amount into more intuitive currency such as USD. Currency exchange rates are, however, volatile, and in between Konoth's May 2018 study and our study, the average Monero price dropped during August and December 2018 to $92.2/XMR. To compare both attack vectors side by side, we thus use the same exchange rate as in [127], which still makes a fair comparison, as the decline would have equally scaled down the revenues attackers could have generated using website-based mining during our observation period. Even if we scale the revenue down with the declined value of Monero, the MITM-based revenues would still be a factor of 10 higher than the website-based earnings made half a year earlier.

Based on the parameters chosen above, Table 3.7 shows the estimated monthly revenues for the top 10 grossing actors for the average visit duration on the Alexa 10K, as well as the first and third quartile. As we can see, in the average case, the top 10 campaigns total a profit exceeding $1,200,000 per month, in which the highest grossing *siteKey* earns $222K. To put this into perspective, the 10 most successful campaigns that are deploying

Table 3.7: Estimated monthly revenue of top 10 grossing actors based on the average visit duration on the Alexa Top 10K, as well as the first and third quartile according to SimilarWeb.

SiteKey	Total # routers	First quartile 2'27" stay	Median 4'53" stay	Third quartile 6'19" stay
48zUYB	52,181	$111,447.18	$222,136.22	$287,336.61
6a9929	30,135	$97,626.82	$194,589.52	$251,704.53
8C7UoT	47,981	$90,532.54	$180,449.21	$233,413.82
BOvlp3	49,640	$82,573.82	$164,585.92	$212,894.42
4983e3	117,502	$70,017.28	$139,558.26	$180,520.75
FgWWtJ	39,384	$50,719.01	$101,092.99	$130,765.33
J3rjnv	45,934	$40,551.39	$80,826.92	$104,550.86
hsFAjj	223,844	$35,396.11	$70,551.44	$91,259.37
BT9k5D	8,459	$31,494.11	$62,773.97	$81,199.10
wjORhf	42,342	$27,671.96	$55,155.67	$71,344.70
Total top 10		**$638,030.22**	**$1,271,720.11**	**$1,644,989.49**

cryptojacking by installing miners on the websites themselves (for example, by hacking the site) were reported by [127] to yield monthly revenues of $41,000. Cryptojacking through an MITM attack on routers is thus a factor of 30 more lucrative than previously observed attack vectors, and the most successful MITM actor earns 5x more revenue than the top 10 website-based cryptojackers combined.

In our analysis above, we have seen the different roles the actors have played in the development and rollout of this attack vector and the different levels of innovation they have embraced. Curiously, we find that innovation and a first-mover advantage do not manifest in earnings. The actor operating *hsFAjj*, who was among the first, dominated proxying and controlled extensive infrastructure, did not translate this advantage into earnings at the same rate as, for example, the key *6a9929*, who would pick up data on vulnerable routers from public lists to roll out infections. Also, the number of infected routers is not necessarily an indicator of the amount of revenue an adversary has generated, as the size (and thus also the type) of the router matters more than the number of compromised devices. Table 3.7 also lists the total number of routers a particular *siteKey* ever had under its control during the 10-month study, and we clearly see that the volume of routers is not an adequate predictor of monetary success. Another unexpected story emerges when we look at the routers that are providing the most revenue. Out of the top 10 most grossing routers, 6 are located in Iraq, and one each in Turkey, France, Brazil, and the Netherlands, which is counterintuitive, looking at the worldwide distribution of MikroTik installations shown earlier in Figure 3.4.

3.6.2. CHARTING THE ECOSYSTEM OF ACTORS

Examining the life cycle of router infections, we observe different levels of sophistication in every stage. In the identification stage, we notice a clear distinction between *siteKeys* installed as a result of scanning and infection based on public sources, such as Shodan. In the exploitation of the routers, we observe a constantly changing landscape

in which actors regularly infect new devices and steal from each other. After infection, a limited number of actors demonstrate a high level of sophistication by setting up an infrastructure. To monetize the hijacked routers, actors initially set up HTTP proxies but subsequently increased their revenues by installing SOCKS proxies with cryptojacking scripts. The used cryptomining scripts diverge to multiple services, and we notice a continuous flow of router infections and removals. Clear geographical differences in mining characteristics are identified, where Brazil and Indonesia are the most infected, while Iraq seems to have the most lucrative infrastructure to infect. Observed maintenance patterns show that specific, anonymized IPs can be linked by behavior to *siteKeys*.

RELATING ACTORS AND SITEKEYS

Based on the results of the different independent components analyzed in the previous sections, we are able to link certain *siteKey*s to each other and/or to IPs. To start with, three *siteKeys hsFAjj*, *SK_LCx*, *oDcuak* show similar behavior as the same infrastructural patterns can be found on routers infected with these *siteKeys*, as well as regular contacts with the same set of attacker IPs for maintenance over SSH. Figure 3.8 confirms this hypothesis by showing numerous routers transitioning between those *siteKeys*. Interestingly, the analysis of SOCKS traffic also links *J3rjnv* to this set. Additionally, this figure depicts the sophistication level of the actor behind *siteKey 4983e3*, as this actor hijacks vulnerable routers infected with numerous other *siteKeys*, but subsequently changes his own *siteKey* to a masked variant, as listed in Listing 3.2. Revisiting Figure 3.11, which shows 4 clear sequential blocks of 10 *siteKeys* having similar installation sizes and evolutionary patterns. This, in combination with the aforementioned figure, which shows 5 clear *siteKey* transition chains, an even larger number of *siteKeys* can be linked to one single adversary. By following each *siteKey* within these transition chains in Figure 3.8, we noticed that these transitions resemble transitions between the sequential blocks in Figure 3.11. All the *siteKeys* in the transition chains are located inside these blocks in the same sequence. For each of the *siteKeys* inside these four blocks, the first two blocks (highlighted in green in Figure 3.11) use a Coinhive miner, with the uncommon option `CoinHive.FORCE_EXCLUSIVE_TAB` enabled, and the latter two (highlighted in yellow) use Omine as a mining service. Additionally, all 40 mining scripts within these blocks were set to the same throttle value of 0.1. As a result, this common behavior across multiple *siteKeys* strongly suggests that we can thus link these 40 *siteKeys* to one single actor.

3.7. CONCLUSION

In this chapter, we have reported on a new attack vector for cryptojacking, which does not infect websites but compromises Internet infrastructure itself. This vector greatly overshadows any cryptojacking campaigns known to date by orders of magnitude in installation size, and we find groups of actors compromising a total of 1.4M vulnerable routers, approximately 70% of all deployed MikroTik routers, with various degrees of sophistication. As the injection of miners into network traffic affects any user visiting any website, we find this attack vector to be highly profitable, estimated to exceed $1,200,000 per month in revenue for the top 10 actors. Curiously, we find that innovation and the first-mover advantage do not pay off in terms of revenue made. The highest-grossing actors are not the ones creating new monetization options, deploying sophisticated in-

frastructure, or creating the largest deployment, but those finding the most productive niche where they can operate relatively undisturbed. In April 2019, Interpol began an investigation into the cryptojacking campaigns using MikroTik routers to investigate the perpetrators, clean up the infected routers, and take the supporting infrastructure out of service [175]. To assist with this effort, the research team has shared the results and additional outcomes with the involved law enforcement agencies.

4

INVESTIGATING THE DUTCH PHISHING LANDSCAPE

Off-the-shelf, easy-to-deploy phishing kits are believed to lower the threshold for criminal entrepreneurs going phishing. That is, the practice of harvesting user credentials by tricking victims into disclosing these on fraudulent websites. But, how do these kits impact the phishing landscape? And, how often are they used? We leverage the use of TLS certificates by phishers to uncover possible Dutch phishing domains aimed at the financial sector between September 2020 and January 2021. We collect 70 different Dutch phishing kits in the underground economy and identify 10 distinct kit families. We create unique fingerprints of these kits to measure their prevalence in the wild. With this novel method, we identify 1,363 Dutch phishing domains that deploy these phishing kits and capture their end-to-end life cycle – from domain registration and kit deployment to take-down. We find the median uptime of phishing domains to be just 24 hours, indicating that phishers do act fast. Our analysis of the deployed phishing kits reveals that only a small number of different kits are in use. We discover that phishers increase their luring capabilities by using decoy pages to trick victims into disclosing their credentials. In this chapter, we paint a comprehensive picture of the tactics, techniques, and procedures (TTP) prevalent in the Dutch phishing landscape and present public policy takeaways for anti-phishing initiatives.

This chapter has been published as: **Bijmans, H.L.J.**, Booij, T.M., Schwedersky, A., Nedgabat, A. & van Wegberg, R.S. (2021). "Catching Phishers By Their Bait: Investigating the Dutch Phishing Landscape through Phishing Kit Detection". In *Proceedings of the 30th USENIX Security Symposium (USENIX Security '21).*

4.1. INTRODUCTION

Phishing is a pervasive type of social engineering that harvests user credentials by tricking targets into disclosing personal or financial information – e.g., credit card details – on a fraudulent website. Deploying a phishing website has become trivial with so-called *'phishing kits'*, which can be bought, leased, or even downloaded for free in the underground economy – like dark net markets [161], social media platforms, or secure messaging services like Telegram [147]. A phishing kit contains full-fledged phishing websites [49], mimicking popular banks or financial service providers. Phished credentials are exfiltrated either through e-mail [240] or collected within an administrator panel. As phishing attacks are often tailored to a specific audience and country [221], understanding the impact of phishing kits on the entire landscape should be investigated per linguistic or geographical area to create coherent insights on phishing tactics, techniques, and procedures (TTP). This specific focus aligns with earlier work finding that deployed phishing kits often victimize a particular audience and target banks in a single country [98].

Given our information position in the Dutch cybercrime ecosystem, enabling us to capture the supply of phishing kits, we take phishing targeted at the Dutch financial sector as the focus of our research. The Dutch retail banking sector is very concentrated, as just three large retail banks and a few smaller ones make up the entire market [12]. More importantly, they all primarily service customers through online banking, which is therefore widespread and popular in The Netherlands [30].

While executing a phishing attack has become quite simple, responding swiftly and adequately to this phenomenon is far from trivial. By the time phishing domains are reported to law enforcement agencies (LEA), many of them are already offline. They can be either taken down by the phishers themselves or by hosting providers, often initiated by notice-and-takedown requests by banks whose clients get phished. All of this makes phishing campaign attribution rather difficult, as the window wherein evidence can be collected closes fast. To overcome this challenge, it is essential to proactively detect phishing domains and get a minute-to-minute overview of the phishing landscape. Measuring the scale and operations is crucial for defining robust countermeasures and deploying them before these attacks can cause any harm. Additionally, the recent adoption of SMS and WhatsApp as a means of phishing message delivery [176] has sped up the execution of these attacks even more. Therefore, decreasing the time between the start of the attack and detection – before the arrival of the first victim – is crucial. In this chapter, we present a novel, multi-stage method to detect phishing domains at scale in real-time, capture their attributes, and identify the presence of phishing kits.

We leverage the fact that many phishing domains are secured by TLS connections [8] and that newly issued X.509 certificates can be monitored in real-time by observing Certificate Transparency Logs [89]. By continuously monitoring these logs for 'phishy' domains and subsequently crawling them, we create a dataset of potential malignant domains. By fingerprinting parts of the source code and structure of gathered phishing kits, we measure their prevalence in the wild by detecting these fingerprints on live phishing domains. We group related kits into families, analyze their deployments, and gain more insights into the TTP used by these phishers.

Our analyses aim to create an overview of the impact of off-the-shelf kits on the

Dutch phishing landscape and to identify commonly used TTP. In this chapter, we make the following contributions:

- We present the first empirical, longitudinal measurement study of the end-to-end life cycle of Dutch phishing campaigns.
- We collect 70 different Dutch phishing kits, identify 10 different families, and create unique fingerprints in order to examine the prevalence of these kits in the wild.
- We leverage the use of TLS certificates by phishers and Certificate Transparency Logs to find 1,363 confirmed Dutch phishing domains deploying these kits between September 2020 and January 2021.
- We compile a comprehensive overview of the Dutch phishing landscape, including commonly used (decoy) tactics, phishing kit characteristics, and preferred hosting providers.

The remainder of this chapter is structured as follows: We analyze the anatomy of a phishing campaign in § 4.2, explain our methodology in § 4.3, and present our results in the subsequent sections. In § 4.4, we discuss the results of our analysis on gathered phishing kits. In § 4.5, we examine the domains used by phishers and show how phishing kits are deployed in § 4.6. We benchmark and validate our methodology with external data in § 4.7 and depict the end-to-end life cycle of phishing campaigns with an example in § 4.8. An overview of related work on phishing measurements and phishing kit analysis is given in § 4.9. Finally, we critically discuss our results and methods in § 4.10, share our public policy takeaways, and conclude our work in § 4.11.

4.2. ANATOMY OF A PHISHING CAMPAIGN

A successful phishing expedition results from many crucial steps a phisher needs to take successively. In this section, we examine common techniques to lure in victims and make them disclose their credentials. Next, we depict the complete end-to-end life cycle of a typical phishing campaign. We end this section with the scope of our work before we elaborate on our measurement methodology.

4.2.1. LURING IN VICTIMS

The chances of successfully executing a phishing attack are highly dependent on the credibility of the phishing message – the *bait*. Therefore, phishers use a wide range of techniques and narratives to craft sophisticated phishing messages to trick victims into disclosing their credentials without thinking twice. We can analyze such techniques by utilizing the work of Robert Cialdini, the author of *The Psychology of Persuasion*, who identified several principles that explain how *'mental shortcuts'* can be exploited for the persuasion of others [36]. Recent work by Van der Heijden & Allodi [247] employed Cialdini's principles on phishing e-mails and have shown that *scarcity* – time is limited, so the victim should act quickly – and *consistency* – the victim is already a customer of this bank, so communication is expected – are the most popular persuasion techniques among phishers.

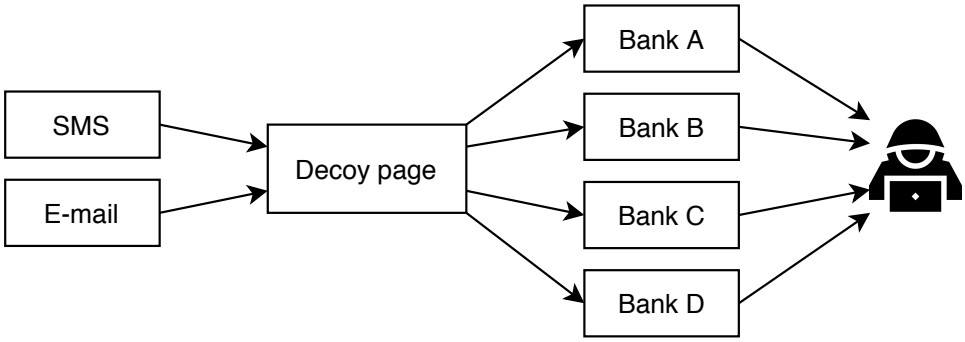


Figure 4.1: Luring technique with a decoy landing page and various fake banking login pages – a so-called *multipanel*.

Although the contents of e-mails or text messages are unknown when analyzing phishing websites, we were able to identify these two principles on pages included in the various phishing kits we examine in § 4.4, as persuasion techniques are exemplified there. Like a request to pay additional shipping costs for postal packages (*scarcity*), an identification request for DigiD – the Dutch online identity to interact with governmental organizations (*consistency*) – or a request to return debit cards to the bank for safe destruction and renewal (both *scarcity* and *consistency*). We noticed that, besides the traditional approach of demanding a victim to log in to their online banking account directly, attackers also deployed more subtle, multi-staged approaches. The first two examples are part of such an approach that phishers follow to improve the credibility of their attack. In such a staged approach, victims are directed towards a decoy page like one of the aforementioned examples first, as shown schematically in Figure 4.1. There are no user credentials harvested on this page, but the victim is directed to a page on which various banks can be chosen to initiate further steps eventually. As the victim is already on a 'trusted' website, it will likely be less observant. Any irregularities are unlikely to be spotted, making disclosing credentials to one of the fake bank login pages deployed by phishers the final step of the fall trap. Phishing kits employing these techniques and containing templates for multiple banks are called *multipanels*, which we will examine in more detail in § 4.4.

4.2.2. END-TO-END LIFE CYCLE OF A PHISHING CAMPAIGN

Whether or not advanced luring techniques are used, the steps to set up a phishing campaign are nearly identical. A typical phishing attack consists of five steps, which we illustrate in Figure 4.2. First, a phisher has to obtain a phishing kit that contains a website created to trick victims into disclosing their credentials. Although phishers could make this website themselves, it is much easier to deploy an off-the-shelf phishing kit that contains all the necessary resources. These phishing kits can be obtained through various sources, such as dark net markets [248] and online forums, but they have become available on public chat applications like Telegram [147] as well. Second, the phisher needs a domain where the phishing website is located. This can either be done by hijacking an

insecure and unrelated website – no costs, more effort – or by simply registering a new domain name – small costs, less effort. Third, when a new domain is registered and a phishing kit is obtained, the phisher needs a Web hosting provider to store the phishing kit files. Consequently, phishers often rent a Virtual Private Server (VPS), which allows them to install a Web server capable of hosting their website. Fourth, to make the phishing website look even more legitimate, the attacker acquires an X.509 (TLS) certificate to create a secure connection between the victim and the website over HTTPS. According to the Anti-Phishing Working Group, 78% of all phishing in 2020 is served over HTTPS [8]. This practice plays into the expectation of Internet users to observe a (green) padlock icon in the browser's address bar when visiting their bank's website – to indicate a secure connection. As Google Chrome started marking Web pages served over HTTP as *'not secure'* in September 2018 [212], potential victims could hesitate to fill in their credentials when the website is not served over a secure connection. Obtaining these TLS certificates is easy and often free through certificate authorities like *Let's Encrypt* [141]. With the website in place, the phisher delivers the *bait* to potential victims by e-mail, text message, or through other means and waits for victims to fill in their credentials.

As we will show in § 4.4, these steps are often explained in great detail by the supplier of phishing kits, allowing their 'customers' to easily set up a phishing website themselves. We, on the other hand, examined these steps in the life cycle of a phishing campaign and identified the fourth step, obtaining the TLS certificate, as a valuable data source for detecting potential phishing domains. More importantly, this is also the only real-time and public data source available to us. For the remainder of this chapter, we follow the steps in this life cycle to present and structure our findings. As the work on examinations and observations of phishing websites in the wild is limited [167, 183] and insights into the complete life cycle of a phishing campaign, combined with thorough phishing kit analysis, are absent, we designed and implemented a measurement system to monitor and analyze the Dutch phishing landscape. The focus on this one consumer market is logical, as Han et al. [98] stated that phishing victims often originate from the same country, which underlines the necessity for country-specific phishing research. Likewise, earlier work on this topic highlighted the fast disappearance of phishing domains [183], making attribution rather difficult. Therefore, it is essential to create a system that could assist law enforcement in quickly responding to these attacks.

4.3. MEASUREMENT METHODOLOGY

To study the Dutch phishing landscape, we follow the life cycle of a phishing campaign, as explained in the previous section. Our measurement approach consists of the following three steps: 1) collect phishing kits on Telegram employing snowball sampling, 2) identify possible phishing domains based on issued TLS certificates, and 3) crawl the

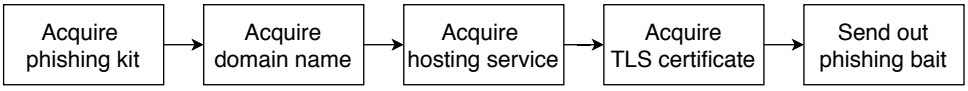


Figure 4.2: End-to-end life cycle of a phishing campaign.

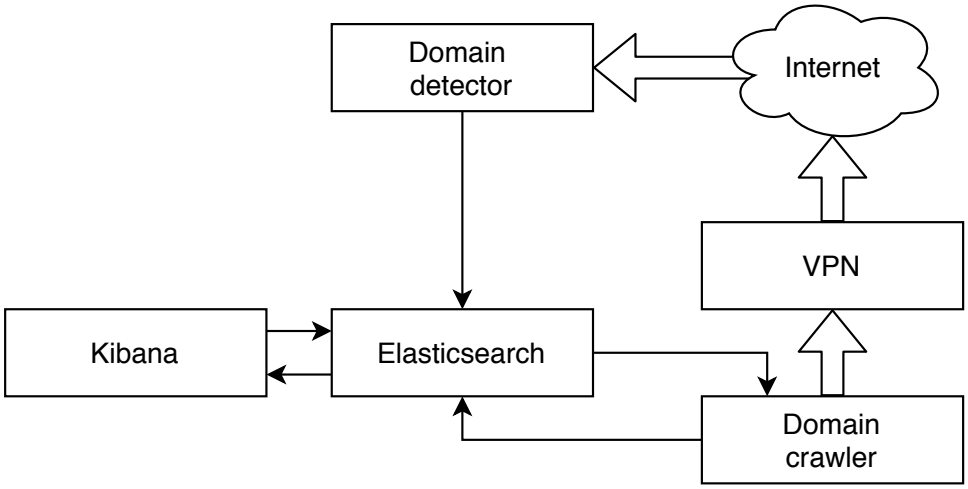


Figure 4.3: Architecture of our measurement system

corresponding Web pages to identify the used phishing kit and capture the end-to-end life cycle of the attack. The methodology used to analyze each of these steps is explained in the following subsections. We store the data produced by all our measurement steps in an Elasticsearch instance, together with Kibana for easy data visualization and monitoring. The complete measurement system is deployed in Docker containers on a cloud server as presented in Figure 4.3.

4.3.1. PHISHING KIT ACQUISITION

We use two approaches to gather phishing kits that target Dutch banking clients. First, we collect phishing kits on public Telegram channels employing a so-called *'snowball sampling'* approach. In addition, we automatically download kits from open directories on crawled phishing domains. We explain both approaches in the following paragraphs.

Telegram is an instant messenger application that allows for secure communications on multiple platforms. The chat application offers a wide variety of channel types, ranging from public broadcast channels to secret chats, with more security features. Encryption is applied to all messages, making it difficult to eavesdrop on communications [234]. The ease of use and the high sense of security on Telegram make it popular among criminals [147] and much easier to use compared to dark net markets or underground forums. Criminals offer illegal drugs, weapons, and phishing kits on public Telegram channels, whereas direct messages on the platform allow them to negotiate prices and make deals with potential customers in private. An example of an advertisement can be found in Figure 4.4, which shows a vendor offering a fake *ING Betaalverzoek* (payment request) decoy page which includes templates for multiple Dutch banks, a so-called *multipanel* as we have explained in § 4.2.1.

To gather phishing kits from Telegram, we manually inspected fraud-related Telegram channels, searched for shared phishing kits, and discovered related channels by

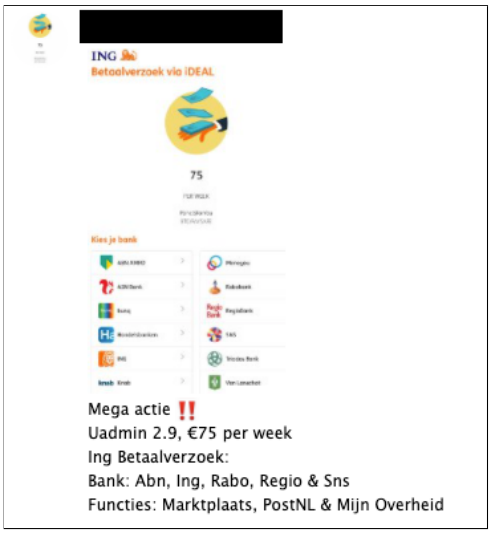


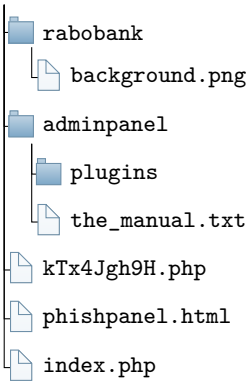
Figure 4.4: Example of a phishing kit offered on Telegram. This vendor offers a phishing-page-for-hire for €75 per week with templates for multiple Dutch banks included – a *multipanel*.

following shared links in the chat. This snowball approach is a common sampling technique that allows reaching saturation in data collection when the total population is hidden or hard to reach [11]. Our data collection saturated after we did not find any new links to our sample of public fraud-related Telegram channels ($n = 50$). Phishing kits shared in these channels are often free – e.g., as a trial version with limited possibilities – can be leased for a customized period of time – as is the case in Figure 4.4 – or bought from the creator or reseller for a fixed price. Kits offered in the latter category are often shared for free (*'leaked'*) afterward to frustrate the seller.

The second approach to obtaining phishing kits is to capture them from suspected phishing domains. As will be explained in § 4.3.3, we crawl each suspected phishing domain, and when such a domain returns an open directory, we follow the same methodology as Cova et al. [49], and search for `.zip` files to find new phishing kits that we then download automatically. Note, we did not search by trying to guess the names of popular phishing kit `.zip` files.

FINGERPRINTING KITS

We manually examined each phishing kit and created *fingerprints* based on the unique properties of these kits. Both the file names, including the full path from the root of the website, as well as strings found on the main page of the website, are used to derive this fingerprint. For example, uncommon file names are considered good candidates for a fingerprint. Next, we inspect the domain's home page to find uncommon strings in the HTML source code. This could be text shown to the victim, but also invisible HTML or JavaScript code included on the page. These fingerprints are used by our crawler to detect the phishing kits deployed on domains in the wild. An example of a phishing kit with the corresponding fingerprint is shown in Figure 4.5.




```
rabobank/background.png
adminpanel/the_manual.txt
kTx4Jgh9H.php
phishpanel.html
```

Kit file structure fingerprint.

```
That's why you have to update your
    banking app and activate it
    again
```

Kit strings fingerprint.

Figure 4.5: Phishing kit file structure and the corresponding fingerprints for both file structure and landing page strings.

4.3.2. DOMAIN DETECTOR

To discover new phishing domains, we leverage the fact that 78% of all phishing in 2020 is served over HTTPS – which requires the use of X.509 certificates – according to the Anti-Phishing Working group [8]. As soon as TLS certificates are issued, they appear in the Transparency Logs Project [89] – a project initiated by Google that collects all issued X.509 certificates. These logs are designed to audit the validity of these certificates, but we use this continuous stream of certificates to find new potential phishing domains. The logs can be monitored continuously using `certstream` – an intelligence feed that shares real-time updates from the Certificate Transparency Log network [27]. We thereby limit ourselves to phishing domains within two of the five categories of the taxonomy created by Oest et al. [182]. Namely, long, deceptive subdomains (type III) and deceptive top-level domains (type IV). Since TLS certificates do not contain paths after the domain name, we can not detect type I and II domains. In addition, as IP addresses – which can be used within TLS certificates – do not contain potentially malignant words, we are unable to detect type V phishing domains.

We advance on the `certstream` Python library [28] to create an application that monitors these logs for potential phishing domains. Just like Lin et al. [146], we were inspired by PhishCatcher [256], an open-source PoC demonstrating the possibilities of finding phishing domains through Certificate Transparency Logs. Our application analyzes all domains present in each certificate and calculates a score based on the features listed in Table 4.1, along with their assigned weighted scores. The first feature extracted is the use of Punycode within the domain name. If that is found, we increase the score by 30 and normalize the domain name for further analysis by converting the Punycode symbols to their regular counterparts. For instance, we convert `xn-pypl-loac.com` to `paypal.com`, which we then use in further steps. We increase the score by 20 for domains hosted on the 10 most abused TLDs according to Spamhaus [197]. Afterward, we split this domain name into words and search for fake TLDs (which could be part of domain names of targeted Dutch banks, so `.com`, `.nl`, `.me`), brand names (of the 13 targeted Dutch banks) and suspicious keywords (a list of 78 words we made ourselves).

Table 4.1: Features used to detect potential phishing websites.

Domain feature	Example & references	Score
Punycode usage	`xn-pypl-loac.com` [79, 148]	30
Suspicious TLDs	`.xyz`, `.icu`, `.top` [8, 197]	20
TLD as subdomain	`x.com.domain.net` [8, 145]	20
Brand name	`brand.domain.net` [8, 145]	40-150
Typosquatted brand	`paypa1.com` [125, 145]	0-110
Suspicious keyword	`login`, `verify` [145, 151]	25-50
Hyphens count	`brand-n--ame.net` [99, 145]	$3x$
Subdomain count	`sub.x.domain.net` [145, 155]	$3x$
Free certificate	`Let's Encrypt` [8, 242]	20
Fake `www`	`wwwbrand.com` [125]	45

We also identify typosquatted variations of the latter two by searching for words with a Levenshtein distance of 1 within the domain name. Additionally, we count the number of hyphens and subdomains and inspect the certificate. The score for domains listed in a free certificate is increased by 20. For domains included in a (paid) certificate with *Extended Validity*, we decrease the score by 100, as we do not expect attackers to pay and complete the verification process. Finally, we disregard domains from Dutch banks and a number of cloud service providers through an allowlist to prevent false positives. When a threshold of 110 is reached, the domain is marked as potentially malicious and added to the Elasticsearch index along with the extracted features and the complete X.509 certificate. This threshold was determined after our testing period in June-August 2020 and was considered a good balance between true and false positives. Do note that we aim to collect as many *potential* phishing domains while keeping the number of false positives manageable. This means that the threshold is not fully optimized to a specific value. Ultimately, our domain crawler – explained in the next section – is responsible for the actual identification of phishing domains.

4.3.3. DOMAIN CRAWLER

To find traces of the gathered and fingerprinted phishing kits, we crawl each of the domains detected by our domain detector. Every hour, the crawler retrieves new possible phishing domains from the Elasticsearch index and starts processing them subsequently. First, it determines if the domain is online, and if so, a Firefox browser controlled by the Selenium WebDriver [217] is launched and visits the domain just like a regular user would. All outgoing Web traffic is routed through a VPN connection to obfuscate our IP address and to easily change our IP address when necessary. While visiting the Web page, the IP address is resolved, HTML sources are stored, and a screenshot is taken. The *favicon* is extracted and hashed using an average hashing function [133], similar to the method suggested by Geng et al. [82]. They showed that more than 83% of phishing websites employ fake *favicons* mimicking the targeted brand or organization. Geng et al. created an algorithm that is able to identify similar *favicons* by comparing the gray values of pixel rows to detect the slightly changed ones. Such hashing is thus per-

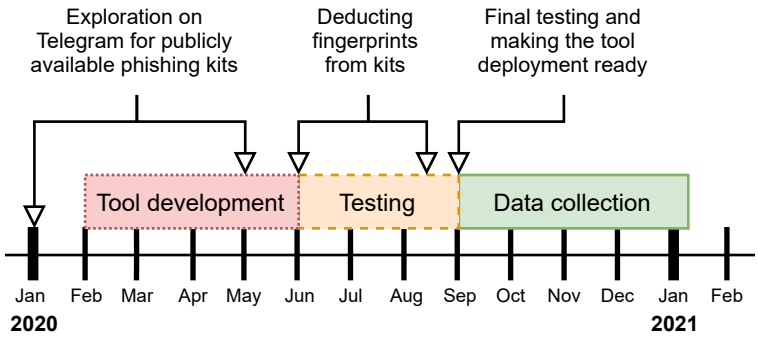


Figure 4.6: Timeline of the creation and testing of our measurement methodology.

ceptual, meaning that small changes in the image result in only minor hash changes. We used their methods to identify domains that do not mimic one of the targeted brands by comparing the *favicon*'s hash to the hashes of Dutch banks *favicons* (12 different brands, 24 icons in total). A domain is omitted from further analysis when the Hamming distance between the found hash and all the hashes of Dutch banks differs by more than 10%. If no *favicon* is present, the domain is analyzed further. Another perceptual hash is generated for the screenshot of the visited page. This hash is used to spot any differences on the page since the last visit. If the hash has not changed since the last visit, we skip further analysis. Otherwise, we continue the analysis by retrieving the `WHOIS` record, which reveals the registrar and the creation date of the domain.

Finally, we start the phishing kit identification phase. In this phase, we adopt a three-layer approach. First, the crawler starts with a search through the list of loaded resources on the Web page. The format of the fingerprints allows us to search for partial file path matches within this list of resources. Given the example in Figure 4.5, resource `https://domain.com/rabobank/background.png` matches fingerprint `rabobank/background.png`. Secondly, we perform a string-based search on the landing page to find matching string fingerprints – e.g., if the page includes the sentence from Figure 4.5, it will be detected. To be able to detect phishing kit resources that are not loaded on the landing page of the website, we perform an extensive search for files and directories on the server using `wFuzz` [159], which tries to `HTTP GET` all resources included in the fingerprint. Given the example in Figure 4.5, resource `adminpanel/the_manual.txt` is not loaded on the landing page of the website but can be detected in this third phase. To harden our detection method against minor changes in phishing kits, we decided to classify a domain as true phishing and identify it as being made with a particular phishing kit when at least 10% of a fingerprint is found in one of these steps. We removed false positives due to this low threshold from our dataset manually in § 4.5. Each domain that is inserted into the Elasticsearch index is monitored on an hourly basis for a maximum of seven days after the initial analysis.

4.3.4. DEPLOYMENT AND TESTING

Figure 4.6 gives an overview of the process of deploying our measurement setup and data collection period. As elaborated on in § 4.3.1, the research started with an exploration of Telegram for phishing kits. These kits were dissected to create fingerprints and then utilized to detect phishing activity on domains. In parallel, we started building our measurement system, and as one can see, we dedicated a significant portion of time to developing, reviewing, and upgrading our deployment.

During our testing phase, newly found phishing kits from open directories are constantly added manually to the crawler application. During this same testing phase, we also identified five new, unknown phishing kits on domains labeled as potentially malicious by our domain detector. However, the crawler could not find any matching fingerprints and labeled these domains as *potentially phishing*. After manual inspection, we determined that these domains were indeed phishing, and we created fingerprints based on the characteristics of these live domains, similar to what we did for the phishing kits in § 4.4.2. We completed this iterative process five times during our testing period and grouped these phishing kits as *unknown*. In September 2020, we stopped testing, made no further changes, and started the data collection.

4.4. PHISHING KIT ANALYSIS

As discussed in § 4.2, phishing campaigns hinge on successful deployment, which can be made easy with a phishing kit. To collect these kits, we manually inspected public Telegram channels following a snowball sampling approach and downloaded `.zip` files from open directories on potential phishing websites. Our initial search in January 2020 resulted in a collection of 36 phishing kits discovered by manually inspecting 50 public Telegram channels. In the following months, we continued to monitor these channels periodically and gathered yet another 10 phishing kits in May 2020. Additionally, as explained in § 4.3.3, we automatically downloaded `.zip` files from open directories on phishing websites, which resulted in a collection of another 24 phishing kits retrieved in the period July – December 2020. In total, we gathered 70 different phishing kits, which we then manually dissected. We analyzed their operating procedures and techniques, came to understand the anatomy of a typical phishing kit, and clustered their features to discern phishing kit families. The results of these analyses are outlined in the following subsections.

4.4.1. ANATOMY OF A PHISHING KIT

A phishing kit consists of many files that together ensure the functionality of the kit when deployed. Among these files, we typically find:

- **Front-end pages** impersonate the original login screens of the targeted banks or can be categorized as decoy landing pages (as explained in § 4.2.1), which direct the victims to fake login screens afterward.
- **Resources** are the files behind the front-end pages, such as JavaScript, CSS, and images. These can either be hosted on the same server – hence included in the phishing kit – or retrieved from the website of the targeted organization.

- **Manuals** are often located in the root folder of the phishing kit and include detailed instructions on how to set up a VPS, acquire a TLS certificate, and install the phishing kit. These files often mention default login credentials and a reference to the creator of the kit.

- **Control panel**, allowing the phisher to access the back-end of the phishing kit, view the phished credentials, or trigger new events for the victim. These panels range from simple text files to extensive dashboards with live visitor manipulations, statistics, and third-party integrations like Jabber – an XMPP instant message service.

- **Anti-detection (cloaking) methods** are present in some kits to prevent detection by law enforcement agencies, independent researchers like us, or anti-phishing services such as Google SafeBrowsing [90]. For example, setting up strict IP blockades on the server-side in an `.htaccess` file as discussed by Oest et al. [182] or by redirecting certain visitors based on their IP address, geolocation, or User-Agent string through PHP scripts. This can also be done client-side by utilizing JavaScript, as discussed by Invernizzi et al. [112].

4.4.2. PHISHING KIT FAMILIES

Precise distinctions between the 70 phishing kits are difficult to make due to the unstructured nature of phishing kit development. During our manual dissection of the gathered kits, we noticed that a large portion of these kits contained copies, older versions, or modifications of one another. Creating unique fingerprints for each of these kits is, therefore, difficult, as such fingerprints could easily match a slightly changed or copied version of another kit. To solve this problem and enable an analysis of their usage, we categorized the gathered phishing kits into 10 families by comparing the files present within each kit. For each of the gathered phishing kits, we calculated the percentage of overlapping files by comparing them pairwise and counting file path matches. Following a similar methodology as in Chapter 2, we used a graph structure to find clusters of similar phishing kits that we can group into families. Displayed in Figure 4.7, we find a directed graph with phishing kits shown as nodes and edges created due to overlapping files. An edge between two phishing kits is created if 75% of the files in a kit overlap. To find families of kits that belong together, we employed a community extraction technique proposed by Blondel et al. [22]. This is a heuristic method based on modularity optimization. The resulting structure describes how the network can be compartmentalized into smaller sub-networks. Utilizing this technique, we determined 10 families of at least two phishing kits per family, in which we grouped 53 phishing kits. The remaining 17 phishing kits have no significant overlap with others and are thus considered not part of any family. An overview of the five largest phishing kit families can be found in Table 4.2.

When taking a closer look at Figure 4.7, we clearly observe one large interconnected network containing four different phishing kit families - the *uAdmin*, *tikkie*, *ics*, and *livepanel* families. From this large community, we can confirm the hypothesis that phishing kits 'learn' – or steal – a lot from each other. The *uAdmin* and *tikkie* families have a lot of overlapping files, but are nevertheless separated into two families. By

Table 4.2: Analysis on the five major phishing kit families.

Family	# kits	Technology	Type	Decoys
uAdmin	17	PHP, SQLite3	multipanel	✓
tikkie	9	PHP, SQLite3	multipanel	✓
bonken	5	ASP.NET	multipanel	✗
ics	4	PHP, MySQL	multipanel	✗
livepanel	4	PHP	single page	✗

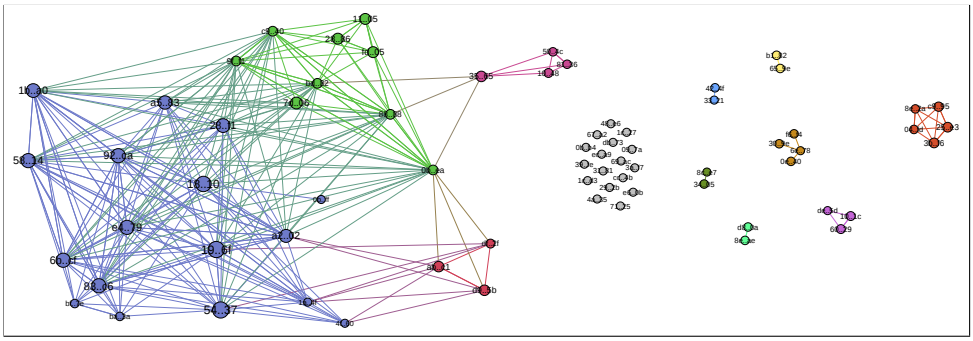


Figure 4.7: Graph of phishing kit families with the first and last two characters of the MD5 hash of the phishing kit name ($n = 70$).

examining the codebase of both these families more closely, we can see that, while they both build upon the same framework – which will be explained in the following paragraph – they have slightly different possibilities. Following this same logic, we took a closer look at the *ics* family. These kits are connected to the larger network through only one kit. The framework used in that phishing kit connects the *ics* family to the network and is again built upon the same codebase as the rest of the cluster. However, it is interesting that the other three kits in the *ics* family are not built upon this framework but do have the same target as the connecting phishing kit. This indicates that this family has 'evolved' into using this framework to perform their phishing activities, adapting to newer technologies. The other, smaller families, positioned to the right in Figure 4.7, clearly employ different tactics than the large interconnected network. For example, the five phishing kits in the *bonken* family are all built upon the ASP.NET Core platform and have nothing in common with the other clusters. As the two largest families and 26 phishing kits in our dataset are built upon the same framework, we highlight its characteristics in the following paragraph.

UADMIN FRAMEWORK

Universal Admin – better known as the uAdmin control panel – is a framework written in PHP and uses a SQLite3 database for information storage. As PHP can be found on almost every Web server and has built-in support for SQLite, this panel can be deployed very easily. It allows for many different templates for most Dutch banks, as well as various decoy pages (as explained in § 4.2.1). A unique feature is that the administrator

panel can be hosted separately from the phishing page. This makes it easy to set up multiple phishing domains and proxy all their connections to a control panel hosted elsewhere. Part of the uAdmin framework is the *O-token* plugin, enabling real-time interaction with the victim. It includes a detailed log of all entered information, as well as buttons to prompt the victim for more input and the possibility to integrate Jabber notifications. This real-time interaction with the victim also allows the attacker to act as a man-in-the-middle to defeat two-factor authentication defenses. uAdmin employs a number of anti-detection methods. There is an `antibot.php` script, which blocks a list of IP addresses, hostnames, and User-Agents. Additionally, when a victim visits one of the pages, a unique folder is created on the Web server, all necessary resources are copied into it, and the victim is redirected to that folder after a timeout of 1 second, as shown in Listing 4.1.


```
$random = rand(0, 10000000);
$md5    = md5("$random");
$base   = base64_encode($md5);
$dst    = 'a1b2c3/'.md5("$base");
...
$src = "def";
dublicate($src, $dst);
...
<script type="text/javascript">
 setTimeout(function(){
   window.top.location.href='<?php echo $dst."?".$_SERVER["QUERY_STRING
       "]; ?>'; },1000)
</script>
```

Listing 4.1: Anti-detection techniques employed (file copy and a JavaScript redirect) by the uAdmin phishing kit family.

This code snippet is very similar to the code mentioned by Han et al. [98] and Oest et al. [183] and tries to prevent detection by anti-phishing services like PhishTank [194] or Google Safe Browsing [90]. Han et al. [98] discovered that these anti-phishing bodies crawl submitted domains themselves and place the landing URL on their block lists. In this case, this is a random path on the Web server, thereby preventing detection. In February 2021, the Ukrainian attorney general's office reported that they arrested the developer of the uAdmin phishing kit after reports that it was used in more than half of all phishing attacks in Australia in 2019 [137]. The Australian Federal Police stated that *"Pretty much every Australian received a half dozen of these phishing attempts."* [137]. Financial institutions in 11 countries, including the United States, Italy, and the Netherlands, were suffering from phishing attacks through uAdmin.

4.5. PHISHING DOMAIN ANALYSIS

Using the method of analyzing the stream of issued TLS certificates in real-time as described in § 4.3.2, our domain detector labeled 7,936 domains as potentially malicious, which meant that these domains reached the threshold value and were further analyzed by our crawler. The domain crawler could match fingerprints of known phishing kits on 1,504 of these domains, which we all manually checked for false positives. We removed 61 domains on which our crawler discovered fingerprints, but no actual phishing took

Table 4.3: Summary of our phishing domains data collection.

Data collection start	September 6, 2020
Data collection end	January 6, 2021
Amount of visits made by crawler	499,497
Amount of potential phishing domains found	7,936
Amount of identified phishing FQDN	1,363
Amount unique phishing RDN	1,112
Average amount of FQDN online every day	31
Median time online (h)	24

Table 4.4: High-level classification of detected domains with examples from our study.

Type III Long, deceptive subdomain	ics-beveiligingsprocedure.zap3.zap-webspace.com mijn.ing.nl.u1234567.cp.regruhosting.ru	66	4.8%
Type IV Deceptive top-level domain	betalingsverzoek-online.link ing-verificatiepagina.eu	1,297	95.2%

place. On most of these domains, this was the result of a fingerprint not being specific enough, and in some cases, the domain responded successfully to all `HTTP GET` requests and thus matched all fingerprints. Finally, as we are investigating the complete end-to-end life cycle of phishing campaigns, we only included domains able to complete a life cycle. This meant that we excluded domains that were discovered in the final week of the data collection period and, therefore, omitted another 80 domains from our dataset. Our final dataset contained 1,363 verified phishing fully qualified domain names (FQDN), which have been online for at least one hour. These were hosted on 1,112 different registered domain names (RDN), as some domains hosted multiple phishing pages on different subdomains. A summary of our dataset is listed in Table 4.3.

4.5.1. DOMAIN NAME CHARACTERISTICS

Setting up a new phishing domain requires a balance between the right amount of persuasion of the victim and stealth to prevent early detection by anti-phishing organizations. As explained in § 4.3.2, common practices to hide malicious activity are to obfuscate (parts of) the URL by using deceptive subdomains, Punycode, or typosquatting. The use of deceptive subdomains is categorized as type III by Oest et al. [182], and we could discover only 66 of such domains in our dataset. As listed in Table 4.4, we identified many more type IV domains (1,297) in our dataset. 16 of the 66 type III phishing FQDNs increased their credibility by including the full FQDN of the target brand as subdomains. This practice can result from either one of the following techniques: this RDN could be hijacked or specially chosen to increase stealth. In the case of hijacked domains, attackers have taken control over the domain and made (multiple) subdomains for their phishing page, a practice discussed extensively by Han et al. [98]. For the other technique, adding the domain of the targeted bank as a subdomain is done to increase the credibility of the URL, which works especially well on mobile devices, on which the com-

plete URL is not always shown in the GUI. Distinctions between these two categories are difficult to make, as we can not determine whether a domain is hijacked or intentionally chosen by the attacker to avoid early detection.

Although mentioned in related and previous work on this phenomenon [138, 148], we did not find any successful usage of Punycode obfuscated domains in our dataset. The use of Punycode did increase the malicious score of a domain in our domain detector, and we identified 21 of such domains, but our crawler did not find matching fingerprints on any of them. This could indicate that the use of Punycode is less popular among attackers focused on Dutch consumers, as we did find references to other banks outside our scope. On the other hand, typosquatting – also known as URL-hijacking – is found 36 times in our dataset. The practice of replacing the character `i` with `l` in domains mimicking the *ING Bank* and *ICS Cards* is popular, as we found respectively 16 and 20 of such domains.

However, most phishing FQDNs in our dataset simply obfuscate their malicious intents by not mentioning the name of the target organization. As shown in Table 4.5, more than half of the domains in our dataset (770) did not include any references to Dutch banks but were detected because of other words mentioned, which we included in our methodology as *suspicious keywords*. These words refer to either banking-related matters – e.g., *payment*, *verification* or *debit card* – or to completely different matters, often related to the decoys mentioned in § 4.2.1.

TARGETED BANKS

An analysis of the FQDNs that do refer to one of the targeted banks results in insights into their popularity. Note, however, that indicators in the domain name are not always directly linked to the actual Web page on that domain – e.g., a domain including a reference to bank A contains the login screen of bank B. Our domain detector searched for references to the ten largest Dutch retail banks and two daughter brands of ABN AMRO – Tikkie and ICS Cards – within all domains and was able to identify 593 FQDNs referring to one of them. As shown in Table 4.5, we found 194 domains referring to Rabobank, which makes it the prime target for attackers. In contrast, only ten domains contained references to Regiobank, making this bank seem a less attractive target.

4.5.2. DOMAIN REGISTRATIONS

When choosing a top-level domain (TLD) as an attacker, it is important to keep in mind that different registries have different policies when it comes to monitoring and cleaning their TLD. Some registries allow registrars – the companies selling the domains used for phishing to the attackers – to sell large quantities of domain names to attackers and are hereby knowingly contributing to online abuse. As The Spamhaus project states: *"Some registrars and resellers knowingly sell high volumes of domains to these actors for profit, and many registries do not do enough to stop or limit this endless supply of domains."* [197]. The Spamhaus Project monitors domains in SPAM messages and calculates the percentage of bad domains within each TLD zone. We compare their data with our results to find out whether phishers focused on Dutch consumers favor these TLDs over the more regularly used TLDs in the Netherlands. The results of our analysis – listed in the first columns of Table 4.6 – show that `.info` is the most commonly used TLD in

Table 4.5: Popularity of targeted banks and suspicious keywords.

Brand name	#	**Suspicious word**	*(translation)*	#
Rabobank	194	Betaal	*(pay)*	300
ING Bank	135	Verzoek	*(request)*	271
ICS Cards	48	Mijn	*(my)*	217
Tikkie	40	Veilig	*(secure)*	159
Knab	37	Betaling	*(payment)*	153
ABN AMRO	25	Omgeving	*(environment)*	119
Bunq	16	Platform	*(platform)*	116
SNS Bank	13	Verificatie	*(verification)*	87
Regiobank	10	iDeal	*(iDeal)*	73
Triodos	8	DigiD	*(DigiD)*	70
Not mentioned	770	Not mentioned		125

Table 4.6: Overview of the top 10 top-level domains (TLDs), domain registrars and hosting providers used by attackers.

TLD ($n = 1,112$)	#	%	**Registrar** ($n = 933$)	#	%	**Hosting provider** ($n = 836$)	#	%
`.info`	202	18.2	Namecheap	678	72.6	Namecheap	280	33.5
`.xyz`	159	14.3	REG.RU LLC	46	4.9	Combahton	84	10.0
`.com`	149	13.4	Porkbun LLC	30	3.2	HS	58	6.9
`.nl`	102	9.2	NameSilo, LLC	21	2.3	Alibaba (US)	56	6.7
`.me`	74	6.7	Eranet International Ltd.	17	2.4	Cherryservers	29	3.5
`.icu`	71	6.4	GoDaddy.com, LLC	12	1.3	First Colo	26	3.1
`.online`	57	5.1	Tucows Domains Inc.	12	1.3	NCONNECT-AS	24	2.9
`.site`	50	4.5	AXC	10	1.1	Serverion	23	2.8
`.net`	28	2.5	Openprovider	8	0.9	YURTEH-AS	14	1.8
`.top`	23	2.1	Registrar.eu	6	0.6	OVH	14	1.7

our dataset, followed by `.xyz`. These phishers tend to choose one of the many 'bad' TLDs, but they also stick to the more commonly used TLDs in the Netherlands, such as `.com` and `.nl`.

DOMAIN REGISTRARS

Using the retrieved `WHOIS` records, we were able to identify the registrar of 933 of the 1,112 RDNs in our dataset. We thus have no information about the registrar for 179 RDNs. Inspecting the `WHOIS` records of the 933 domains revealed that *Namecheap* is by far the most popular registrar used by phishers, as 72.6% of all phishing domains were registered through that registrar. Other large registrars, such as *Porkbun* and *GoDaddy.com*, are significantly less popular than one would expect. Another interesting observation is the use of *REG.RU*, a Russian domain registrar, which is found 46 times in our dataset. An overview of the 10 most popular domain registrars can be found in the middle of Table 4.6.

CERTIFICATES AUTHORITIES

The fourth step in the end-to-end life cycle of a phishing campaign is acquiring a TLS certificate. As explained in § 4.3.2, we leverage this step to detect phishing domains in our analysis. *Let's Encrypt* is the main supplier of TLS certificates in our dataset, as 67% of all FQDNs use such free certificates. Additionally, we found 146 domains with a certificate issued by *cPanel*, software often used to manage the domain. Most certificates (99%) are Domain Validated (DV). We gathered 33 Organisation Validated (OV) TLS certificates issued through *CloudFlare*'s free certificate service. These certificates require additional validation steps, which are highly unlikely for a phisher to fulfill, as this would disclose their identity.

4.6. PHISHING WEBSITE DEPLOYMENTS

In the four-month data collection period, our domain crawler made a total of 499,497 visits to 7,936 unique FQDNs. As explained in § 4.3.3, the crawler visits every domain labeled as potential phishing by our domain detector and monitors it for a period of a maximum of seven days after the initial discovery. Properties such as the used phishing kit, the IP address, and `WHOIS` record are gathered during this process.

Besides choosing a suitable TLD and a domain name to be used for their phishing attack, phishers also need a place to host their website. By resolving the IP addresses of identified phishing domains and mapping them to their corresponding Autonomous System (AS), we determined the hosting provider of each domain. An overview of the top ten providers can be found in Table 4.6. Similar to the domain registrations mentioned in § 4.5, *Namecheap* is the most popular hosting provider among attackers in our dataset. The overall popularity of *Namecheap* has various reasons. First of all, it is – like it says – cheap, and as attackers want to maximize their profits, it makes sense to rent an inexpensive VPS instead of an expensive one. Second, *Namecheap* accepts payments in Bitcoin [174], which offers more operations security to attackers due to the relative anonymity of Bitcoin transactions. Finally, it is mentioned explicitly by various phishing kit creators in their manuals.

Surprisingly, none of the hosting providers in this list can be regarded as *bulletproof* – i.e., very reluctant to LEA requests – except HS, short for Host Sailor. This provider does have a disreputable background [135] but is used by only 58 domains in our dataset. Another interesting entry in Table 4.6 is *Combahton*, an inexpensive German hosting provider used by services like *zap-webspace.de* and *gamingweb.de*. From the lack of bulletproof hosting providers, we derive that these phishers are not concerned about the extended lifespan of their domain. As long as they act quickly, they are long gone before their domain is taken offline by third parties. However, the choice of these services does open avenues up for possible law enforcement interventions, as mainstream hosting providers – such as *Namecheap* – are willing to cooperate with law enforcement.

4.6.1. PHISHING KIT PREVALENCE

As stated in § 4.4.2, we obtained a total of 70 phishing kits, which we dissected and grouped into 10 families of similar kits. During the data collection period, our crawler found matching fingerprints for 7 of the 10 different families. We show the size of the Dutch phishing landscape and the popularity of the different phishing kit families in Figure 4.8, in which the total number of active and online phishing domains is shown per day, categorized per phishing kit family.

Although we expected a wide variety of phishing kits to be used, the opposite turned out to be true. The overwhelming majority of phishing domains our detector found were made using one of many variants of phishing kits within the *uAdmin* family. Almost 89% of all identified phishing websites were made with a kit within this family, shown by the size of the lower blue bars in Figure 4.8. As explained in § 4.4.2, these phishing kits contain many templates for different banks and often include decoy pages, making them attractive to aspiring phishers. The support for many different bank login templates also explains why many of the domains are labeled as *'multiple'* in Figure 4.8. These domains have fingerprint matches of both *uAdmin* and another phishing kit family. It seems that phishing kit creators are integrating as many templates as possible from different kits into the uAdmin framework. The structure of this framework remains often unchanged, as we could locate the control panel on its default location on 775 of the 1,211 FQDNs (64%) that matched a fingerprint of a phishing kit in the *uAdmin* family. Finally, as shown in red in Figure 4.8, the category *unknown* consists of new, unknown phishing kits found on live phishing domains. As explained in § 4.3.4, we manually verified that these domains were indeed phishing and created fingerprints of the used kits according to the characteristics of these live domains.

4.6.2. CAMPAIGN DURATION

Since our crawler monitored each identified phishing domain for a maximum of seven days (168 hours), we were able to closely follow these domains and capture the end-to-end life cycle of a typical phishing campaign. Additionally, as stated before in § 4.5, we manually checked the dataset to prevent any false positives from being included in the data and only included domains with a complete end-to-end life cycle in our analysis, which allowed us to analyze this in the next paragraphs.

First, we plot a histogram of the uptimes of all domains in our dataset in Figure 4.10 with a logarithmic Y-axis. As one can see, the majority of domains have an uptime of 0 to

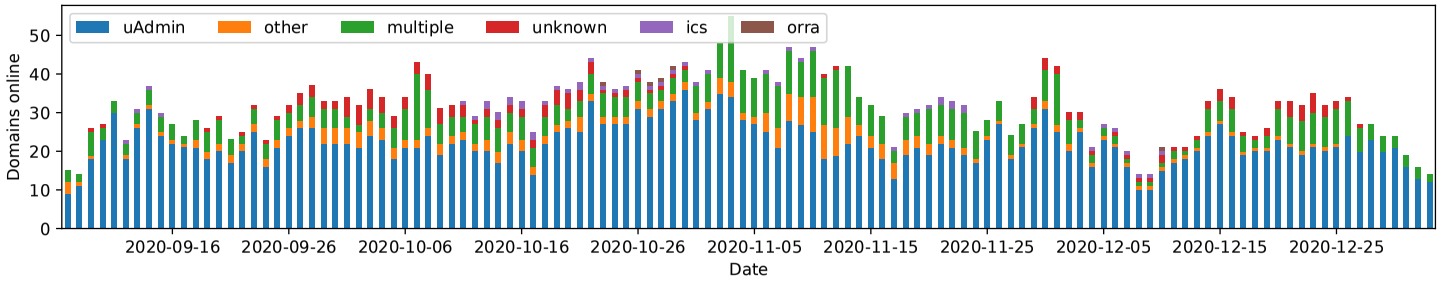


Figure 4.8: Number of domains active per day, grouped per phishing kit family ($n = 1{,}363$).

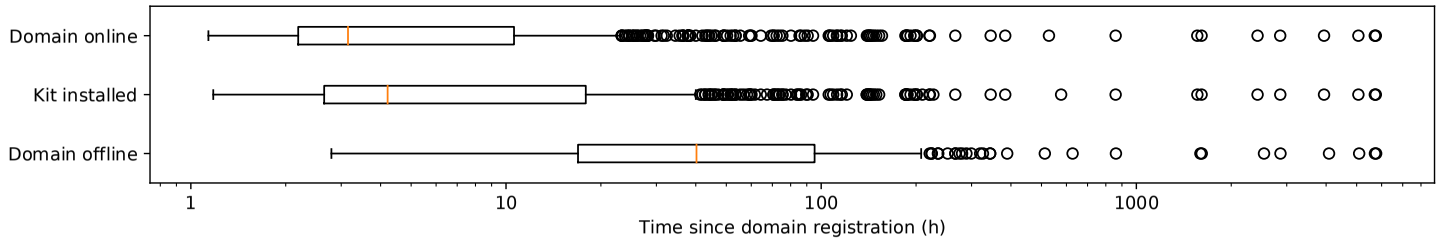


Figure 4.9: Boxplot of timestamps in the end-to-end life cycle of the identified phishing domains ($n = 818$).

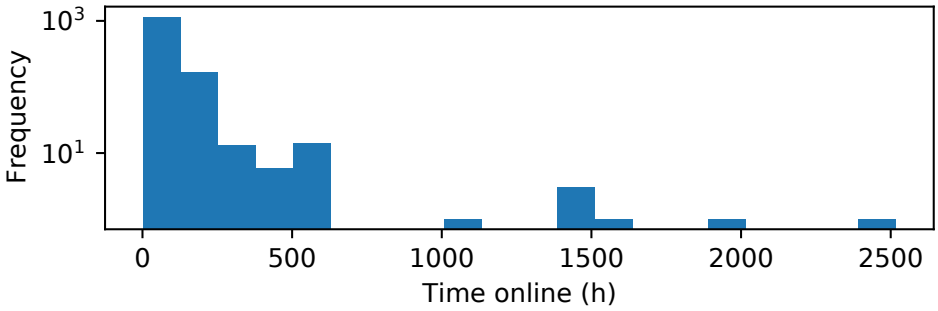


Figure 4.10: Histogram of phishing domain uptimes, including domains with multiple certificates ($n = 1,363$).

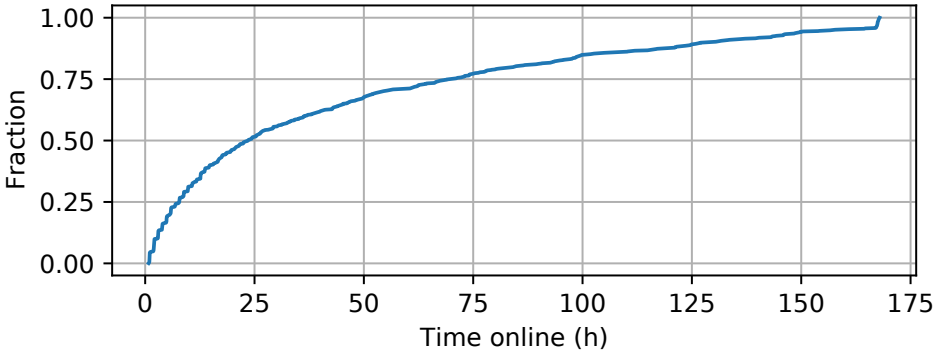


Figure 4.11: CDF of a phishing domain uptime ($n = 1,288$), domains with multiple certificates issued excluded.

200 hours, which coincides with our maximum analysis period of 168 hours. However, there are 75 domains with an uptime of more than 168 hours (7 days). After manually inspecting this unexpected result, we found that these domains requested multiple TLS certificates during their uptime, which caused our domain crawler to restart the crawling cycle as soon as a new certificate was issued. Since these outliers heavily influence the results and prevent us from determining the timestamps of the steps in the life cycle, we exclude them for the remainder of the analysis in this paragraph.

Now, we are able to calculate the uptime of the 1,288 remaining phishing domains in our dataset. On average, a phishing domain in our dataset is online for 45 hours, but we find a median uptime of 24 hours. The uptimes are shown as a cumulative distribution function (CDF) in Figure 4.11. Thus, 50% of all domains have a lifespan of less than a day, whereas just over 30% are online for more than two days. These numbers again stress the fact that speed is key in anti-phishing initiatives.

INSTALLATION OF PHISHING KITS

Although it is hard to determine which actors are behind phishing attacks on Dutch consumers, the timestamps of the first identification of an active phishing kit installation do

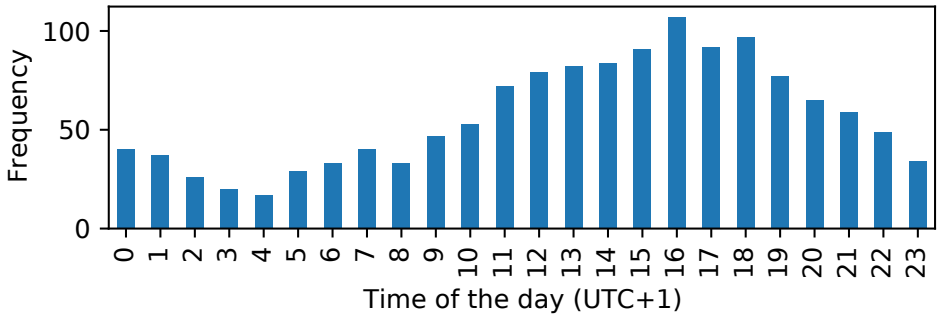


Figure 4.12: Histogram of kit installation hours (n = 1,363).

give some clues into the region of the world these attackers operate from. As shown in Figure 4.12, the phishing kit installation times (in UTC+1) align with the Dutch circadian rhythm. Most phishing kits are installed during the day, while almost none of them are installed in the middle of the night. This finding, and the fact that most manuals of the gathered phishing kits are written in Dutch, extends the conclusions of Han et al. [98], as this would indicate that both victim and attacker originate from the same country.

During installation and testing of the kit, visitors are occasionally redirected to popular benign domains like Google or Bing or to the website of the target organization. During our crawls, we observed 49 different phishing domains doing this before their phishing kit was fully deployed and operational.

END-TO-END LIFE CYCLE STEPS

We can determine timestamps of all steps within a typical phishing campaign – shown in Figure 4.2 and explained in § 4.2 – by combining the retrieved `WHOIS` records and crawling timestamps of all identified phishing domains. Unfortunately, 460 FQDNs in our dataset lack `WHOIS` information due to inconsistent information formatting or server errors beyond our control. Therefore, these domains are excluded from the analysis in this paragraph. Additionally, we focus this analysis on type IV phishing domains only because type III domains include hijacked domains that have not been registered purposefully for phishing.

The end-to-end life cycle analysis on the remaining 818 domains is summarized in Figure 4.9 as a horizontal box plot, with the hours since domain registration on a logarithmic X-axis. As indicated by the red bars inside the boxes, a phishing domain is online – i.e., returns a successful `HTTP 200` response – three hours after registration on average. Oftentimes, it is quickly followed by installing a phishing kit, on average, only one hour later. After a successful installation, the phisher sends out the *bait* to its potential victims and waits for credentials to be filled in. The domain goes offline after 40 hours on average. Most domains complete this full life cycle within a couple of days. Note, however, that there are also outliers. In these cases, the domain was registered many days in advance, waiting to be used by the attacker. In our dataset, only 114 of the 818 domains (14%) were registered more than 24 hours before coming online.

4.6.3. EXTERNAL RESOURCES & EVASION TECHNIQUES

During our analysis of phishing domains in the wild, we noticed that some websites make external connections. As explained in § 4.4.1, phishing websites could either include all impersonated resources – e.g., JavaScript, CSS, and images – on the domain or refer to resources hosted externally. Analyzing the resources loaded by all identified phishing domains tells us that only 104 domains (7.6% of the total dataset) load their resources directly from their benign counterparts. This finding contradicts the assumption underlying the work of Oest et al. [183] and makes their method of analyzing Web server logs for malicious external requests less robust, as only a very small portion of websites in our dataset are pursuing this method. However, it does confirm the findings of Han et al. [98] and Cova et al. [49], who also observed a negligible portion of phishing kits with resources loaded from the target organization. These authors studied attacker behavior on honeypot domains, which is based on the assumption that attackers hijack domains to use for phishing. Although our measurement methodology is not perfectly suited to finding such hijacked domains – as these domains often already have TLS certificates – we did find 18 of them. All of these domains include the full FQDN of the target organization as subdomains and have a slightly longer uptime of 72 hours on average.

EVASION TECHNIQUES

As explained in § 4.4, some phishing kits deploy evasion techniques to prevent detection by anti-phishing services such as APWG [97] and Google SafeBrowsing [90]. These techniques, often referred to as *cloaking*, allow phishers to show a different page to a potential victim than to a crawler [112, 260]. Although our methodology is focused on detecting the use of specific phishing kits in the wild and not identifying cloaking, we did observe such evasion techniques many times. In fact, 946 (69%) of the detected phishing domains returned a blank screen – and no favicon – to our crawler when we visited the domain, meaning that the phishing website detected us and deployed cloaking techniques. However, our phishing kit detection was still possible because these websites returned a successful response for the files included in the fingerprint. The phishing kit responsible for most of these cloaking activities was again the *uAdmin* kit, which combined some server-side and client-side cloaking. On the server side, it checked the IP address with a block list and created a random path for every visitor, as explained in § 4.4.2. On the client side, it deployed a simple JavaScript timeout to evade non-JavaScript crawlers. The combination of both techniques is shown in Listing 4.1.

4.7. EXTERNAL VALIDATION

To benchmark and validate our methodology, we compare our results with data from the APWG eCrime Exchange (eCX) [9]. This repository contains phishing activity from all over the world, including many Dutch phishing domains. A comparison shows that our methodology covers a much broader spectrum of phishing domains, capturing known differentiations in the phishing landscape. In total, only 77 phishing domains detected using our methodology overlap with the APWG database, meaning that 1,286 domains are not listed in their repository. By comparing the date on which a phishing domain was initially detected by our crawler with the data it had been submitted to the eCX, we find that our method was able to identify phishing domains much faster. In 76 out

of the 77 cases (99%), our crawler detected the phishing domains faster than APWG, with a median time difference of 11.3 hours (almost half a day) earlier. Interestingly, the domains that overlap with the eCX repository clearly had more bank names included in their domain name. 61 of the domains (79%) overlapping with eCX contained a reference to a bank, whereas only 44% had this in the complete dataset. This external validation shows that our methodology has the potential to detect phishing websites very swiftly, which could save unsuspecting people from this kind of fraud.

4.8. THROWING OUT THE BAIT

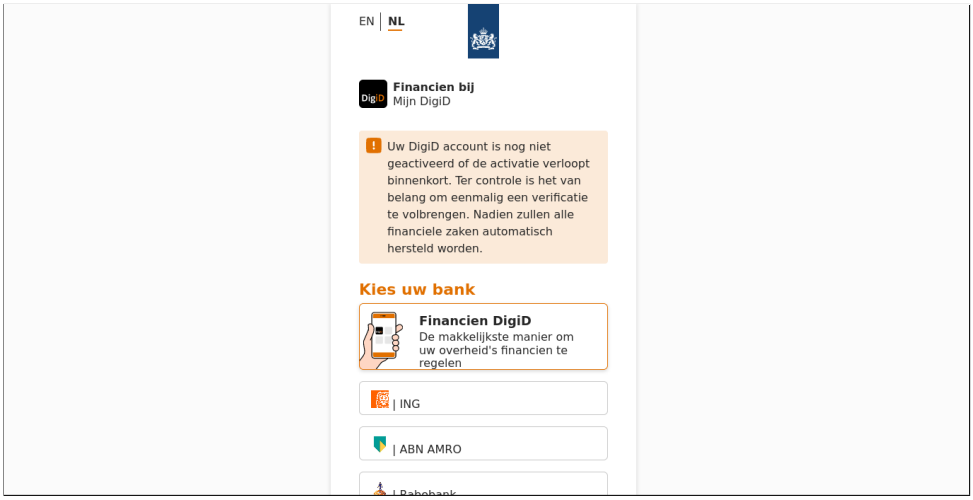
In the previous sections of this chapter, we have unraveled the characteristics of every step in the end-to-end life cycle of a phishing campaign, except for the last step: sending out text messages, e-mails, or social media posts, the so-called *bait*. Although our measurement system does not contain the input data necessary to thoroughly analyze this step of the life cycle, the authors are among the target population of phishers and thus regularly receive the thrown-out bait themselves. During our data collection period, we collected these messages and looked into the ones that contained links to domains in our dataset. This allows us to show the complete timeline of events in a phishing campaign life cycle. We discuss an example in the following section.

VERIFY YOUR IDENTITY

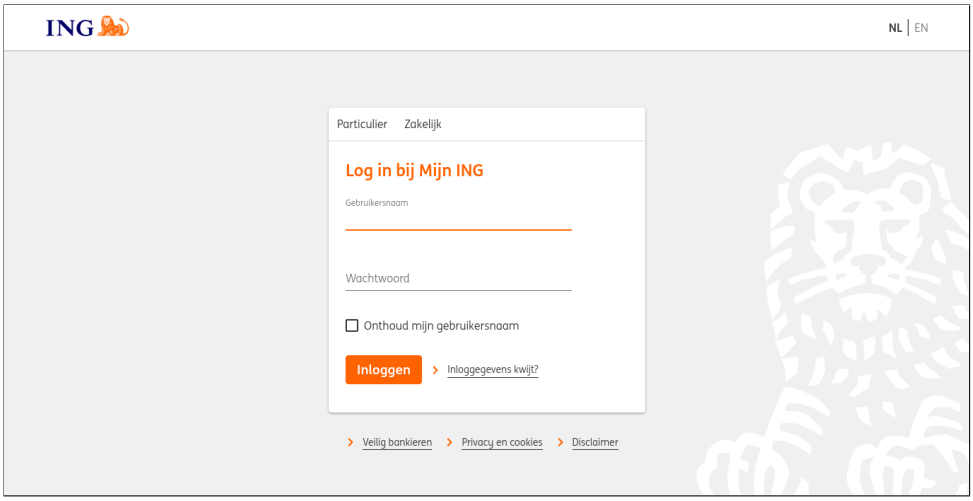
Within the first two weeks of our research, we received a text message seemingly originating from DigiD, the official Dutch digital identity service. The message shown in Figure 4.14 mentioned a detected suspicious login and requested immediate action to prevent the cancellation of the account. This is a prime example of the *scarcity* and *consistency* luring techniques as described in § 4.2.1. The link included in the message directed victims to `https://deblokkeren-digid.xyz`, a type IV domain made with a phishing kit belonging to the *uAdmin* family. This website was registered only six hours before the message was received and was fully operational just three hours later. On the website, potential victims were asked to verify their identity by logging into their online bank account. Multiple options are displayed on the decoy page as shown in Figure 4.13a, allowing the victim to choose their bank. Upon clicking on one of the buttons, the victim is redirected to yet another phishing page, as shown in Figure 4.13b, which mimics the chosen bank's login screen. That page eventually captures the login credentials of the victim. Using the DigiD decoy page is a prime example of the technique depicted in Figure 4.1. Within a day, only 12 hours later, the domain was taken offline.

4.9. RELATED WORK

Earlier work on phishing involves many different angles. Ranging from creating robust domain detection methods [23, 80, 82, 145, 155, 233], phishing kit analysis [49, 98, 182], evasion techniques [112, 260] to research focused on victim behavior [247]. Much effort has been devoted to creating robust detection techniques, but less is known about the life cycle, ecosystem, and actors behind such attacks. Only a limited number of researchers have investigated this part of phishing [167, 183], which we deem essential to fully understand the ecosystem and to be able to create robust countermeasures.



(a) Decoy (landing) page demanding verification through a bank account to prevent account deactivation.



(b) The phishing page is shown after selecting a bank on which user credentials are harvested.

Figure 4.13: The two pages involved in a multi-stage phishing attack using a decoy.

SMS Message – 17-09-2020 21:32 *(translated)*

[My DigiD] There has been a suspicious login in your My DigiD account. Verify this directly to prevent cancellation of your My DigiD account through: `https://deblokkeren-digid.xyz/inloggen`

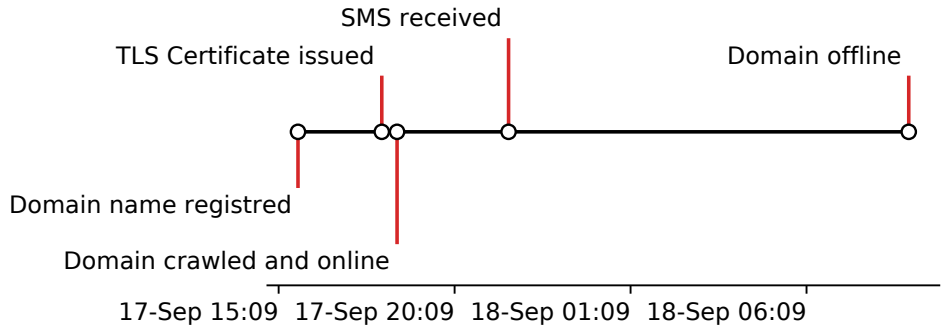


Figure 4.14: Text message demanding DigiD verification and corresponding timeline of the domain `deblokkeren-digid.xyz`.

ANALYSIS ON PHISHING KITS

Early work on phishing kits in 2008 by Cova et al. [49] focused on analyzing *'free'* phishing kits. They noticed that packages containing easy-to-deploy phishing websites often contained backdoors, which exfiltrated the gathered information also to third parties, and that 100% of the investigated kits were written in the PHP language. In their PhishEye study, Han et al. [98] share insights into live phishing websites created by deploying phishing kits on honeypot domains. Using their sandboxed approach, they were able to lure phishers into installing phishing kits on their honeypot servers, the behavior of which was closely monitored. The authors analyzed both phisher and victim actions on the phishing website and showed that phishing kits are only active for less than 10 days since their installation, and that most of the victims share the same country of origin. During their five-month analysis period (Sep 2015 – Jan 2016), they collected 643 unique phishing kits, of which 74% were correctly installed by 471 distinct attackers. Additionally, they discovered that only 10 phishing kits loaded the resources directly from the website of the targeted organization.

MEASUREMENTS ON LIVE PHISHING DOMAINS

In recent work, Oest et al. [182] analyzed `.htaccess` files – commonly used on Apache Web servers – to capture the evasive behavior of phishers. These files allow phishers to protect themselves against anti-phishing or search engine crawlers. Their paper states that *deny IP* and *User-Agent* filters are the most prevailing blocklisting technologies, while the *allow IP* filter type is often used to target specific countries. Additionally, they proposed a new high-level classification scheme for phishing URLs that builds upon the work of Garera et al. [80]. This taxonomy categorizes phishing URLs into five categories

with different hiding and lure strategies. We also used that taxonomy to classify the URLs detected by our measurements in § 4.5.1.

The work closest to ours is from the same authors, who continued their research by investigating the end-to-end life cycle of phishing attacks in 2020. This work relied on the observation that a substantial proportion of phishing pages request Web resources to the websites that the attackers impersonate [183]. A unique collaboration with a large payment provider enabled them to link such Web requests to the phishing websites from which they originated. This gave the authors an in-depth look into phishing campaigns from the moment the attacker installs the phishing page to the moment victims disclose their credentials. They found that the average phishing attack spans 21 hours and that modern Web browsers display a warning for a detected domain after 16 hours. Oest et al. [183] called the gap between the launch of the attack and detection by anti-phishing bodies the *'golden hours'* of phishing, in which the attackers gather 38% of their phished credentials. As our work shared a similar goal – analyzing the end-to-end life cycle of a phishing campaign – we share a number of findings. Namely, the use of extensive use of server-side cloaking, victim-specific paths, and the presence of MITM-proxies in phishing kits. Additionally, our conclusions regarding the duration of an average phishing attack are comparable. However, there are also notable differences. Their work is focused on one single organization and includes both HTTP and HTTPS traffic, whereas our work focuses on the entire Dutch financial sector but is limited to domains served over HTTPS only. Furthermore, they relied on the assumption that phishing domains load resources directly from the target website, whereas we discovered that only a negligible portion of domains in our analysis did so.

4.10. DISCUSSION

This section discusses the inherent limitations of our work and presents public policy takeaways.

LIMITATIONS

Analyzing a phenomenon like phishing always brings its inherent limitations, and so does this study. As with all other work on this topic, our methodology is only able to capture part of the phishing landscape. We identify the following limitations:

We are aware of the fact that by our choice of methodology, we are limited to phishing domains secured by HTTPS connections only. Yet we believe, as 78% of all phishing in 2020 is delivered through HTTPS according to the APWG [8] and the fact that Oest et al. [183] concluded that phishing served over HTTPS was three times more effective, the effects of that concise decision to be limited. Also, note that our approach results in our ability to identify type III and IV phishing domains only and thus misses the three other types. Another limitation of this work is that we are limited to identifying known phishing kits. Phishing domains that do not match any of our predefined fingerprints are simply not marked as phishing. Besides these missing kits, phishers could also change the file names or structure of their phishing kits, which would also render our detection methodology less effective. However, the main advantage of phishing kits is that they are easy to deploy for any criminal who wants to go phishing. Therefore, we do

not expect that phishers who deploy these kits are either capable or willing to make numerous changes each time they deploy a new phishing website. On the other hand, our fingerprinting methodology also has a detection advantage for websites that deploy certain cloaking strategies. As explained in § 4.4.1, some phishing websites ban IP ranges or User-Agents known to be used by anti-phishing services through PHP scripts on the homepage or show a different landing page depending on the country of origin. These methods make detection based on the characteristics of the page – e.g., login forms, bank icons, etc. – rather difficult. However, searching for known files – our fingerprints – on such domains bypasses these evasion methods and results in robust detection of a phishing kit.

We started our crawling infrastructure three months before data collection started, which allowed us to carefully examine the domains missed by our crawler. As explained in § 4.3.4, we created fingerprints based on the source code of live phishing websites missed by our crawling during testing. So, even without obtaining the actual phishing kit, we could create robust fingerprints.

Unfortunately, the largest limitation is in the missing data we do not see. As explained in § 4.5.1, many domain names do not contain references to bank names but only use common words. Before data collection started, we added 78 such words to our suspicious keywords list, but we have definitely missed some. As these domains did not reach the threshold set in our domain detector, they remain undetected. The validation with APWG eCrime Exchange data in § 4.7 showed that such domains are less prevalent in this anti-phishing repository. It is, therefore, important to include such words. We identify the validation with only one data source as a limitation of our work, but leave validation with more datasets for future work.

PUBLIC POLICY TAKEAWAYS

Taking decisive action on phishing is complex. Ironically, the standardized notice-and-takedown (NTD) procedure that banks generally outsource to the security industry has resulted in a game of whack-a-mole, leaving the police chasing these criminals often empty-handed. And, as concluded by Moore & Clayton [167] in 2017, website removal is only part of the answer to phishing but is not fast enough to completely mitigate the problem. If and when phishing campaigns are reported to law enforcement agencies (LEA), phishing domains are often already taken down, making attribution of the actors behind phishing campaigns nearly impossible. Therefore, implementing a system as presented in this chapter would be very beneficial for LEA investigations.

With WhatsApp and text messages being a popular delivery mechanism [176], the interaction with victims has sped up, highlighting the need for early-stage detection even more. This chapter presented a measurement methodology leveraging the increasing use of phishing kits and TLS certificates in the phishing scene to make early-stage detection possible. This would open a window where phishers have their phishing gear ready but have not yet thrown out the bait. Our findings pinpoint clear choke points in using phishing kits in campaigns, which law enforcement agencies, in turn, might exploit for disruption before a takedown occurs. Our measurements of the life cycle of campaigns using phishing kits show a pattern wherein a persistent time gap exists between domain registration, deployment, and sending out the bait. This is a window of oppor-

tunity that can be used to take preventive action when the campaign has not made any victims yet. Leveraging our methodology, kit fingerprints can be used to automate the detection of domains where a kit is ready to be deployed. We show that the use of these kits is widespread in the Dutch phishing landscape and have found that distinct families of kits exist, wherein certain common characteristics are identified – likely because the source code of one kit has evolved into the next. When these characteristics relate to a vulnerability – e.g., the standard admin password is `'password'` and the control panel can be approached via a typical subdomain – this brings novel opportunities for automated exploitation for law enforcement purposes towards attribution rather than disruption. Having a clear picture of the popularity of phishing kits could assist LEA in prioritizing their anti-phishing efforts to dominant kits. Interventions – e.g., exploiting a vulnerability – on these kits would immediately impact a large portion of campaigns. Next, these shared traits can also be used to keep track of the phishing landscape. For instance, *uAdmin* allows for multiple domains contacting the same control panel, making in-depth analysis possible on these domains to find new, related campaigns or actors.

A system like ours could complement the threat intelligence process of many organizations, especially financial institutions, that suffer from these attacks. Additional measurements in the landscape can also be enriched by a repository of phishing kit fingerprints. Similar to repositories for malware fingerprints, the community – from hosting providers, to volunteers and researchers – can contribute their analyses on phishing kits to keep track of this pervasive phishing tactic. In turn, standardization on how to describe phishing kits, their tactics, and detection methods is necessary before such an exchange can be successful. The creation of such a standard is a gap that future work can fill. In the meantime, our system can be extended with (semi)automatic submission to anti-phishing services and block lists, which would hopefully lead to quicker responses.

4.11. CONCLUSIONS

In this chapter, we have presented the results of our investigation of the Dutch phishing landscape. We designed an empirical method to study phishing campaigns in the wild using fingerprints derived from phishing kits. We leverage that phishers are using TLS certificates to capture the end-to-end life cycle of phishing campaigns. We were able to find 1,363 confirmed phishing domains that deploy such kits in a four-month time period – with, on average, 31 phishing domains online every day, waiting for victims to arrive. Most of these domains are online for only 24 hours, but half of them (much) longer. External validation with APWG data has shown that our methodology has the potential to detect phishing websites swiftly and that it covers a complementary spectrum of phishing domains. Additionally, we show that attackers have increased their abilities to lure victims into disclosing their credentials by using decoy pages, which do not directly demand credentials from the victim but do so eventually. These decoy pages split the target organization from the organization impersonated on the phishing page, which allows for numerous possibilities for attackers. Referring to the target organization in the domain name is less prevalent, as regular words are more often used to trick victims into clicking on a phishing link. Through a combination of our analysis of the anatomy of phishing kits and the crawls of phishing domains in the wild, we demonstrate that the Dutch phishing landscape is less diverse than expected and that many

phishers are building their campaigns on the same framework, *uAdmin*. The arrest of the developer of this framework in February 2021 and the corresponding news coverage allow us to conclude that our findings are also useful outside the Netherlands, as uAdmin is actively used all around the world. Through both data analyses and a real-world example, we have reconstructed a timeline of the complete end-to-end life cycle of a typical phishing campaign – proving that phishers move fast. In turn, these fast-moving campaigns require swift and decisive interventions. We believe the insights of this work will help LEA and intermediaries design faster responses to this ever-evolving threat, and we encourage them to do a similar analysis of their local phishing landscape.

5

GOVERNANCE INSTRUMENTS IN THE ANTI-ABUSE ECOSYSTEM

Various governance instruments aim to fight Internet abuse – from legislation to take down copyrighted material to blocklists to stop spam. In turn, these instruments rely on industry standards to handle abuse: reporting abuse to the network owners requesting mitigation. Although many hosting providers swiftly take action to keep the Internet clean, some do not. This raises the question as to what type of abuse receives follow-up and what rationale is behind a decision to either mitigate or ignore abuse. Through a unique collaboration with law enforcement in the Netherlands, we were granted access to the operational back-end of a hosting provider with a reputation for abuse. A rare peek at its internal abuse handling allowed for the investigation of what mechanisms in the anti-abuse ecosystem influence anti-abuse actions. We find that client notification rates highly depend on the reporter and abuse category. CSAM and spam-related abuse reports lead to mitigating actions, whereas reports regarding copyright and port scanning are neglected. Governance instruments like blocklisting, de-peering, and law enforcement inquiries that could directly hurt business continuity affect client notifications, whereas individual abuse reporting is easily ignored. We hope our work can inform policymakers to align governance repertoire with abuse handling in practice.

This chapter is under submission as: **Bijmans, H.L.J.**, van Eeten, M.J.G. & van Wegberg, R.S. (2025). "Tickets to Hide: Scrutinizing the Anti-Abuse Ecosystem with Internal Abuse Data". *Under submission at NDSS'26*

5.1. INTRODUCTION

To combat abuse – like spam emailing, hosting harmful content, or sharing copyrighted material – the Internet relies on the practice of abuse reporting, also known as 'notice and takedown'. Here, an abuse report is sent to a service provider, who subsequently notifies its client [116]. As a hosting provider, handling abuse reports for servers operated by many clients requires an effort. Industry best practices recommend swift follow-up for abuse – though different priorities are assigned to different types of abuse [152]. Adhering to these best practices is voluntary, though some jurisdictions have basic legal obligations for the hoster to evaluate abuse reports. However, evaluating can also mean deciding not to act. New legislation in the European Union – e.g. the Digital Services Act [65] – has codified a "notice and action" procedure and introduced "trusted flaggers" – designated independent entities whose abuse reports should be acted on with higher priority. This shows that we rely heavily on abuse reporting as a governance mechanism in the fight against online harms.

Although abuse reporting has been around for years [108, 152], there is limited scientific insight into the abuse-handling processes of hosting providers. Prior studies have taken an external viewpoint of hosters and analyzed how abuse events – like phishing sites, spamming, and malware C&C servers – are distributed across providers [219, 230, 232]. Some researchers leveraged external characteristics of providers to identify potentially malicious networks [128, 219]. However, high concentrations of abuse can also occur at legitimate providers simply because of the size of their infrastructure [219]. Hence, it does not tell us much about the network operator's lack of actions. Are they trying to mitigate the abuse or actively facilitating it? Operators who are perceived to not swiftly respond to abuse reports are referred to as 'bad' or 'bulletproof' hosters – that is, they are seen as impervious to abuse reports [4]. Some characteristics of bulletproof hosting have been described in both earlier work [86, 180] and industry reports [153]. Labeling a network as 'malicious' or calling a company a 'bulletproof' hoster assumes that the hoster is at least knowingly ignoring abuse reports or even enabling abuse. However, examining intent requires an inside view instead of network characteristics. To the best of our knowledge, only a single study has provided such an inside view on a bad hoster [180], but it did not analyze abuse report handling. However, comparing inside and outside views on cybercriminal operations is crucial, as adjacent research on dark market measurements found significant differences between the two [51].

To address this gap, we present a study of the internal abuse handling processes of a hosting provider known for abuse. In this study, we aim to answer the following questions: **RQ1**: "Which type of abuse is followed up on?", **RQ2**: "What factors influence the decision to follow up on abuse reports?", and **RQ3**: "How do external network characteristics relate to internal abuse handling?". We do not disclose the name of the studied hosting provider (and discuss the ethics involved in our research in 5.4.3), but it has been listed as one of the top bad hosters since the early 2010s [10], albeit under different brand names. It was referred to as a bulletproof hoster by both law enforcement [185], industry [10, 153], and media [134]. Over the years, the company regularly changed names, created new brands, and moved its registration to the Seychelles.

In 2020, the Dutch Fiscal Information and Investigation Service (FIOD) raided the company and seized company records [185]. Through a unique collaboration with this

agency, we acquired access to its operational back-end, which allowed us to investigate its abuse-handling practices. We collected 1.3M abuse reports from nine years of operation, categorized them, and identified corresponding client notifications. Furthermore, we investigate time-to-notify and notification rates over time per abuse category. Lastly, we connect our work to previous studies by relating our work to the aforementioned malicious network characteristics.

We find that notification rates highly depend on the reporter and the abuse category. Child sexual abuse material (CSAM) and spam-related reports do result in notifications, whereas copyright and port scanning reports have low client notification rates. Governance mechanisms like blocklisting, de-peering, and law enforcement inquiries that could directly hurt business continuity affect client notifications, whereas individual abuse reporting is easily ignored. We identify previously found indicators of malicious networks, yet argue that calling hosters bulletproof only based on external data is a tough label to sell.

In short, we make the following contributions:

1. We are the first to report on internal data from a hosting provider, leveraging over 1.3M abuse reports and over 9k client notifications spanning nine years.
2. We find that CSAM and spam-related abuse reports result in mitigating actions, whereas reports regarding copyright and port scanning are neglected.
3. We empirically show that abuse reports from reporters with a trusted status or who are in a position to hurt business processes have higher client notification rates than from individual reporters.
4. Through our longitudinal analysis, we find that government pressure and direct threats as a result of ignoring reports affect client notification rates.

The remainder of this chapter is structured as follows. First, we discuss the anti-abuse ecosystem and the public debate on anti-abuse legislation in § 5.2, followed by an overview of related work in § 5.3. Thereafter, we describe our dataset and reflect on the ethics involved in our work in § 5.4 and discuss our methodology in § 5.5. We present our results in two consecutive sections, take an inside look by quantifying abuse in § 5.6, and take an outside look through an analysis of malicious network characteristics in § 5.7. We contextualize our findings and discuss their limitations in § 5.8. Finally, we conclude our work in § 5.9.

5.2. BACKGROUND

This section describes the anti-abuse ecosystem and elaborates on the public debate on anti-abuse legislation.

ANTI-ABUSE ECOSYSTEM

Similar to the inner workings of the Internet, the ecosystem fighting Internet abuse is decentralized. Three roles are defined within this ecosystem: the abuse reporter, the intermediary – i.e., hosting provider – and the resource owner responsible for the abusive resource [116]. Every party has different incentives and possibilities for participation.

Table 5.1: Complaint priorities for abuse according to the M^{3}AAWG anti-abuse common practices [152].

Abuse Category	Priority
CSAM / Harmful content	Critical
Botnet C&C / DDoS attacks	High
Malware / Phishing / Brute-force attacks	Medium
Spam(vertising)	Low
Port scanning / Comment spamming	Very low
Copyright / Trademark issues	*Depends*

Abuse reporters have several incentives to voluntarily collect and report abuse data, such as altruism, quid pro quo [116], or being victims themselves. Intermediaries have a business relationship with the resource owner affected by monetary incentives and can decide to forward abuse reports to their clients. The Message, Mobile, and Malware Anti-Abuse Working Group (M^{3}AAWG) outlined the anti-abuse common best practices in 2015 to assist intermediaries in their anti-abuse efforts [152]. Besides guidelines to keep systems safe to prevent abuse – such as vetting new customers and keeping software up to date – it also outlines abuse report handling. This includes setting up an abuse e-mail account according to RFC2142 [108], making community abuse reporting straightforward, responding promptly to those reports, and considering trusted reporters to handle certain reports with higher priority. Additionally, the M^{3}AAWG presents a complaint prioritization that lists the prioritization for different types of abuse reports – shown in Table 5.1. In short, an intermediary can either *ignore* a received abuse report, *notify* the client and wait for it to be fixed, *assist* to fix the problem, *suspend* the server, or ultimately *terminate* the client.

ANTI-ABUSE LEGISLATION

Over the years, the decentralized structures to combat Internet abuse have been pressured by governments all over the world to safeguard the Internet. The resulting legislation differs significantly. The United States took the approach to criminalize computer fraud and Internet abuse in the Computer Fraud and Abuse Act of 1986 (CFAA) [77]. The bill, last updated in 2008, prohibits intentionally accessing a computer without authorization and committing fraud with a computer, but fails to state definitions or to present instruments to combat abuse online. As a result, many modern-day Internet activities can be prosecuted under the CFAA with severe punishments [53]. To stop the proliferation of child sexual abuse material (CSAM), the U.S. Senate introduced the "STOP CSAM" act, which makes reporting such material easier and adds administrative penalties when providers fail to remove CSAM within a certain period [60]. It would also institutionalize NCMEC's CyberTipline by requiring companies to report discovered CSAM material. The introduction was met with both enthusiasm and skepticism, as opponents fear it would jeopardize constitutional rights to privacy and freedom of expression as a result of risk-averse companies blocking any sensitive material. Legislation by the European Union has already come into force since February 2024 through the Digital Services Act (DSA), which aims to improve digital safeguards and to prevent illegal and harmful activities on-

line [65]. Besides codifying a "notice-and-action" procedure, nationally appointed Digital Services Coordinators award a number of organizations the "trusted flagger" status, whose abuse reports must be treated with priority. Leveraging expertise from organizations specialized in detecting abusive content could enhance the quality of abuse reporting and improve follow-up actions. Some large online platforms like Google or Meta have created similar programs, yet be it under their own terms [94, 160]. The DSA shifts this power to the E.U. member state governments and turns voluntary cooperation into mandatory compliance. Reactions to this new legislation have been mixed. Although it was warmly welcomed by civil rights groups, it was criticized by tech companies for creating a heavy burden and by some politicians and scientists [225] for undermining freedom of speech.

5.3. RELATED WORK

Past academic work that advances our understanding of Internet abuse can be branched into two categories: identification and analyses of malicious networks [128, 219, 259] – including bulletproof hosting [4, 130, 180] – and measurements of the (anti-)abuse ecosystem [144, 179, 232]. We summarize previous studies in the following section.

MALICIOUS NETWORKS

In 2009, Stone-Gross et al. [230] presented *FIRE* to actively monitor botnets, drive-by-downloads, and phishing website feeds to identify organizations and networks that show persistent malicious behavior. Shue et al. [219] augmented that by analyzing additional data sources and examining the BGP behavior of malicious Autonomous Systems (AS). They argue that ASes can be malicious due to either malicious intent by the operator or lax administration and poor security practices. Zhang et al. confirmed this [259]. Shue et al. [219] discovered that ASes with the most malicious activity have a greater number of BGP connectivity changes than benign ASes and that larger ASes are more likely to contain malicious IP addresses. Leveraging these BGP observations, Konte et al. [128] created *ASwatch* to identify malicious ASes by their routing behavior. They found additional indicators for malicious networks, such as aggressive AS rewiring (many changes in providers and peers), BGP routing dynamics – e.g., short prefix announcements – and fragmentation and churn of the advertised IP address space.

BULLETPROOF HOSTING

The aforementioned network indicators do not distinguish between networks that are willingly or unwillingly participating in abuse. Networks that intentionally participate in abuse are commonly referred to as bulletproof hosting (BPH). Such providers do not respond to reports of abuse originating in their networks [4], often rebrand their businesses by creating different storefronts [153], and offer different packages with tiers of permitted abuse [4] – closely resembling the M^{3}AAWG complaint priorities outlined in Table 5.1. In 2018, Alrwais et al. [4] observed that such networks moved from large malicious ASes – e.g., CyberBunker [130] – to fragmented infrastructure located at multiple lower-end service providers through sub-allocations. They report that in 2016, only 19.7% of IP addresses blocklisted by Spamhaus were directly allocated – i.e., managed by its service provider – whereas 80.3% were sub-allocations, half of them owned by a

client of a legitimate service provider. Investigating these sub-allocations showed that many legitimate service providers are not responsive to reports of abuse in their networks and heavily rotate IP blocks within their network to evade blocklisting. BPH services are registered as resellers with these service providers and practice a similar behavior of rapidly registering and dropping network blocks. Noroozian et al. [180] dug deeper into this phenomenon by analyzing a platform providing fragmented BPH infrastructure in 2019 called *Maxided*. Leveraging ground truth data extracted from seized back-end databases, they characterized its business model, supply chains, and clients. It was shown that the BPH landscape shifts from agile resellers towards marketplace platforms with an oversupply of hundreds of legitimate upstream hosting providers. Clients prepay the rent for their servers and treat them as disposable resources. The authors argue that calling this platform an *agile abuse enabler* is more fitting than calling it a bulletproof hoster, as it does not provide any resilience against takedowns, yet it does enable abuse.

ABUSE MEASUREMENTS

Jhaveri et al. [116] constructed a model of the abuse reporting infrastructure to study voluntary action against cybercrime within the anti-abuse ecosystem. By focusing on stakeholder incentives, they found only a few incentives for intermediaries to act on reported abuse since it often does not directly harm them. One incentive would be to prevent being blocklisted after ignoring abuse reports or reputation damages, although there is mixed evidence as to whether the latter incentive holds a significant effect in practice. Levchenko et al. [144] analyzed the spam value chain and showed that spam affiliate programs employed a very distributed hosting strategy in 2011. Tajalizadehkhoob et al. [232] investigated the providers behind networks with abuse to study how abuse levels are determined and if measurements are performed correctly. They observed that the structural properties of hosting providers – e.g., IP space size and domains hosted – and the prevalence of popular content management systems, like WordPress, explain most of the variance in phishing abuse counts and that expensive hosting providers are home to less abuse. Noroozian et al. [179] performed related research but took abuse feeds as input and found that all providers have an equal probability of reported abuse when controlled for exposure effects.

Previous work shows extensive efforts into the identification of malicious networks [128, 219, 230]. It observed a shift of abusive networks moving from monolithic ASes to resellers operating within sub-allocations [4] to platforms [180].Such analyses are based on both external observations – e.g., BGP routing [128, 219] – and ground truth data [180]. Following the systemization of Jhaveri et al. [116], we lack insights into one crucial stakeholder within the abuse ecosystem: the intermediary handling abuse reports. What makes an intermediary notify its client? And, how do the identified external indicators from previous studies relate to internal anti-abuse efforts?

5.4. DATA

Our analysis is based on data seized by law enforcement. On September 22nd, 2020, the Dutch Fiscal Information and Investigation Service (FIOD) raided the hoster, collected

Table 5.2: An overview of the two mailboxes and their contents.

Mailbox I (55,979 e-mails)	Items	Incl.	Mailbox II (2,624 e-mails)	Items	Incl.
Handled	1,227	✓	Handled	117	✓
Handling	0	✓	Handling	0	✓
Conversations	100		Conversations	7	
Deleted messages	4,236	✓	CP reports handled	1,682	✓
FMTS	4,965	✓	Deleted messages	12	✓
FMTS.FAPL	324	✓	Ignore	537	✓
Ignore	29,686	✓	Ignore - Cloudflare	276	✓
Ignore - Cloudflare	12,830	✓	Sent items	21,256	
Ignore - Netcraft fakeshop	2,653	✓			
Ignore - PhishLabs unauth host	58	✓			
Sent items	39,581				

Table 5.3: Selected database tables. *Including 14,644 accounts registered with the same email address – see *Clients*.

Database table	Count	Missing
`ticket`	2,350,168	6,685
`ticket_post`	2,815,503	6,721
`client`*	31,389	0
`devices`	2,023	0
`devices_events`	488,516	0
`ip_assignments`	8,407	23,468
`packages`	63,193	0

cash, seized bitcoin, and copied company records [185]. Through a collaboration with this agency, we were allowed to analyze parts of its back-end systems. This unique internal data allows us to perform empirical research that otherwise would not be possible. Note that this data assisted daily operations and, therefore, is not structured to support scientific research. Hence, we provide a detailed description of the data, discuss how we assessed its validity, and how we performed pre-processing.Legal and ethical considerations that come with the use of this data are discussed in § 5.4.3.

5.4.1. ABUSE MAILBOXES

As mentioned in § 5.1, the company operated under a variety of brand names over the years. Law enforcement was able to seize the mail servers of the last two brands and shared a copy of these mailboxes used for handling abuse reports. Both mailboxes were stored in `Maildir` format, which also saves the folder structure. Table 5.2 lists the folders, the emails it contains, and whether or not we included this folder in our dataset. Some folder names already suggest default follow-up actions, such as the folder names starting with *ignore*. In both mailboxes, we could find folders for handled reports in the

folder *handled* and folders related to certain reporters, such as Cloudflare. We omitted the *sent items* folders in both mailboxes, as manual analysis revealed no abuse reports nor client notifications, merely (automated) responses to reporters – e.g., we found 38,439 auto-replies to CyberTip demanding them to stop reporting through email and to use a provided takedown tool instead. The mailbox used by the last brand name contained 55,979 emails from 2019-02-01 until 2020-09-22, and the mailbox used by the second-to-last brand name contained 2,624 emails in the period 2019-03-07 until 2020-09-22. For every email, we extracted the timestamp, subject header, sender, and the message. Although we included the *deleted messages* folder in our dataset, emails could have been permanently deleted from these mailboxes.

5.4.2. OPERATIONAL DATABASE

The other dataset we use is the back-end database of a customer relationship management (CRM) system that the company employed to manage its operations. It was copied bit-wise during the raid by law enforcement, resulting in a 101GB database dump. Based on our research questions and in close collaboration with the involved law enforcement officers, we were granted access to a limited set of seven tables. These are listed in Table 5.3 and detailed in the following paragraphs.

TICKETS

Two tables enable communication with clients through tickets. Such tickets are related to password reset requests, overdue invoices, and abuse reports. The *ticket* table contains 42 columns, including a ticket identifier, client ID (when a ticket is created by a client), timestamp, creator email address, subject, and a message. It also includes the origin of a ticket, as they can be automatically created by e-mails directed towards a set of e-mail addresses (i.e., {abuse, billing, info}@company.com) or can be made by the company itself (i.e., because of late payments). Both the author of the ticket, the involved client, and employees can reply to a ticket. Those reactions are stored in a separate table, *ticket_post*. For our analysis, we joined the *ticket* and *ticket_post* tables to obtain an overview of all tickets and their responses – i.e., each ticket has one or more *posts* attached to it. As shown in Table 5.3, we found over 2.3M tickets and identified 6,685 missing tickets (0.3%) thanks to missing auto-incremented ticket IDs.

CLIENTS

The *client* table contains 57 columns related to personal details and billing information. To avoid analyzing personally identifiable information (PII), we only got access to the client ID, email, and registration date of each client. All other columns were removed before access was granted. A total of 31,389 clients were found in this table, without any missing values. Manual inspection of the email addresses used for registration revealed that someone registered 14,644 new accounts using the same email address in 2014. As we found no abuse related to them, we removed these automatically generated accounts from our analysis in subsequent sections.

IP ASSIGNMENTS & DEVICES

The remaining four tables can be used to determine the ownership of devices within the company's data center and IP address assignments to clients over time. First, two tables

capture the most up-to-date state of all devices and IP assignments. The *devices* table lists all devices and contains 34 columns related to, among others, every device's location in the data center, its status, and which client is associated with it. Upon removal of a device, records are not deleted but nulled. As a result, we found 2,023 devices in this table without any missing rows. IP addresses assigned to devices at the moment of the raid are stored in the table *ip_assignments*, which does not store any past assignments. When an IP assignment is removed, it is deleted from this table – which explains the 23,468 missing rows listed in Table 5.3. Since IP addresses are assigned to specific devices within the data center, we can leverage the *device* table to find which IP address was assigned to which device owned by whom. Again, just the final state is stored in this table. A combination of two other tables is necessary to gather information on historic IP assignments: *device_events* and *packages*. The *device_events* logs every update made to devices – ranging from client changes to power disruptions – in an appending log containing the before and after state. A total of 488,516 records, without any missing rows, were found. Lastly, some devices are shared by multiple users, each using a different IP address on one Virtual Private Server (VPS). VPS access is offered as a package, and records related to those are stored in the *packages* table, which contains information on both the client using it and the technical management of these packages.

5.4.3. LEGAL & ETHICAL CONCERNS

First, we detail the ethical considerations and privacy-preserving steps we took in handling the seized data. Then, we use the Menlo report [123] to outline how we dealt with the sensitive nature of our data for our analyses.

DATASET

In line with applicable laws and regulations, Dutch authorities were able to seize company records, including the mailboxes and CRM database. While we use data from a legal seizure, one should not assume that users were engaged in illegal behavior. Note that providing any evidence of any kind for any law enforcement effort is not the purpose of this study. Before back-end data was made accessible for academic research, public prosecutors weighed, among other things, the impact of the work on the rights and privacy of all parties. A Dutch law enforcement privacy officer vetted that our data subset was limited, only contained data vital to our research, and contained no PII. All of our analyses were conducted on-site at Dutch law enforcement agencies, where the data was stored and protected under their safety and security guidelines. We conferred with our IRB beforehand, and they viewed this work as outside of their jurisdiction, yet were satisfied with the assessments and applied procedures outlined above.

ANALYSES

In order to protect the privacy of clients, we took great care not to analyze any PII. This process was outlined by the involved privacy officer following strict regulations that go beyond the GDPR or IRB institutional frameworks and then implemented by Dutch law enforcement. As a result, we only had access to data essential to our analyses that was stripped from any PII by law enforcement before we were granted access. We only report on aggregated values and use (translated) excerpts of anonymized conversations in

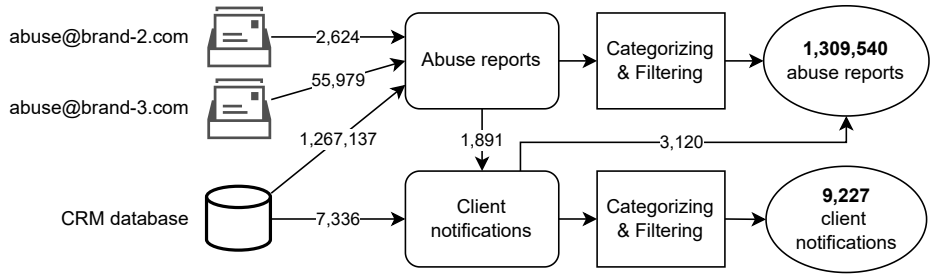


Figure 5.1: Overview of the use and processing of data sources.

abuse tickets. Extracting aggregate data points for our tables and figures was done under strict supervision through one specific monitored channel. To respect the privacy of all individuals involved, we do not refer to any person – neither clients nor employees – in particular. Both we and involved law enforcement professionals believe the benefits of a comprehensive understanding of the anti-abuse ecosystem outweigh the potential costs that come with making our work public. Additionally, to limit the potential negative effects on the company we are reporting on, we do not disclose its name but refer to it as "the company". In conclusion, we see no direct impact on persons, as we do not report on any individuals in particular and note that one should not assume any criminal wrongdoing on the part of these individuals.

5.5. METHODOLOGY

The following section details our approach to collecting and categorizing abuse reports and client notifications from the aforementioned data sources, depicted in Figure 5.1.

COLLECTING ABUSE REPORTS

We extract abuse reports from both the CRM database and the two abuse mailboxes. As mentioned in § 5.4.2, abuse reports directed towards a set of mailboxes are automatically stored in the database as tickets. To identify and extract those tickets, we selected all tickets originating from outside the company and filtered for tickets directed towards an '@abuse' e-mail address or containing the word 'abuse' in the subject header. This resulted in a set of 1,546,217 tickets, from which we removed 279,080 tickets containing spam emails, newsletters, and failed email delivery attempts, resulting in a set of 1,267,137 tickets. By analyzing the two abuse mailboxes, we obtained another 58,421 abuse reports in the period 02-2019 until 09-2020, which we analyzed similarly. The two data sources – database and mailboxes – were merged for further analysis, hereby omitting the 37 abuse reports that were present in both data sources. Duplicates were identified by comparing abuse reports with a similar subject header, originating from the same sender, and dated within a two-hour time frame from each other.

CATEGORIZING ABUSE REPORTS

As we had to perform our analyses on-premise at law enforcement agencies in secure environments, we adopted a simple keyword-based approach to categorize the collected

abuse reports based on their subject, author, and message body. Following the categories of abuse defined by the M^{3}AAWG (listed in Table 5.1), we brainstormed to gather an initial list of five keywords per category. Then, we followed a snowball sampling approach to gather more related keywords. We applied the initial list of keywords to the abuse reports, manually inspected the top remaining uncategorized reports, sorted by reports per sender (to affect the most significant portion of unlabeled reports), defined the label it should have, and identified new keywords to include these reports. We selected a random sample of reports per category (from a list aggregated by the sender), checked if errors had occurred, and fine-tuned the keywords afterward. This process was repeated ten times and resulted in the list of words that can be found in Table 5.4. Abuse reports can be assigned multiple categories (0.9%) or not assigned any category and labeled as "Unknown" (3.7%). Comparing 100 randomly sampled and manually labeled abuse reports with the assigned category revealed that our keyword-based approach was able to correctly categorize 95% of them.

DETERMINING CLIENT IP ASSIGNMENTS

To match abuse reports to their responsible clients, a mapping of the client using which IP address at any moment in our measurement period is required. To gather this information, we leverage the last four tables mentioned in 5.4.2 in the following order: *ip_assignments*, *devices* & *devices_events*, and *packages*. We apply a three-step approach to find the associated client based on the trustworthiness of the data sources. First, we search for an active IP assignment for any mentioned IP address at the moment of the incoming abuse report by using the *ip_assignments* table. If this yields no results, we search for historic IP assignments listed in the *device_events* table combined with the *devices* table to find the client that was associated with a historical IP assignment. If both searches do not yield any results, we use the *packages* table to retrieve a client. However, as this table does not contain an end date for IP address assignments – as it is tied to a contract with a set duration, e.g., a month – we consider this data source the least trustworthy. From 87% of these tickets, we could extract a company IPv4 address, which we were able to link to a client in 99.7% of the cases. For unknown reasons, we could not determine the corresponding client for 493 abuse reports containing a valid IP address

IDENTIFYING PROVIDER ACTION

Anti-abuse actions, such as client notifications, are stored in the CRM database in two ways. Either as a post linked to the abuse report ticket or as a separate ticket created by the company itself. The 1,891 client notifications in the database linked through posts are easily matched based on their corresponding ticket identifier. Matching the separately created client notification tickets is less trivial. To match these to their originating abuse report, we selected all company-initiated tickets not containing a set of 25 keywords related to billing, orders, and maintenance in the subject header. This set of 7,336 tickets was enriched similarly to the abuse reports – extracting the timestamp, category, and IPs. Since these tickets are directed toward a client, the corresponding client identifier is always present. We use the timestamp and the mentioned IP addresses in the client notification ticket to search for its underlying abuse report(s). For every client notification, we searched for abuse reports that mention the same IP address within a time frame of 336 hours – i.e., 2 weeks – prior to the client notification and did not

Table 5.4: List of words used to categorize abuse reports.

Abuse category	**Words**
CSAM / Harmful content	child, csam, cybertip, klpd, (CP), underage, iwf, kinderporno, inhope
Botnet C&C / DDoS Attacks	command-and-control, botnet server, c&c, ddos, dos, udp flood, portflood, dns-attack, dos attack, botnet controller, spamhaus botnet controller list
Malware / Phishing / Brute-force attacks	phish, fraudulent, malware, trojan, bruteforce, brute force, intrusion, hack, breakin attempt, break-in attempt, network attack, unauthorized login attempts, possible malicious activity from this host, unauthorized access, security incident, unallowed network access, service: ssh, auth_login authenticator failed for, sshd:auth, was found attacking, sasl login authentication failed, clean-mx-phishing, clean-mx-viruses, spyeye, malware distribution
Spam(vertising)	spam, spam email, unsolicited, clean-mx-spam, spam emitter, spam sites, clean-mx-portal, spamcop, spamming
Port scanning / Comment spamming	scanning, scanner, portscan, port scan, netscan, objectionable traffic, badbot regbot, clean-mx-trackback, port_scanning
Copyright / Trademark issues	piracy, dmca, copyright, piracy, infringement, urgent live stream escalation, notice of unauthorized use of, unauthorized sound recordings, notice of infringing activity, notice of claimed infringement, infringing material, fapl, counterfeit, balenciaga, yves saint laurent, tag heuer, dior, gucci, moncler, chanel, longchamp, trademark infringement, fake shop, torrent, warez, brand protection

Table 5.5: Dataset descriptives.

Dataset start	August 1, 2011
Dataset end (raid by LEA)	September 22, 2020
Abuse reports	1,309,540
Abuse reports linked to a notification	34,240
Client notifications	9,227
Client notification rate	2.6%
Abuse report senders	9,594
Registered clients	16,030
Abusive clients	3,114

yet contain a (linked) client notification. This time frame was chosen after a manual inspection revealed that such long time frames were not uncommon. As a result, one client notification can be matched to multiple received abuse reports. We consider this a valid method, as widespread abuse can trigger many different abuse reports, and it would make sense for a company to group those reports into one client notification. This is illustrated by one case in which we could match one client notification to 1,575 automated spam reports, all received in one day. Through this process, we were able to match 4,216 (57%) of the separately created client notifications to their initiating abuse report(s), hereby linking a client notification to 29,229 abuse reports. A notification is linked to 3.98 abuse reports on average, yet the median is 1. Abuse reports could also have originated from phone calls, as one response to an abuse report highlights: *for urgent cases, please call us, we noticed your email 9h later*. Hence, we added the remaining 3,120 client notifications, for which we could not find the originating abuse report, to the dataset as well, including every client notification ever sent by the company in our final dataset. Throughout our approach to measuring abuse handling, we apply a conservative take. That is, when in doubt, we assume the company notified the client.

To obtain our final dataset, we removed all uncategorized abuse reports not containing an IP address – 16,438 tickets, again mostly spam – and obtained our final dataset of 1,309,540 abuse reports. 87% of these reports contained a company IPv4 address, which enabled us to assign 1,120,201 (86%) abuse reports to 3,114 different clients. Matching client notifications to abuse reports and the linked notifications within tickets allowed us to identify 9,227 distinct client notifications linked to 34,240 abuse reports. Table 5.5 lists the final dataset to be used in the remainder of this chapter.

5.6. QUANTIFYING ABUSE & PROVIDER ACTIONS

In this section, we quantify and characterize the abuse reports and typologize categories, frequency, and origins. Next, we present insights into the actions taken by the company to combat abuse to answer our first two research questions.

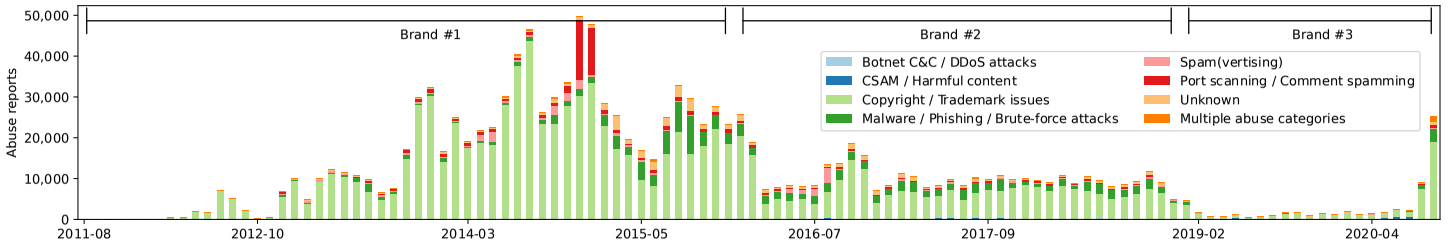


Figure 5.2: The number of monthly abuse reports divided by abuse category.

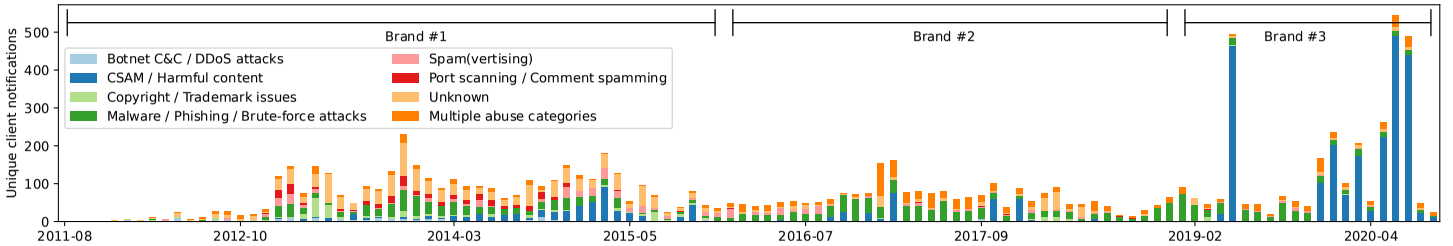


Figure 5.3: The number of unique monthly client notifications divided by abuse category.

5.6.1. ABUSE FOLLOW-UP

To answer our first research question, we scrutinize the company's 1,309,540 abuse reports received within the nine years spanning our dataset. An overview of the monthly received abuse reports is depicted in Figure 5.2. It shows roughly three periods of abuse volumes that line up with changes in brand names. For the first period (until 2016), we observed a slow increase in monthly abuse reports – growing from around 10k in 2013 to almost 50k by the end of 2014. Most of them are related to hosting copyrighted material but over time start to include reports in other categories too. In the second period (2015 – 2019), when operating under the second brand name, the number of abuse reports stabilized to around 10k monthly reports. This is mostly due to a shrinking number of copyright-related abuse reports; the other categories have similar abuse report counts within this time frame. Introducing the third brand name in late 2018 has likely changed the abuse-handling process. We believe abuse reports are no longer automatically converted into tickets but are only handled within the abuse mailbox, as we did not find many abuse reports in both the CRM database and in one of the mailboxes (only 37). It is likely that reports have been deleted from the mailboxes in this time frame and that the volume of abuse reports was higher. As in the last month for which we obtained abuse reports from the abuse mailboxes (September 2020), the number of reports returned to the monthly average of over 20k per month.

As listed in the first row of Table 5.6, copyright-related abuse makes up the largest portion of reports. Almost 77% of received abuse reports are categorized as such, which involves hosting torrent websites and illegal live sports streams. Other prevalent categories are (comment) spam(ing) and malware, phishing, and brute-force attacks. Reports related to ongoing DDoS attacks, botnet C&C servers, and CSAM are less prevalent. Reporters of copyright-related abuse do so very often, with an average of 609 reports per reporter. This is much lower for all other categories of abuse, which range from eight for botnet C&C servers and DDoS attacks to 40 for port scanning and comment spamming. In total, we found 3,114 clients to be involved in one or more abuse cases. Most abusive clients can be found in the malware, phishing, and brute-force category (1,980), followed by the port scanning and comment spamming category (1,575).

The next rows in Table 5.6 report on the client notification by the company, presenting both the number of reports with a linked notification as well as the number of unique client notifications. The latter is also depicted in Figure 5.3, which shows a very low number of copyright-related client notifications and many CSAM-related client notifications, especially in the last period. A total of 9,227 client notifications have been sent, which we could link to 34,240 abuse reports. Since abuse reports can be assigned to multiple categories, the client notification counts within this row add up to more than the total amount – we discuss the effects of this in 5.8. The overall client notification rate – i.e., what fraction of abuse reports are linked to a client notification – is 2.6%. If we remove the largest category – Copyright & Trademark issues – from this statistic, this number is 9.73%.

There are significant differences in client notification rates between the categories, and they change over time as well. For example, we could identify 867 client notifications for copyright-related abuse linked to 4,200 abuse reports, resulting in a 0.4% notification rate, whereas 45.6% of all CSAM-related reports are linked to a client

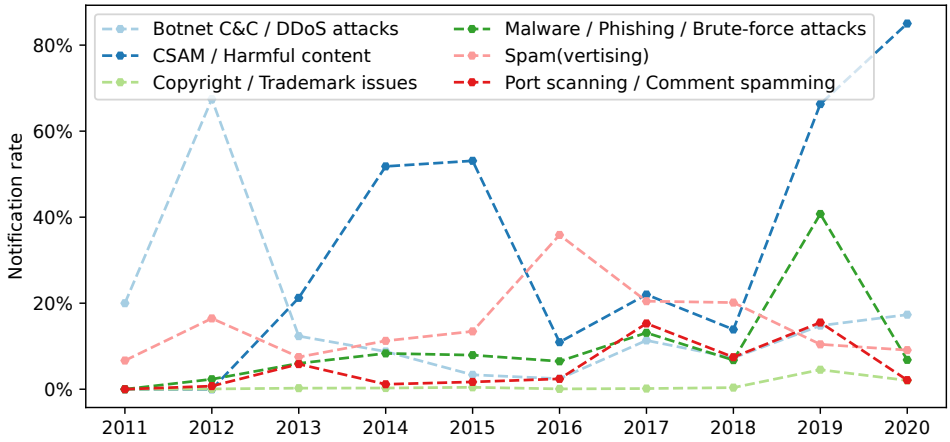


Figure 5.4: Yearly notification rates for the different categories.

5

Ticket conversation on 17/09/2016

Client: On my website, I will stream live sports events. About DMCA, can you confirm that there will be no problem?

Company: DMCA is USA law. We are not in the USA.

Figure 5.5: Conversation regarding the DMCA.

notification. Remarkably, the client notification rate for spam(vertising) (19%) is much higher than for malware, phishing, and brute-force attacks (8.6%), whilst their place in the prioritization according to the M^{3}AAWG in Table 5.1 would suggest the inverse. To investigate abuse handling over time, we plotted the client notification rates per year for each abuse category in Figure 5.4. It shows that the few botnet servers and DDoS abuse reports in 2012 were met with a 68%-client notification rate, which decreased in the years afterward. Client notifications originating from abuse reports related to (comment) spam increased slowly from 8% in 2011 to almost 40% in 2016 and decreased in the years afterward. Other categories remained at stable client notification rates below 20%. The years 2019 and 2020 marked a significant change for CSAM-related abuse reports, as the client notification rate increased to 66% and 85%, respectively.

Takeaway: The most reported type of abuse is related to copyright and trademark issues (77%), followed by malware, phishing, and brute-force attacks. Abuse reporters in the first category send out large amounts of reports (over 600 per reporter), whereas reporters in the other categories share fewer (9 - 40 reports per reporter). CSAM-related abuse reports are followed up on the most (45%), followed by spam(vertising) (19%). Notification rates fluctuate over the years.

Table 5.6: Overview of abuse reports per category, the involved clients and reporters, and corresponding notification rates.

	Botnet C&C DDoS	**CSAM Harmful content**	**Copyright Trademark issues**	**Malware / Phishing Brute-force attacks**	**Spam (vertising)**	**Port scanning Comment spamming**	**Unknown**	**Total**
Abuse reports	9,480	9,247	1,000,730	146,572	40,118	68,546	48,167	1,309,540
Reports with notification	1,050	4,213	4,200	12,614	7,640	2,647	3,621	34,240
Unique reporters	1,150	310	1,646	3,720	1,037	1,696	2,851	9,594
Avg. reports per reporter	8.2	29.8	609.5	39.5	39.0	40.5	16.9	136.8
Client notifications	757	3,125	867	2,862	1,436	1,046	1,587	9,227
Client notification rate (%)	11.1	45.6	0.4	8.6	19.0	3.9	7.5	2.6
Median time to notify (h)	27.5	0.0	123.7	67.7	34.0	46.6	44.5	48.0
Clients involved	1,221	334	832	1,980	907	1,575	1,664	3,114
Avg. reports per client	8.3	36.3	999.5	73.2	42.8	44.0	29.6	367.0

5.6.2. ABUSE FOLLOW-UP FACTORS

To understand what factors influence client notification rates, we examine each abuse category in increasing order of priority according to the M^{3}AAWG . Like abuse reports, most notifications are identically structured due to the use of templates. To gather more insights into the rationale of the company's operators, we searched for notifications containing non-standard messages or conversations with its customers. Some of them, which characterize the company's stance on abuse, are depicted in various figures as anonymized, translated quotes, some of which have been edited to improve readability.

COPYRIGHT & TRADEMARK ISSUES

A total of 1,000,730 copyright and trademark issues-related abuse reports have been received within our measurement period. These reports originated from 1,646 unique reporters; some reporters sent over 100k reports each. Among them are predominantly companies providing content security, anti-piracy services, and Cloudflare. The latter forwards received abuse reports, whereas the others filed the reports themselves. Many of the anti-piracy companies never got any response to their reports. An example is the now-defunct NetResult, a DMCA takedown service. It has filed 126,085 reports, of which only 411 could be linked to 4 client notifications. When client notification does take place, it happens slowly – as we found a 123.7 hours (5 days) median time to notify. Looking at the list of abuse reporters whose reports did result in a notification, we find several lawyers and BREIN, the private Dutch copyright watchdog. Their reports are not ignored, as multiple client notifications demand immediate action because *"we cannot afford problems with these people"*, according to the company. Clients get only 12 to 24 hours to mitigate these reported issues, and the company threatens to suspend servers if they do not do so. However, in some cases, the company thinks along with clients to allow their operations to continue. For example, after repeated requests from copyright holders acting for the English Premier League (FAPL), it emails a client to ask permission *"to change the IP address of this server, more offshore due to some issues"* instead of demanding them to take down services. A practice they later had to reconsider as soon as legal action was taken by the FAPL in 2018. Another option it offered was to host 'streaming relay' servers, which do not host any copyrighted material but merely relay it. The company seems willing to facilitate the streaming of any kind of material, as Figure 5.5 illustrates. Cloudflare, operating its distributed reverse-proxy service, forwarded over 30k abuse reports, 1,247 of them linked to client notifications. Although this is more than any other reporter in this category, we found that client notifications depended on the source of the original report and not because it originated from Cloudflare. For example, abuse reported by BREIN to Cloudflare is followed up on, whereas other reporters are ignored. The discovery of a folder called *Ignore - Cloudflare* in both mailboxes (see Table 5.2) underlines this finding. The company's stance on copyright-related abuse has changed over time. Although the client notification rates remain very low, as depicted in Figure 5.4, the company has been requesting that clients who stream copyrighted material provide a takedown tool on their websites since 2015, thereby transferring future abuse reports directly to the resource owner. Removing itself from the abuse reporting chain – i.e., reporters communicate with the client directly – explains the decrease in copyright-related abuse reports since late 2015.

Ticket conversation on 12/07/2020

Company: We are getting a lot of reports regarding your servers. They are generating reports on an hourly basis. It appears they are used for scanning services, which is only possible if people can opt out and you have an introduction page.

Figure 5.6: Conversation regarding scanning.

Takeaway: Enormous amounts of copyright-related abuse reports are ignored, except when there is a threat of legal action from piracy watchdogs like BREIN or lawyers. A compulsory takedown tool for clients operating streaming services reduced the volume of abuse reports by orders of magnitude.

PORT SCANNING & COMMENT SPAMMING

68,546 abuse reports make up this category, most of them related to port scanning. We can associate 1,575 clients with these complaints, with 40.5 reports on average. The majority of reports are very concentrated on a few clients. While one client was responsible for 29,623 abuse reports within two years, this does not seem to have influenced notification rates. Only in the rare case that a client receives an extraordinary amount of reports in a short period of time – as shown in Figure 5.6 – the company does notify. The second-highest number of 1,696 unique abuse reporters stands out because of the many automated abuse reports within this category. Among the top reporters are honeypot operators, intrusion detection systems, and data center network operators, who automatically file abuse reports after a port scan is detected on their servers. Only some of these reports resulted in client notification (3.9%). Spamhaus, the blocklist operator known for its fight against email spam, also lists illicit vulnerability scanners and comment spamming IPs in its blocklist. Such listings do trigger client notifications in 83% of the cases. Another type of abuse within this category is trackback abuse, a form of comment spamming. Blogging systems like WordPress allow for notification of new content on other blogs, which is abused by spam websites to promote their own content. One honeypot operator monitors abusive trackbacks and reports automatically, which was done 30,467 times without any response or notification.

Takeaway: Numerous abuse reports regarding port scanning and comment spamming from unvetted, automated systems – i.e., Fail2Ban – do not result in many client notifications. Abuse reports from Spamhaus do trigger frequent notifications.

SPAM(VERTISING)

We collected a total of 40,118 abuse reports related to spam(vertising), concentrated on several clients operating as resellers. Resellers, as mentioned in earlier work [4, 180], are frequent clients, especially in the category spam(vertising), and resell rented infrastructure to their clients. In doing so, they introduce another intermediary in the abuse notification chain. This is illustrated by one ticket: *"we are resellers, we can't control every client, and we didn't notice all the recent reports"*. Spam-related abuse reports received the second-highest client notification rate of 19%. 1,037 reporters have filed

Ticket conversation on 25/12/2011

Company: We hosted your website for a long time. Spamhaus listed your IPs, and we did not care. After that, they listed a few /23 blocks, and now they listed all our IPs because we did not take any actions. This affects our whole network, thousands of people can not email because of your domain.

Figure 5.7: Conversation regarding Spamhaus listings.

reports regarding spam activity, yet only one reporter received significant follow-up: Spamhaus. The Spamhaus Block List (SBL) contains IP addresses with known spamming activity [229]. As soon as an IP address is listed, the owner of the IP range is notified. If spamming is not handled within a certain period, Spamhaus can escalate the listing to block extended ranges – e.g., a \24 range – or eventually list the entire network of the involved AS. This happened several times, as we learned from the ticket conversations in Figure 5.7. Such 'escalation listings' bother the company because clients complain that their emails can not be sent or demand new IP addresses outside the listed ranges. As a result, SBL listings are handled swiftly, and temporary solutions are offered, such as email relays through non-blocklisted IP addresses. As a result, 60% of its reports lead to client notifications. Another party that received significant follow-up from its reports is Level 3, a peering partner. Individuals who encounter activities that violate Level 3's acceptable user policy can file a report, which Level 3 forwards to the network operator. The company, possibly afraid to lose connectivity, created 93 client notifications based on Level 3 reports, of which at least 35 were related to spam(vertising).

Takeaway: Although categorized as a low priority by the M^{3}AAWG , spam-related abuse is met with the second-highest client notification rate due to sanctioning by Spamhaus.

MALWARE, PHISHING & BRUTE-FORCE ATTACKS

This category is the second-largest category of abuse within our dataset, totaling 146,572 abuse reports from 3,720 different reporters. This category also involves the most clients, namely 1,980, with 39.5 abuse reports on average. The most abusive client has gathered over 18k reports and has a long business relationship with the company as a reseller offering offshore VPSes. Despite the many abuse reports, this client has never been terminated. The second client on this list, amassing 3,764 abuse reports, is another reseller offering unmanaged VPSes and was threatened to be terminated. After receiving many abuse reports – and forwarding just a handful of them – the company decided to terminate all its servers. However, after some back-and-forth, business continued as usual. Reports regarding dictionary or brute-force attacks come from the majority of reporters and are often the result of intrusion detection systems with automated abuse reporting. Fail2Ban, a popular system to protect (Web)servers from brute-force attacks, can also automatically send an abuse report to the owner of an IP address after a certain number of failed login attempts. At least 57,562 (39%) abuse reports within this category have been the result of this system. Such reports rarely lead to client notifications. Phishing reports originated predominantly from NetCraft

and PhishLabs, both take-down services that vet abuse reports thoroughly and provide detailed information, thereby facilitating swift client notifications. Additionally, services like NetCraft monitor reported phishing pages over a period of time to ensure their takedown. Although the folder names in Table 5.2 would suggest otherwise, 72% of NetCraft phishing reports resulted in client notifications, and 54% of the PhishLabs reports. Malware reports originate from various reporters – like community services and country CERTs – and receive varying notification rates.

Takeaway: The category with the most linked client notifications shows that vetted, trusted abuse reporters are met with higher client notification rates than automated systems like Fail2Ban or individual reporters.

BOTNET C&C & DDOS ATTACKS

We could identify a total of 9,480 abuse reports related to Botnet C&C servers and DDoS attacks within our measurement period. A total of 757 client notifications were sent, which we could link to 1,050 abuse reports, resulting in an 11.7% client notification rate. The median time to notify is the second-lowest, namely 27.5 hours, which seems in line with the priority assigned by the M^{3}AAWG [152]. Abuse reports originated from 1,150 different reporters – many filing only a single report – and are evenly distributed between DDoS attacks and botnet C&Cs. Among the top reporters that report botnet C&C servers are Spamhaus, a botnet researcher, and a Dutch SOC. Unlike the name suggests, Spamhaus also fights botnets by operating its Botnet Controller List (BCL) [229]. Similar to the reports related to spam(vertising), the influence of Spamhaus is evident, as 84% of its abuse reports were met with swift client notifications. Clients get just six hours to resolve reported issues and are automatically suspended if there is no immediate reaction. For DDoS reports, there are no reporters who file significant amounts of reports – most of the reports originate from direct victims of DDoS attacks who were attacked by (one of) the company's servers. The use of automation in abuse reporting causes noise for abuse-handling departments. An example of such is an automated DDoS reporting system sending out the same abuse report every 15 seconds. Unlike other categories, the company also detects DDoS attacks itself through its data center monitoring systems. When a high volume of outgoing packets is detected – e.g., a client sending spoofed packets – the company steps in and notifies the resource owner since such volumes could damage their network – as depicted in Figure 5.8. From Figures 5.3 and 5.4, we learn that this happened frequently between 2012 - 2015, and diminished in the years after. Four clients were terminated because of DDoS-related abuse, the only category in which terminations took place.

Takeaway: DDoS is the only type of abuse predominantly reported by direct victims. Outgoing DDoS attacks hurt the company's network as they affect the connectivity of other clients and are, therefore, quickly addressed. Botnet C&C servers listed by Spamhaus are removed rapidly as well.

CSAM & HARMFUL CONTENT

On top of the M^{3}AAWG priority scheme, we find CSAM or other harmful content. We identified a total of 9,247 such abuse reports within our measurement period, associated

Ticket conversation on 07/01/2012

Company: Hello, your server is sending DDoS attacks for a few hours now. We do not allow our network to be abused for DDoS. To protect other clients on our switch, we limited your port speed.

Figure 5.8: Conversation about an ongoing DDoS attack.

with 334 clients, having 29.8 reports on average. 3,125 client notifications were created, which we could link to 4,213 abuse reports. Abuse reports originated from 310 unique reporters. Among the very active ones are national hotlines that cooperate within the InHope network, such as the British IWF and the Dutch Meldpunt Kinderporno. Their reports were taken care of to a certain degree (notification rates of 16% and 24%, respectively), whereas reports from individual reporters received almost no follow-up. The clients associated with the reported abuse hosted either forum boards or operated image hosting services. In both cases, clients are given a maximum of 24 hours to handle reports. For example, one client operating multiple image hosting websites is responsible for 585 CSAM-related abuse reports in four years. Most of these reports led to a client notification and swift action from the notified client. However, after four years of abuse reports, law enforcement stepped in and forced the company to shut down this website. Another client received 382 reports and operated multiple forums from 2017 until 2020. From 2018 onward, all reports related to these forums led to client notifications, followed by the deletion of files by this client. In 2020, with the Dutch Justice Department putting more pressure on bad hosting companies [203], the company suggested stopping business with this client, as depicted in Figure 5.9. Ultimately, no client was ever terminated due to CSAM-related abuse. Government pressure likely resulted in the launch of a website to process takedown requests operated by the company. The effects of this platform are significant, as 94% of the reports filed through this platform resulted in a swift client notification – which explains the median time to notification of 0 hours in Table 5.6 and the increase in client notification rates in Figure 5.4. Many notifiers successfully used it, except for CyberTip, a Canadian initiative to fight CSAM. After repeated messages, the company sets up an autoresponder to instruct CyberTip to use their platform instead of emailing their reports. Despite the 38,425 sent auto-responses (discovered in the *Sent items* folder in one of the mailboxes), CyberTip never did so. In communication with clients, the company's standpoint is clear – they only take action when certain parties – e.g., law enforcement – demand them to. They explicitly state this as an excuse to their clients – e.g., *"please understand we only sent you this because authorities demand us, we don't want to play judge ourselves"*.

Takeaway: Most CSAM-related reports are met with swift response. Government actions and trusted notifiers with access to automated takedown portals have an effect, as client notification rates increased massively in 2019 and 2020.

Ticket conversation on 14/07/2012

Company: We can better stop working together. Have a look at your ticket history, you have had over 200 CSAM reports this year, that is way too much. The government is pushing very hard on us to fix this.

Figure 5.9: Conversation regarding CSAM reports.

5.7. EXTERNAL NETWORK CHARACTERISTICS & ABUSE

We now take an external look to answer our third research question and relate external network indicators found in previous work [4, 128, 153] to internal abuse handling by scrutinizing historical IPv4 prefix announcements.

METHODOLOGY

To gather insights into the company's IP presence, we leverage the RIPE NCC announced prefixes API [204]. Here, we collect historic IPv4 prefix announcements by the ASes associated with the companies in the operational database within the same time frame (2011 – 2020). While collecting this data, we noticed an abnormally large IPv4 range to be announced by this company. These 262,144 IPv4 addresses were part of the AFRINIC heist [114]. Since we found only 10 abuse reports related to IPs in this large range and their existence is disputed, we excluded them from further analyses. The resulting dataset consists of 727 historic announcements related to 259 IPv4 prefixes, totaling 39,936 IPv4 addresses spread over three different ASes operated by three different brands.

RESULTS

In Figure 5.10, we plotted the number of announced IPv4 prefixes over time per AS. We observe a slowly increasing number of 13,000 to over 17,000 announced IPv4 addresses in the period 2011 - 2015, except for two quarters in 2021, which we consider erroneous data. In 2015, the company rebranded for the first time and moved its registration to the Seychelles [185], which did not concur with significant changes in IPv4 prefix announcements. However, in 2016, its second brand – which had been active since late 2011 on a different AS –stopped announcing IPv4 prefixes until 2019. In early 2019, after another rebranding, all IPv4 prefixes were transferred from Brand #1 to Brand #3. At the same time, the second brand started announcing a few IPv4 prefixes again. Figure 5.11 shows the yearly additions and removals of IPv4 prefix announcements over time. Like previous academic studies [128], we witness a churn of over 2,000 IPv4 prefixes in the advertised IP space per year. Although the total number of announced IPv4 prefixes remains relatively stable, around 16,000, the total number of involved (i.e., ever announced) prefixes is 25,344 – a churn rate of 59% over nine years. IP space fragmentation and (very) short prefix announcements, other indicators for malicious networks identified by Konte et al [128], are present as well. 117 (45%) of the 259 announced IPv4 prefixes are small /24 IP addresses, and 124 (48%) prefixes are even single

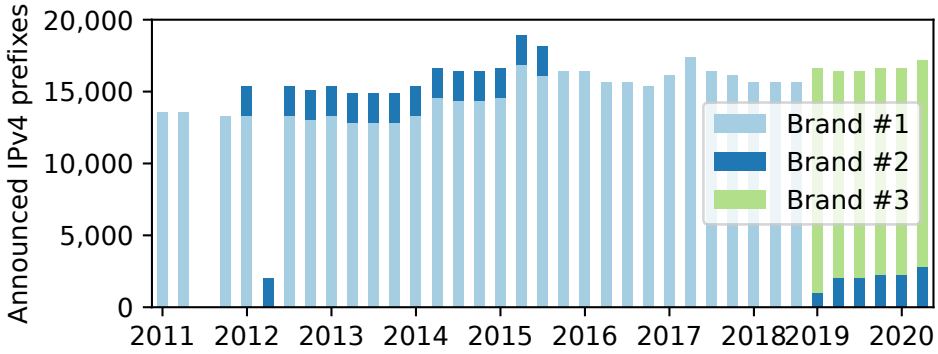


Figure 5.10: Quarterly IPv4 prefixes per AS.

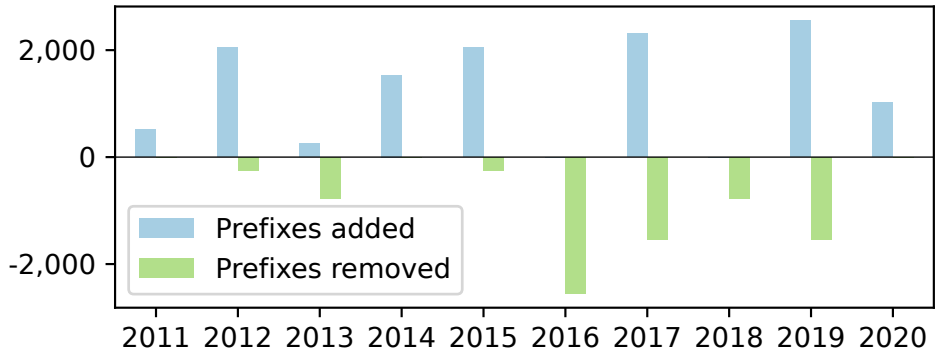


Figure 5.11: Yearly IPv4 prefixes additions and removals.

/32 blocks. 47% of all IPv4 prefix announcements last less than half a year, as depicted in Figure 5.12. For this last analysis, we only included the prefix announcements from 2013 onward to account for the missing data in 2012. All ASes, except Brand #3, are home to more than 50% prefix announcements lasting less than half a year. Following the findings of Alrwais et al. [4], this could be the result of rotating IP blocks to evade blocklisting, as we have seen anecdotal evidence for in 5.6. The last brand has only 28% of such short announcements.

Takeaway: Indicators for malicious networks are observed at this company, such as frequent rebrands, IP churn, IP space fragmentation, and short prefix announcements. These indicators can indicate evading actions to prevent blocking.

5.8. DISCUSSION

In this section, we return to the term bulletproof hosting, discuss the public policy takeaways of our findings, and elaborate on the inherent limitations that arise from our work.

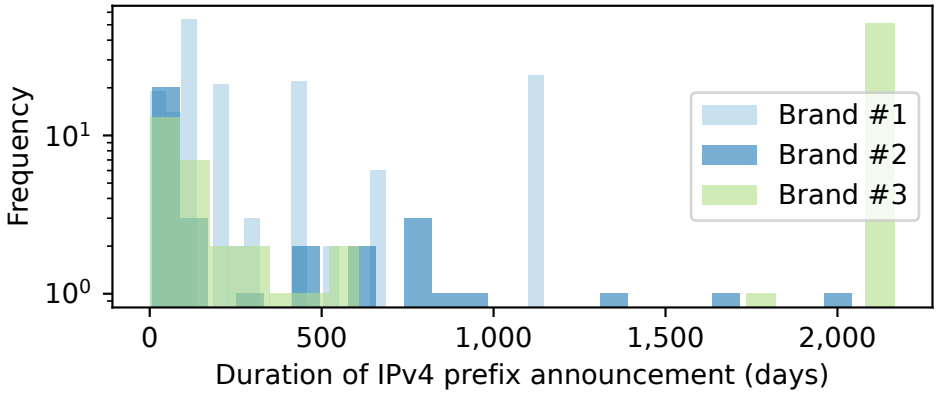


Figure 5.12: Duration of IPv4 prefix announcements per AS.

BULLETPROOF HOSTING

The term bulletproof was first coined by industry reports [86] and later found its way into academic work [128, 130] to describe hosting providers systematically and intentionally ignoring abuse reports. The analyses of such work are dominated by external viewpoints. We argue that bulletproof hosting – i.e., a behavioral pattern of purposely ignoring abuse reports – cannot be deduced solely from an external perspective, as it is impossible to measure intent without knowing internal abuse-handling processes. Measuring takedown rates of abusive content does indicate how willing a hosting provider is to fight abuse. However, without an inside look, it remains unknown to what extent neglecting abuse reports is a result of an intermediary not notifying its clients or clients not removing their abusive content. In our case, the studied company is often referred to as a bulletproof hoster by both law enforcement [185] and researchers [153]. Through similar analyses as performed by earlier work [128] in § 5.7, we do see indicators of malicious networks as well. However, our analysis of its abuse-handling processes shows that it was not impervious to abuse reports, as it did act upon a – be it small – portion of the received reports. Although some would question the morality of the company's decisions, our analysis and many of the tickets do not show up-front intents of enabling abuse. We do see, however, that it puts an absolute minimum effort into anti-abuse actions and solely prioritizes minimizing negative business effects. As a result, Spamhaus listings (that could hurt client connectivity), Level3 reports (that could result in de-peering), and CSAM-related abuse (with legally binding actions) are met with swift client notifications. In contrast, individual phishing, spam, or port scanning reports are not. Hence, our analyses show that the term bulletproof, when leveraging only external measurements, is a tough label to sell. The term assumes intent, which can only be captured through an insider view.

PUBLIC POLICY TAKEAWAYS

From § 5.6, we learn that certain instruments lead to abuse follow-up, whereas others do not. Some reporters within the industry have gained significant power and have hereby obtained de facto trusted reporter status. Abuse reports from these trusted reporters – e.g., CSAM hotlines – or reporters who can pressure the company into taking action – e.g., Spamhaus escalated blocklisting – result in more client notifications than individual reporters. This implies that individual abuse reporting, either manually – e.g., after receiving a phishing email – or automated – e.g., Fail2Ban abuse reports – seems less effective. Abuse reports originating from automated reporting systems operated by individual networks result in many similar yet unstructured and less detailed abuse reports that are more likely to pollute abuse mailboxes than assist abuse-handling personnel. Security practitioners wanting to report abuse could, therefore, better report to trusted or powerful reporters instead. Our analyses have shown this to be functioning in the case of phishing (NetCraft), spam (Spamhaus), and CSAM (InHope hotlines). Recent E.U. regulations to institutionalize and appoint "trusted flaggers" through the DSA [65] seems a deliberate action to make the Internet safer. However, we question why these trusted flaggers are appointed nationally. It makes sense to appoint trusted flaggers for copyright-related abuse per member state since copyright laws differ per jurisdiction. However, to fight spam or malware-related abuse, there is no need for 27 different nationally appointed trusted flaggers performing similar work as Spamhaus is currently doing. Here, European or even worldwide trusted flaggers would seem more effective.

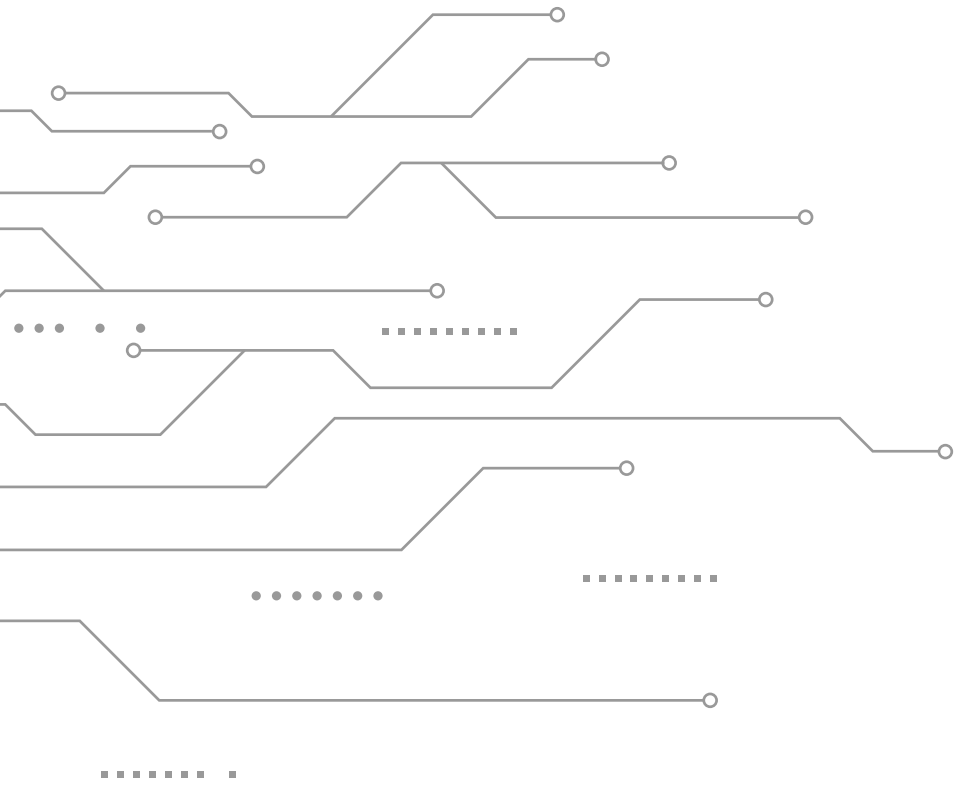
LIMITATIONS

First, our study is focused on a single hosting provider, which inherently limits the generalizability of our findings. This is, however, not the goal of our study – by no means do we argue that the observed numbers are generalizable to the entire hosting industry. We merely use our observations on abuse-handling processes to study the effects of governance mechanisms to fight abuse. Second, our analyses could be hampered by missing or deleted data. We trust our law enforcement partners to have gathered and copied both the database and the mailboxes correctly during the seizure. However, the fact remains, as we listed in Table 5.3, that over 6,000 tickets have been deleted from the database (0.2%), and we suspect email deletion from the two mailboxes used for handling abuse reports as well. The latter is probably due to a change in abuse handling processes, as we expect the company to handle at least a part of the abuse reports directly from the mailbox instead of from their ticketing system, as they did before. Additionally, some abuse reports, as well as client notifications, could well have never been part of our datasets. They could have been handled by phone or other means of communication. Third, our method of categorizing abuse reports is straightforward. As listed in Table 5.4, the word 'DMCA' in the report's subject would classify a report as copyright-related. However, this rigid categorization is not always correct and could have influenced our results. On top of that, from 2017 on, the company notified abusive clients 220 times with emails containing the subject header `URGENT: MALWARE / PHISHING / SCANS / SPAM` and the corresponding IP address, making these notifications match all of these categories. Lastly, some communication was not as structured as we assumed in our methodology. An example of this is a nine-month-long email thread between the company and one of

its clients consisting of 56 client notifications and responses. Restructuring such outliers within our large dataset was considered impossible. However, we do not think solving such issues would have affected our findings significantly.

5.9. CONCLUSIONS

In this study, we empirically investigated internal abuse data of a hosting provider with a reputation for abuse to study the effectiveness of governance instruments in the anti-abuse ecosystem. By gathering 1.3M abuse reports and 9,227 unique client notifications, we find large differences in client notification rates among different reporters and abuse categories. That is, CSAM and spam-related abuse reports do result in client notifications, whereas reports regarding copyright and port scanning have low client notification rates. Through our longitudinal analysis, we find that reporters with either a trusted status – e.g., NetCraft, InHope hotlines – and governance instruments like blocklisting (Spamhaus), de-peering (Level3), or governmental pressure that could directly hurt business continuity affect these client notification rates, whereas individual reporters of abuse are easily ignored. We find a mismatch between the severity of certain abuse types and their corresponding anti-abuse governance instruments. We identify previously found indicators of malicious networks at this company, yet argue that calling this company bulletproof based solely on external measurements is a tough label to sell. Our findings support recent E.U. legislation that appoints trusted flaggers to fight abuse, but we question the implementation per member state for certain types of abuse.



6

TECHNICAL MEASUREMENTS AND CYBERCRIME POLICING

Over the past two decades, researchers have advanced large-scale technical measurements of cybercrime to analyze techniques, tactics, and procedures (TTP) in cybercrime operations. These quantifications are typically based on analyses of technical artifacts such as domains, binaries, or attack traffic and could potentially inform cybercrime policing – i.e., assisting law enforcement agencies (LEAs) in determining how their scarce resources can be best put to use. Yet, we do not know if this potential is being used, nor how such measurement studies align with LEA needs. This chapter investigates the nexus of large-scale technical measurements and cybercrime policing by combining a survey of previous scientific work with a user study involving LEA professionals. We leverage the concept of value chains to structure 38 studies featuring measurements of phishing, booter services, and remote access trojans (RATs). We scrutinize their data sources and characterize their findings to identify common denominators. Then, we let LEA professionals reflect on some of these measurements and jointly identify the unexplored potential for novel measurements that align with current needs in cybercrime policing. We find that many academic studies focus on components in the value chain that are considered less valuable to LEAs and that most measurements lack geographical or attacker differentiation, thereby not allowing for concrete action perspectives.

This chapter has been published as: **Bijmans, H.L.J.**, van Eeten, M.J.G. & van Wegberg, R.S. (2025). "A Measured Response – On the Nexus of Large-Scale Technical Measurements and Cybercrime Policing". In *Proceedings of the 24th Workshop on the Economics of Information Security (WEIS '25)*.

6.1. INTRODUCTION

Cybercrime – referring to crime facilitated or committed by using a computer [193] – has grown worldwide [76] and even surpassed traditional crime in damages in some countries [32]. Traditionally, LEAs relied heavily on criminological data collection efforts like victimization surveys [202] or police reports [59, 113] for designing and evaluating their interventions. However, it is a well-established fact that the majority of cybercrime incidents are never reported to the police [59, 113], so the actual amount of cybercrime could be much higher than law enforcement agencies (LEAs) estimate. To understand modern-day cybercrime, computer scientists create novel detection methods to perform large-scale technical measurements and capture data on cybercriminal techniques, tactics, and procedures (TTP). Such measurements have the potential to inform law enforcement in policing cybercrime in a robust and scalable way. Yet, we do not know if this potential is being used, nor how such measurement studies align with LEA needs.

In recent years, a wealth of large-scale technical measurements of cybercrime have been published. Many of these studies revolve around the creation of innovative detection methods, which are afterward deployed to examine a subset of the Internet to assess their workings [54]. For example, within one year, Konoth et al. [127], Wang et al. [250], and Kharraz et al. [124] set out to create a robust method able to detect browser-based cryptojacking. Other studies examined phishing through large-scale Web scraping [15, 58, 126, 231, 260, 261], studied DDoS attacks trough network telescopes [96, 117, 132] or deployed honeypots to discover RAT operators [74, 201] Such measurements demonstrate how well a newly designed method functions, but these studies have not been performed with a focus on generating new insights into cybercriminal prevalence or TTP.

Demonstrating the success of a novel detection method can be straightforward. Performing robust measurements of cybercrime and assessing, for example, prevalence is, however, far from trivial. In an overview of cybercrime studies, Clayton et al. [41] found that the existence of concentrations of cybercriminal activity often leads researchers to suggest that such concentrations are potential vantage points for law enforcement interventions. While economic factors cause some of those concentrations, others are the result of measurement biases that mislead researchers into drawing wrong conclusions. Even without such issues, it is unclear what value these studies represent to LEAs. Additionally, in 2013, Anderson et al. [7] reported on the insufficiency of cybercrime statistics and encouraged governments to put more effort into detecting and prosecuting cybercrime. Their follow-up study in 2019 reconfirmed their findings, demanding governments increase detection efforts to improve cybercrime statistics [5]. Recent academic studies involving large-scale technical measurements could assist with these efforts. However, we do not know how such measurements can assist LEAs. To the best of our knowledge, no prior study has investigated how law enforcement professionals assess the value of academic studies featuring large-scale technical measurements of cybercrime for policing efforts.

In this chapter, we combine a literature survey and a small user study involving LEA professionals to address this gap and to answer the following two research questions. First, **RQ1:** What are the characteristics of large-scale technical cybercrime measurements? Second, **RQ2:** How do LEAs evaluate these measurement studies regarding their alignment with policing needs?

We answer the first question by surveying the top computer security venues for large-scale technical measurements of cybercrime published in the last fifteen years (2007–2023). For feasibility, we scope our search to three cybercrimes that are extensively empirically studied and received widespread attention by LEAs worldwide [68, 70, 71], namely phishing, booter services, and remote access trojans (RATs). Measurement studies on these three cybercrimes help us derive relevant characteristics of research that might influence whether a study is more or less aligned with LEA needs. Such characteristics are then relevant for measurement studies beyond these three cybercrimes. To structure our analysis, we leverage the concept of cybercrime value chains [248], which dissects cybercrimes into components and resources. For each study, we identify the components in the value chain it investigates and review its data collection, methods, and findings. Next, we report on a workshop with LEA professionals that elicits their assessment of a sample of studies and answers our second research question. Additionally, we let participants generate ideas on how to improve measurements to assist in cybercrime policing.

By combining our survey of 38 studies with the LEA workshop's results, we find that academic measurements focus on the deployment and execution of cybercrime, whereas LEAs desire to learn more about development and monetization. We observe that the majority of measurement studies lack geographical or attacker differentiation, thereby not allowing for concrete, actionable perspectives for law enforcement.

In short, we make the following contributions:

1. We survey 38 large-scale technical cybercrime measurements and characterize datasets, methods, and findings, which we combine with the first-ever assessment of scientific measurements by law enforcement.

2. We find that measurements often focus on the center part of the value chain (deployment and execution), whereas LEAs value insights at the ends of the value chain (development and monetization) to support cybercrime policing.

3. We observe that current measurement approaches largely overlook geographical and attacker differences, resulting in less actionable measurements for LEAs.

The remainder of this chapter is structured as follows: we detail our methodology in § 6.2 and present the results of our survey in § 6.3–6.6. Then, we report on our workshop with law enforcement in § 6.7, critically discuss our work in § 6.8, and conclude in § 6.9.

6.2. METHODOLOGY

This section first introduces the concept of cybercrime value chains, explains our focus on the three selected cybercrimes, and elaborates on our methodology to survey past scientific work.

6.2.1. CYBERCRIME VALUE CHAINS

To structure our synthesis of past scientific work, we leverage the concept of *cybercrime value chains* [248] to map large-scale measurements to value chain components. As cybercrime relies on a delimited mix of resources to turn a profit [248], we can structure

components in the value chain with a required set of resources as an input and a resource as an output. For example, as illustrated in Figure 6.2, to deploy a phishing page, one needs a phishing kit and a domain as input, which results in a phishing page as output. Components in the value chain are not strictly sequential, as some components can be executed in parallel (e.g., a booter service can deploy its storefront website and perform reconnaissance operations simultaneously). Instead of fulfilling all components in the value chain themselves, modern-day cybercriminals rely on specialists to fulfill specific components for them [14, 104, 248]. Specialized third parties provide resources, which can be products (e.g., software) as well as services (e.g., hosting infrastructure). For every outsourced component, a specialist will take a cut. Hence, to avoid these cuts or to expand operations, cybercriminals can decide to self-organize components in the value chain. We refer to this practice as *vertical integration* [239]. Only a few, such as Thomas et al. [239], have studied the cybercrime ecosystem with such a holistic view. As we demonstrate in this work, most scholars focus their measurements on specific components of a value chain instead of considering the complete value chain.

6.2.2. SELECTION OF CYBERCRIMES

We survey a large body of such measurements to study the characteristics of large-scale technical measurements of cybercrime and assess their value for law enforcement. However, as scrutinizing all past measurements of every type of cybercrime is near impossible, we scope our survey to three types of cybercrime that received significant attention from both academics and LEA, and argue that the identified characteristics are then relevant for measurement studies beyond these three specific cybercrimes. To discover what types of cybercrime measurements are valuable for LEA, we search Europol's 'Internet Organised Crime Threat Assessment' (IOCTA) from 2019 until 2023 [68, 70, 71]. These reports give an overview of the cybercrime landscape, as well as the efforts LEA has made to police it. Cybercrimes mentioned in these reports include ransomware, DDoS attacks, Business Email Compromise (BEC) fraud, dark markets, phishing, bulletproof hosting, botnets, and many more. We constructed value chains for each cybercrime, determined whether measurements were possible at different phases along the chain, and initiated an initial literature search to find large-scale technical measurement studies. It turns out that certain cybercrimes allow for more measurements than others. For example, we could find a plethora of phishing measurement studies in the top venues, but none related to BEC fraud. Consequently, we selected three types of cybercrime that received attention from LEA and were measured frequently by scientists, namely: *phishing*, *booter services*, and *remote access Trojans (RATs)*.

6.2.3. SURVEY APPROACH

We employ a systematic approach to search for studies performing large-scale technical measurements of one of the selected cybercrimes. We start by selecting conferences and journals. First, we take the Google Scholar top ten computer security conferences and journals [93], supplemented by their relevant co-located workshops. We extend this list with computer science venues focused on Internet measurements and cybercrime. Table 6.1 contains the complete list of included conferences and journals. In June 2023, we used the ACM Digital Library and IEEE Xplore to search for papers published in these

Table 6.1: Conferences and journals included in our literature survey.

Type	Conference or journal
Top 10 Computer Security	IEEE Symposium on Security and Privacy (S&P)
	IEEE Transactions on Information Forensics and Security
	ACM Symposium on Computer and Communications Security (CCS)
	USENIX Security Symposium
	Computers & Security
	Network and Distributed System Security Symposium (NDSS)
	IEEE Transactions on Dependable and Secure Computing
	International Conference on Theory and Applications of Cryptographic Techniques (EUROCRYPT)
	International Cryptology Conference (CRYPTO)
	Journal of Information Security and Applications
Co-located workshops	USENIX Workshop On Offensive Technologies (WOOT)
	USENIX Large-Scale Exploits and Emergent Threats (LEET)
Internet measurements	ACM Internet Measurement Conference (IMC)
	Passive and Active Measurements (PAM)
Cybercrime	APWG Symposium on Electronic Crime Research (eCrime)
	International Symposium on Research in Attacks, Intrusions, and Defenses (RAID)
	Workshop on the Economics of Information Security (WEIS)
	ACM The Web Conference (WWW)

conferences or journals since 2007 – covering the last 15 years of academic research. Conferences and workshops that are not in these libraries were searched manually. To be included, the paper title or abstract must contain one of the following terms: *phish*∗, *booter*∗, *ddos*, *rat*, *remote access trojan*, *cybercrime*, *cyber crime*. Manual title and abstract screening was performed on the resulting 615 papers. Through this process, we identified 32 papers that contained a large-scale technical measurement of one of the three selected cybercrimes. We added 6 other relevant works discovered during our literature research. In total, we found 38 studies, of which 19 papers measure phishing, 14 examine booter services, and 5 investigate RATs. We examine these papers through the lens of cybercrime value chains. We first identify which value chain component(s) the measurement leverages. Next, we classify measurements as either passive (based on an existing dataset, involving no scanning for artifacts or interaction with infrastructure) or active (based on an active collection of artifacts, involving scanning for cybercriminal infrastructure, etc.). Finally, we extract data sources and collect time ranges.

6.3. BOOTER MEASUREMENTS

In a DDoS (Distributed Denial of Service) attack, a server, service, or network is flooded with a massive volume of traffic, rendering it unavailable to legitimate users. Executing a DDoS attack requires significant resources and technical capabilities. To allow low-skilled criminals to perform such attacks, criminal entrepreneurs have set up so-called *booter* or *stresser* services, which offer DDoS attacks as a service. In this section, we examine large-scale technical measurements of booter services in earlier work. The booter service value chain is depicted in Figure 6.1. Here, we identify four components. First, during *reconnaissance*, the attacker finds the resources (e.g., vulnerable protocols) and scans for infrastructure to abuse. During *deployment*, criminal entrepreneurs set up shop and organize their attack servers, domains, and their protections. *Execution* involves the actual DDoS attack, involving both a client and a target, and *monetization* revolves around all financial aspects of running a booter service. Although DDoS attacks can also originate from botnets or nation-state actors, several studies concur that booter services exert a significant impact on the DDoS landscape [132, 181, 238]. We find 14 large-scale technical measurements of booter services published between 2013 and 2021, list them in Table 6.2, and detail them in the next paragraphs.

DATASETS

Two main data sources are used to study booter services: booter operations data found in databases or website scrapes and DDoS observations through honeypots or darknets. Since 2013, a variety of booter operations databases have been examined. First, an analysis of *TwBooter* by Karami & McCoy [120], followed by an analysis of 14 booter services by Santanna et al. [210]. A year later, Karami, Park & McCoy [121] scrutinized the databases of *Asylum Stresser* and *Lizard Stresser*, and complemented their data collection with website scrapes from *vDOS*. Thomas, Clayton & Beresford [238] used website scrapes and API logs from *vDOS* together with the leaked database of *CMDBooter* to validate their honeypot measurements. The *vDOS* database was used again in 2017 by Brunt, Pandey & McCoy [26]. The other data source used to study booters is honeypots, which capture amplification DDoS attacks. Deploying such honeypots allows researchers to track at-

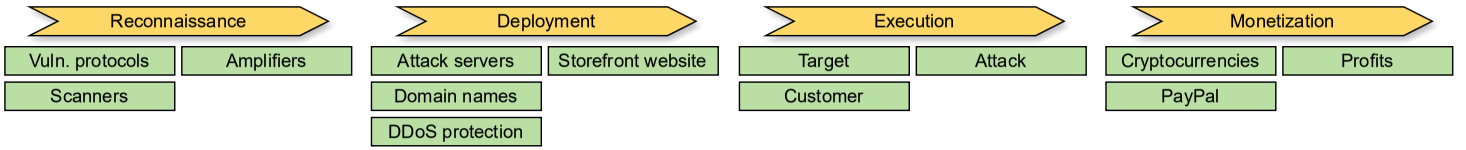


Figure 6.1: Booter value chain with its components at the top and the required resources at the bottom.

Table 6.2: Overview of large-scale booter service measurements. Measurements can be active (*A*), passive (*P*), or both (hybrid, *H*).

Authors	Year	Value chain				Data sources					Misc.			Time frame
		Reconnaissance	Deployment	Execution	Monetization	Databases	Website scrapes	Honeypots	Darknets	ISP/IXP	Measurement	Attack purchased	Intervention	
Karami & McCoy [120]	2013		●	●	●	●					*H*	✓		01/2013 – 03/2013
Chromik et al. [35]	2015		●				●				*A*			09/2014
Krämer et al. [132]	2015	●		●				21	●		*P*			2007 – 2015
Santanna et al. [210]	2015		●	●	●	●					*H*			2011 – 2014
Santanna et al. [211]	2015		●	●			●			●	*H*	✓		2013
Karami, Park & McCoy [121]	2016	●	●	●	●	●	●				*P*	✓	✓	2011 – 2015
Noroozian et al. [181]	2016			●				8			*P*			2014 – 2015
Brunt, Pandey & McCoy [26]	2017			●	●	●	●				*P*		✓	2014 – 2016
Jonker et al. [117]	2017			●				24	●		*P*			03/2015 – 02/2017
Thomas, Clayton & Beresford [238]	2017	●		●		●		60			*P*			2014 – 2017
Collier et al. [47]	2019			●			●	60			*P*		✓	2014 – 2019, 2017 – 2019
Kopp et al. [131]	2019			●			●			●	*P*	✓	✓	2018 – 2019
Griffioen et al. [96]	2021	●		●				549	●		*H*			08/2019 – 11/2019
Kopp, Dietzel & Hohlfeld [129]	2021			●				●		●	*P*			09/2019 – 04/2020

tacks over time and was initiated by Krämer et al. [132], with the design of *AmpPot* – a honeypot network that was deployed at 21 locations worldwide. The same honeypot was later used by Noroozian et al. [181], which deployed eight of them in Japan, and by Jonker et al. [117], which deployed 24 of them worldwide. Thomas, Clayton & Beresford [238] designed a different honeypot to measure amplification attacks and deployed ±60 of them. Data collection continued, allowing for two more years of data to be analyzed by Collier et al. [47]. The most recent honeypot measurement was presented by Griffioen et al. [96], in which an entirely new honeypot network was built. This study deployed 549 instances in five public clouds worldwide, demonstrating that the number of honeypots needed to obtain sufficient attack coverage is much higher than shown in the earlier work. Some studies complemented their honeypot measurements with data from darknets – unused IP ranges, also known as network telescopes [96, 117, 132].

RECONNAISSANCE

Scanners are deployed to find vulnerable infrastructure to be used in DDoS attacks, which can be observed through darknets. Deployment of such scanners was limited before 2012, as found by Krämer et al. [132]. Since then, scanning for DNShas gained more popularity and increased for more protocols in 2014. However, by attributing scanning IP addresses, Krämer et al. [132] found that over 40% of all scanners are operated by universities and security organizations instead of DDoS providers. Thomas, Clayton & Beresford [238] claimed to have excluded such 'white-hat' scanners from their data and reported on 5,070 IP addresses scanning daily. During the measurement, they noticed an increase in NTP and SSDP scanning and a slight decrease in DNS scanning. The largest honeypot network to study reconnaissance was deployed by Griffioen et al. [96]. It confirmed the work of Krämer et al. [132] by observing prevalent research scans (30% of all scans), finding that responsive IPs make scanners come back twice as fast, and noticing the same packets used for both testing and attacking. Additionally, by periodically switching honeypots between active and passive mode, the existence of a 'memory' of previously exploited servers was discovered, indicating that attackers track vulnerable servers instead of opportunistically selecting them for their attacks. A similar pattern was learned from leaked booter databases by Karami, Park & McCoy [121] that noticed booters gravitating to using more stable amplifier infrastructure when possible instead of scanning for vulnerable machinery.

DEPLOYMENT

During deployment, booter operators assemble their attack infrastructure and create a website to serve customers. In 2013, Karami & McCoy [120] examined *TwBooter* and revealed that only 15 servers were used to perform their attacks, most of them hosted in the Netherlands. Three years later, such servers were purchased by Karami, Park & McCoy [121], which allowed them to conclude that the required high uplink bandwidth can be obtained with small investments. A different approach was discovered by Santanna et al. [210] who found that all but one (*TwBooter*) based their attack infrastructure on Web shells instead. Web shells are scripts that allow backdoor access to compromised machines, making them part of the DDoS infrastructure. Scrutinizing 42 booter websites by Chromik et al. [35] learned that websites calling themselves 'stressers' did so to avoid

legal problems. Most websites show a verbose page with text and appealing advertisements to sell their services. Additionally, they noticed that nearly all websites have DDoS protection. This was confirmed by Santanna et al. [211], who tracked 102 booter websites over time and observed increased use of such protection since 2011. In 2014, none of the analyzed booter services were unprotected. This finding was confirmed by Karami, Park & McCoy [121], together with the notion that this was also to hamper take-down by law enforcement.

EXECUTION – CUSTOMER

Behind every DDoS attack launched by a booter is a customer. The various leaked databases give interesting insights into the customer base of booters. Karami & McCoy [120] identified three types of customers: gamers (launching attacks of less than 10 minutes), website attackers (launching attacks of one or two hours), and privileged users (performing long attacks for more than two hours). Most of the *TwBooter* customers could be categorized as gamers, attacking roughly three targets per day for a short period. Santanna et al. [210] highlighted the importance of differentiating between registered users and paying customers of booters, as the latter is significantly smaller. It turns out that many users are just attracted to take a look at what a service can offer, whereas only a few are interested in performing attacks. Karami, Park & McCoy [121] and Brunt, Pandey & McCoy [26] drew similar conclusions, as they found that only roughly 13% and 15-23% of the users in their examined databases ever paid for an attack, respectively. Multiple booter databases showed differences in OPSEC of customers as well [210]. Frequent customers are more likely to take precautions by obfuscating their real IP address – e.g., by using a VPN or Tor.

EXECUTION – TARGET

Several studies tried to map booter service attack targets. Karami & McCoy [120] categorized most customers as gamers and, therefore, concluded that most targets were game servers and forums. Subsequent analysis of leaked databases by Karami, Park & McCoy [121] learned that targets are predominantly residential links and gaming-related servers, with only a handful of higher-profile targets, such as government, media, and law enforcement websites. A finding later confirmed by Brunt, Pandey & McCoy [26]. Observing ongoing DDoS attacks through honeypots by Krämer et al. [132] showed that victimization rates differ much per country. The U.S. stands out (one-third of all victims), followed by China (14%), and France (8.6%). Similar numbers were reported by Collier et al. [47] four years later. Additionally, Krämer et al. [132] found that 79% of victims are targeted just once. Analysis of the attacked ports showed a special interest in gaming-related services, such as Xbox Live, Minecraft, and Steam. A similar analysis performed by Jonker et al. [117] observed that attacks targeted at HTTP(S) are most prevalent for TCP, whereas the most attacked ports for UDP are associated with various online multiplayer games and Steam. As many websites are hosted on IP addresses operated by large hosting companies, it was evinced that 64% of the `.com`, `.net` and `.org` websites were hosted on IP addresses ever targeted by DDoS attacks. Targeted networks were aggregated by their infrastructure by Kopp, Dietzel & Hohlfeld [129], who showed that content hosting networks were attacked the most (37%), followed by access networks such as ISP s (35%). Using the same honeypot technology as Krämer et al. [132],

Noroozian et al. [181] published the most comprehensive work on booter victimization in 2016. It demonstrated that most attacks are directed towards users in access networks and not at hosting or enterprise networks. The number of victims in an ISP network is proportional to the number of ISP customers, just as the victimization rate in a hosting network is proportional to the number of hosted domains. For the identified Web hosting victims, the authors discovered almost no high-profile targets, whereas the largest victim group was again gaming-related websites, mostly related to Minecraft. Noroozian et al. state that "in the Minecraft community specifically, DDoS attacks seem to be part of the culture" [181]. The authors speculate that attackers and victims of booter services are geographically close, and the low entry barrier of booter services allows victims to easily become attackers themselves. Griffioen et al. [96] concluded that victimization has changed in 2021. The U.S. and China are still popular among attackers. Yet, a disproportional share of attacks on South Africa, Poland, and Kuwait was found – 51% of all targeted IP addresses had an associated domain name. There were also numerous DDoS attacks on residential IP address space.

EXECUTION – ATTACK

Earlier work has characterized booter DDoS attacks by, for example, attack duration and used protocols. An overview of such studies is presented in Table 6.3. It shows that most research effort is put into analyzing UDP reflection or amplification attacks. Only four studies [117, 120, 121, 210] report on the use of the TCP protocol in DDoS attacks. Looking at the popularity of UDP protocols for abuse, we observe a constant dominance of DNS and NTP, even though their disclosure as DDoS amplification vectors was made long ago [129]. Collier et al. reported on the rise of LDAP as a protocol in DDoS attacks, but this was never confirmed by later research [47]. Kopp, Dietzel & Hohlfeld [129] analyzed several other new attack vectors – such as WS-Discovery, ARMS, and OpenVPN – and confirmed the active abuse of these protocols for DDoS attacks, yet not at the scale of traditional ones. Daily attack numbers are hard to grasp, as each study covers a different dataset and measurement approach. Yet, the numbers in Table 6.3 show a declining number of daily attacks while attack duration remains relatively stable. Parallel attacks introduce complexity in counting the number of attacks. For example, should a booter customer that launches both an NTP and a DNS reflection attack toward a victim be counted as one or two attacks? Santanna et al. [210] noticed that 32% of attacks have been launched in parallel, which means that new attacks against the same target are launched during an ongoing attack. However, they also find that 38% of users do not perform such attacks and perform just one attack per day on average. Attack durations differ per protocol [96] and victim type [181]. Both Thomas, Clayton & Beresford [238] and Noroozian et al. [181] noticed spikes in their attack duration measurements. They observed many attacks with a specific duration of either 5, 10, 60, or 120 minutes, likely caused by booter services offering exactly these amounts. Lastly, Griffioen et al. [96] discovered a new adversary tactic – attack pulses – in which the attacker does not launch its attack as a continuous flow but in powerful, periodic pulses. This maximizes the attack power while minimizing the costs for the attacker.

Table 6.3: Overview of DDoS attack characteristics. Grey-colored rows are based on self-reported numbers; the others are based on network measurements. Protocols in bold are the most popular attack vectors. The average attack duration is converted to seconds, and the number of attacks is per day.

Ref.	Year	UDP	TCP	Attacks	Duration (s)
[120]	2013	DNS	SYN	-	-
[210]	2015	**UDP flood**	SYN	-	260
[132]	2015	**NTP**, DNS	-	10,235	62% ≤ 900
[121]	2016	**SSDP, DNS**	SYN	-	1,620
[181]	2016	**DNS**	-	7,844	272 – 300
[117]	2017	NTP, DNS	**HTTP(S)**	30,000	255 – 454
[238]	2017	**NTP**, DNS	-	5,120	50% ≤ 658
[47]	2019	**LDAP**, NTP	-	30,000	-
[131]	2019	**NTP**, Memcached	-	-	-
[129]	2021	**DNS**, NTP	-	809	±360
[96]	2021	**NTP**, SSDP	-	673	394

MONETIZATION

Four studies investigated the monetization strategies of criminal entrepreneurs. Database analysis by Karami & McCoy [120] learned that 277 active users subscribed to *TwBooter*, totaling a profit of $7,727 a month. A similar conclusion was drawn from analyzing *VDoS* databases by Brunt, Pandey & McCoy [26]. A median revenue stream of $25,985 was reported, with new customers making up the largest sum of revenue. Although the studied booter supported both PayPal and Bitcoin as payment methods, most profit was generated through clients paying with PayPal – which has ease-of-use that stands out compared to Bitcoin. The popularity of PayPal was also noted by Karami, Park & McCoy [121], who monitored the payment infrastructure of 23 booter services. In confirmation with Brunt, Pandey & McCoy [26], only a small portion of booters accepted Bitcoin payments. Santanna et al. [210] learned that most paying customers paid only once to perform their attacks, and over 50% of customers paid $5.00 or less for the booter's services. Although booters offer differently priced services, the cheapest services are the most popular.

INTERVENTIONS

As booter services impact the DDoS landscape, interventions either by law enforcement [47, 131] or as part of scientific studies [121] have focused on disrupting their business. Both Karami, Park & McCoy [121] and Brunt, Pandey & McCoy [26] studied the effects of PayPal interventions on booter operations. While monitoring the payment infrastructure of 23 booter services, Karami, Park & McCoy [121] reported booter merchant accounts to PayPal. Through this intervention, the average lifespan of such accounts dropped from 8 to 3 days, and PayPal unavailability increased from 20% to above 60% in the days after. Most booters eventually added alternative payment methods, such as Bitcoin. A similar analysis was performed on *VDoS* by Brunt, Pandey & McCoy [26] and observed decreasing revenue as soon as PayPal was removed as a payment method. This hampered subscriber growth and eventually led to a decreasing user base. Only

11% of the existing customers switched to Bitcoin when PayPal was no longer available. The smaller user base also resulted in fewer attacks being launched, decreasing by 31% in their analysis period.

The effects of booter takedowns by law enforcement are studied by Kopp et al. [131] and by Collier et al. [47]. During the analysis of Kopp et al. [131], an FBI-led operation seized 15 booter services [244]. Those takedowns caused significant reductions in DDoS traffic to DNS, NTP, and Memcached reflectors, as observed from an IXP perspective, but no significant reduction in traffic from those reflectors to targets. Kopp et al. note that "seizing the domains of booter websites does not improve the situation for DDoS victims, as the underlying infrastructure of reflectors remains online and can be utilized by third parties without disruption" [131]. A longitudinal analysis of UDP amplification attacks by Collier et al. [47] revealed that after each police intervention, the number of observed attacks decreased significantly for a short period but kept increasing in the long run. Search engine adverts (discouraging the use of booters) and the closure of multiple booter websites had a longer-lasting effect on the booter market than arrests.

BOOTER TAKEAWAYS

In contrast to other cybercrimes, much ground-truth data (e.g., leaked databases) is available for research [26, 120, 121, 210], allowing for valuable insights into booter operations, attackers, and targets. As a result, all components in the value chain have been studied within the last 15 years. However, most measurements have focused on the execution component. Reconnaissance has been studied throughout the years, but although booters remain on the radar for LEA worldwide [70], insights into its deployment and monetization have not been gathered since 2016 and 2017, respectively. Another unique aspect of this cybercrime is the number of interventions that happened during measurements, which allowed researchers to show that most LEA interventions seem to have a short-term effect [47], whereas other interventions, such as on payment infrastructure [26, 121], seem to have a longer-lasting effect. Customers seem to be diverse, yet a large portion can be related to online gaming. Analysis of groups of booter customers showed great differences, both in terms of victim selection and OPSEC. Attack characteristics across studies show that long-established attack vectors (e.g., DNSor NTP amplification) remain popular while others exist. Research into booter operations also introduces noise, as scanning by security researchers is prevalent [132, 238]. This has to be considered to avoid measurement biases, especially in the reconnaissance phase [96]. Lastly, the popularity of analyzing the same booter databases in multiple studies [26, 121, 210] suggests that some measurements are driven by data availability.

6.4. PHISHING MEASUREMENTS

Phishing is the nefarious practice of harvesting user credentials through various means of deception. In our study, we specifically include work on phishing used for gaining direct profits – e.g., obtaining bank credentials to steal funds. We are aware that phishing can also be used to gain initial access to a (company) network, but we do not include such use in our survey. Illustrated in Figure 6.2, we present the components of the phishing value chain. *Development* is typically served through a *phishing kit* – an off-the-shelf package containing Web pages mimicking a company login page. Occasionally,

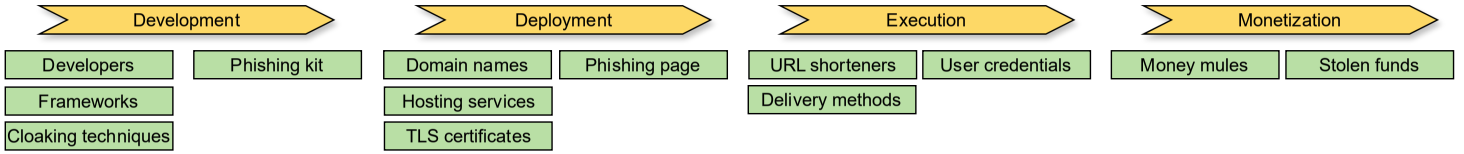


Figure 6.2: Phishing value chain with its components at the top and the required resources at the bottom.

they feature a back-end *panel* that allows access to the phished credentials. *Deployment* involves registering a domain name and acquiring hosting facilities. During *execution*, victims interact with the phishing website, followed by the *monetization* of the phished credentials. We find 19 large-scale technical measurements of phishing published between 2007 and 2022, which we list in Table 6.4 and detail in the next paragraphs.

DATASETS

A diverse set of data sources can be found in phishing studies, as evident from the overview provided in Table 6.4. Most studies gather a large body of phishing URLs or domains from a database of known phishing pages to analyze afterward. Until 2019, this source was predominantly PhishTank [195], a community-vetted database of phishing URLs. Until 2019, 7 of the 11 measurements relied on this source [49, 52, 140, 158, 167, 192, 241]. This shifted towards other databases such as the APWG eCrime Exchange [9] and OpenPhish [186] in the years that followed. From 2020 onward, combining different data sources gained traction [20, 58, 126, 192]. The introduction and adoption of Certificate Transparency (CT) [92] in 2018 - 2019 opened up new methods to discover phishing domains and was immediately put to use in several studies [15, 20, 126]. *WHOIS* records in phishing research are mostly used to complement other datasets [15, 58, 126, 158, 261]. In terms of data types, we observe a shift from (passive) domain and URL-based measurements [140, 158, 182, 222] towards active measurements using a crawler that inspects live phishing pages [15, 20, 58, 126, 231, 260, 261].

DEVELOPMENT – PHISHING KITS

The development component of a phishing attack is typically served through a phishing kit. The first study on such kits by Cova et al. [49] showed that more than a third contained obfuscated backdoors exfiltrating phished credentials to a third party. This was quantified by Peng et al. [192] years later, with 5% of phishing pages also disclosing credentials to third-party collectors. Additionally, Cova et al. [49] noted that all kits were written in PHP, likely because of the easy deployment on (shared) hosting servers. This was confirmed twice more, in 2017 by Thomas et al. [240], and in 2021 by Bijmans et al. [15]. Most kits impersonated one specific organization – mostly a U.S bank or PayPal –, and email was found to be the most frequently used method to deliver disclosed information. PayPal was also found to be most mimicked by Han et al. [98] in 2016 and by [192] in 2019. A similar analysis as Cova et al. [49] was performed years later by Bijmans et al. [15] by gathering phishing kits from Telegram and open server directories on live phishing pages. In contrast to Cova et al. [49], high utilization of *multipanels* was observed by Bijmans et al. [15] – a type of kit that targets multiple organizations simultaneously – and they found many versions of the same *uAdmin* phishing kit, which was the most deployed phishing kit at the time. Like Thomas et al. [240], it was found that only a small number of phishing kits are actively used by many different attackers. Email was no longer used to deliver disclosed credentials, as most kits contained a panel hosted on the same domain to access them instead. Additionally, the use of *decoy* pages was discovered, enabling multi-stage phishing attacks involving multiple brands in one campaign. Phishing websites were grouped through vector clustering on the DOM by Cui et al. [52], which allowed for the observation that attackers only search for a new

domain for their attack instead of modifying their phishing pages. However, as pointed out by both Thomas et al. [240] and Bijmans et al. [15], this could result from the same phishing kit being used by many different actors. Through an analysis of live phishing pages, Subramani et al. [231] report on phishing kit capabilities. A third of pages logged keystrokes and shared them with the attacker as soon as they were typed, and almost half of the websites required users to disclose their credentials in multipage Web forms, hereby confirming the findings of Bijmans et al. [15]. Although modern phishing sites impersonate a certain brand's Web page, 42% of phishing pages are not direct clones of the corresponding legitimate ones. This was also mentioned by Tian et al. [241] four years earlier, which concluded that such evasions would render visual similarity detection ineffective. Finally, a new type of phishing through man-in-the-middle (MITM) attacks was evinced by Kondracki et al. [126], circumventing multi-factor protections.

DEVELOPMENT – CLOAKING

To prevent phishing websites from being detected, attackers deploy *cloaking* mechanisms to thwart security researchers from accessing their phishing websites. Here, the server presents different content to scanners instead of regular visitors. The first analysis of such mechanisms was by Thomas et al. [240], who found that many phishing kits deploy a `htaccess` policy or employ a blocklist to frustrate visits from cloud providers, anti-virus brands, and anonymous proxies like Tor. A year later, 2,313 of such `.htaccess` files were examined by Oest et al. [182] to discover that blockades based on IP address, hostname, referrer, and User Agent are most common. Most `.htaccess` files are constantly reused and not kept up to date – as most of them were last modified over a year before their deployment. 23% of all phishing websites were found to implement client-side cloaking techniques in 2018 according to Zhang et al. [260], which grew up to 34% in 2019. The most common technique was the use of pop-ups (content remains hidden until a button in a pop-up window is clicked) and click-through interaction (content is shown when a visitor clicks somewhere on the page). Follow-up work by Zhang et al. [261] showed that intentionally triggering server-side cloaking behavior could be used as a method to detect phishing websites. Analysis of live phishing kits revealed that 96% employed cloaking techniques in 2022, and Kondracki et al. [126] showed 85% of MITM phishing kits did too.

DEPLOYMENT

The deployment of phishing websites in the wild was first studied by Moore et al. [167] by monitoring 1,685 domains from PhishTank [195]. They found an average uptime of 62 hours, with a median of 20 hours. Additional analysis of domains related to the *RockPhish* gang revealed the use of fast-flux domains, which resulted in a longer average uptime of 95 hours. As listed in Table 6.5, this research inspired others to perform similar measurements of phishing website life cycles. *WHOIS* records were used by McGrath et al. [158] to discover that most domains are used almost immediately after registration. Periodic DNS resolving of the examined domains revealed that, on average, a phishing domain lasts just over three days, but a third of all domains last only 55 minutes. An estimated lifespan of eight days for phishing kits installed in a honeypot was calculated by Han et al. [98]. Honeypot monitoring further revealed that attackers act fast when installing and testing their kits. This was quantified by Oest et al. [183], who found a

Table 6.4: Overview of large-scale phishing measurements. Measurements can be active (*A*), passive (*P*), or hybrid (*H*), with a worldwide (*W*) or local (*L*) focus.

Authors	Year	Value chain				Data sources							Data types					Misc.			Time frame
		Development	Deployment	Execution	Monetization	PhishTank	APWG eCX	OpenPhish	Other sources	CT logs	DNS zonefiles	WHOIS	Domains	URLs	Phishing kits	Web pages	E-mails	Measurement	Focus	Crawler	
Moore et al. [167]	2007		●	●		●							●	●				*A*	*W*	✓	02/2007 – 04/2007
Cova et al. [49]	2008	●	●			●			●					●	●			*P*	*W*		04/2008 – 05/2008
McGrath et al. [158]	2008		●	●		●			●		●	●	●	●				*H*	*W*		2007 – 2008
Han et al. [98]	2016	●	●	●											●			*P*	*W*		09/2015 – 01/2016
Cui et al. [52]	2017	●	●			●								●		●		*A*	*W*	✓	01/2016 – 10/2016
Thomas et al. [240]	2017	●		●					●						●			*P*	*W*		03/2016 – 03/2017
Oest et al. [182]	2018	●	●				●						●		●			*P*	*W*		2016 – 2017
Le Page et al. [140]	2018			●		●			●					●				*H*	*W*	✓	01/2016 – 12/2017
Tian et al. [241]	2018		●			●					●		●	●		●		*H*	*W*	✓	04/2018
Peng et al. [192]	2019	●	●	●		●	●	●	●					●	●	●		*A*	*W*	✓	09/2018 – 03/2019
Bitaab et al. [20]	2020		●	●			●	●	●	●			●	●		●	●	*A*	*W*	✓	01/2020 – 05/2020
Oest et al. [183]	2020		●	●	●		●		●					●			●	*H*	*L*		10/2018 – 09/2019
Simpson et al. [222]	2020		●								●	●	●					*P*	*W*		2009 – 2019
Bijmans et al. [15]	2021	●	●	●			●			●		●	●		●	●		*A*	*L*	✓	09/2020 – 01/2021
Kondracki et al. [126]	2021	●	●			●		●		●		●	●	●	●	●		*A*	*W*	✓	03/2020 – 03/2021
Zhang et al. [260]	2021	●					●							●		●		*A*	*W*	✓	06/2018 – 11/2019
Drury et al. [58]	2022		●			●		●	●	●		●		●		●		*A*	*W*	✓	07/2021 – 02/2022
Subramani et al. [231]	2022	●						●						●		●		*A*	*W*	✓	03/2022 – 05/2022
Zhang et al. [261]	2022	●					●					●		●	●	●		*A*	*W*	✓	11/2020 – 07/2021

one-hour window between the first attacker tests and the first victim. Most URLs were hosted on paid domain names, whereas only a very small portion used subdomains offered by free hosting services. Two-thirds of distinct URLs were served over HTTPS, but 86% of the compromised visitors visited over HTTPS, meaning that the use of HTTPS proved more successful than HTTP. A percentage that was much lower one year earlier, when 34% of websites were served over HTTPS according to Peng et al. [192]. Subsequent studies leveraged the TLS certificates for HTTPS connections to detect phishing websites in CT logs. Bitaab et al. [20] observed a spike in newly issued certificates for COVID-19-related domains during the pandemic, peaking at more than seven thousand per day. CT logs also allowed for the identification of 1,363 domains targeted at the customers of Dutch financial institutions by Bijmans et al. [15]. Further analysis confirmed the one-hour testing window found by Oest et al. [183], as most domains had a kit installed one hour after they first responded. A surprising amount of these domains were hosted (73%) and registered (34%) through Namecheap. In the same year, Kondracki et al. [126] found most MITM domains hosted at DigitalOcean – demonstrating the need for more demanding requirements (e.g., a VPS instead of shared hosting) for such attacks. TLS certificates were not used for detection by Drury et al. [58], but to timestamp phishing websites. They found TLS certificates to be often requested close to the occurrence of a phishing website on a blocklist – 27.6% were requested within 24 hours of inclusion. Date information included in file headers yielded insights into the resource's creation and indications of resource sharing among phishing websites.

DEPLOYMENT – URLS & DOMAINS

Much research focused on URLs and domain names used in phishing, which started relatively unstructured. From Cova et al. [49], we learn that 63% of live phishing kits were hosted on trustworthy domains with the targeted brand inserted in the path, and 30% of phishing URLs had no clear relation with the targeted brand. That phishing domains tend to be online for shorter periods than benign ones was discovered by McGrath et al. [158], whereas their URLs are typically longer. Additionally, phishing domains typically have fewer unique characters, and more than half of the examined domains contain the targeted brand name. In 2007, the first efforts were made to structure the analysis of phishing URLs. A taxonomy defining four types was proposed in [80], only to be used by Moore et al. [167]. It was later updated by Oest et al. [182] who proposed five types, used by Bijmans et al. [15] and Oest et al. [183]. Phishing URLs contain either an IP address as hostname and a deceptive path (Type I), a random domain and a deceptive path (Type II), a long and deceptive subdomain (Type III), a deceptive top-level domain (Type IV), or are unintelligible (Type V). 61% Type V domains were found by Oest et al. [182], followed by 21% Type IV domains. Follow-up work in 2020 by Oest et al. [183] showed fewer Type V domains, 29% Type IV domains, an increase in Type II domains (35%), and 28% Type IV domains. Almost solely Type IV domains (95%) were found by Bijmans et al. [15] by monitoring CT logs for phishing domains. Half of the domains did not contain references to the targeted brands, just deceptive keywords. The gTLDs `.info` and `.com` were among the most popular ones, followed by cheap TLDs such as `.xyz`. Similarly, Kondracki et al. [126] reported that combo-squatting (Type IV) and target embedding (Type III) were most prevalent but varied significantly per target for the MITM phishing websites. Typosquatting – e.g., replacing one character of a target domain name – was

Table 6.5: Phishing lifetime measurements.

Ref.	Year	Amount	Type	Uptime
[167]	2007	1,685	URLs	62h average, 20h median
[158]	2008	7,394	Domains	72h average, ±30% 55m
[98]	2016	474	Kits	192h
[241]	2018	1,741	Domains (sq.)	80% ≥ 1 month
[183]	2020	404,628	URLs	21h avarage
[15]	2021	1,288	Domains	45h average, 24h median
[126]	2021	1,220	Websites	40% ≥ 24h, 15% ≥ 480h

hardly employed. This deceptive method was studied by Tian et al. [241] and [223]. Tian et al. [241] searched proactively for phishing domains by crafting squatting domains and checking for their existence in DNS records. Half of the registered domains were live during the crawling window, and 3% of domains redirected users to domain marketplaces. Verification revealed that just 0.2% were phishing pages. However, 91.5% of these phishing pages remained undetected for at least a month, which suggests that they are more challenging to detect. A similar approach was followed in [223] by combining company registrations with `.com` zone files. 95% of the studied companies had at least one potential visual impersonation domain (VIDN), yet only 7% had at least one registered misspelling during the ten-year analysis period. Historical *WHOIS* records allowed for the clustering of VIDNs and showed that only a handful of companies register VIDNs defensively.

EXECUTION

Half of the examined works have studied the execution of a phishing attack. Analyzing publicly available page logging on phishing websites by Moore et al. [167] estimated that when a phishing website was removed within one day of reporting, the average number of visitors disclosing their credentials was 18, with eight more for each day thereafter. Thirteen years later, an acceleration of this process was reported by Oest et al. [183], who found that most visits take place in the nine hours between the first victim visit and detection. During these nine hours, phishing websites lure in 62% of their victims. In the 2007 analysis by Moore et al. [167], half of the responses entered were fake, and many visits to the landing page of a phishing website were observed. This was quantified by Han et al. [98], who analyzed the visitors of the websites installed in honeypots. Many visits originated from security scanners. Just 9% of all real visitors (security scanners excluded) disclosed any credentials. Crafted credentials were fed into 150 live phishing websites in an experiment by Peng et al. [192] to examine their progression. Only seven leaked accounts received logins quickly after disclosure. Some accounts received multiple attempts from different IP addresses, probably the result of credentials being disclosed to multiple attackers through backdoors in the phishing kit. The longitudinal study of the underground ecosystem fueling credential theft by Thomas et al. [240] was in collaboration with Google. 3,785 credential leak dumps were gathered by monitoring various online sources, which were checked against Google's user base. Over two million

vulnerable Google users could be tied back to deployed phishing kits, 25% of them with a matching password. Examining the login geolocation of Gmail accounts that were involved revealed that 42% of them were last accessed in Nigeria. Oest et al. [183] showed that the most prevalent geolocations coincide with countries disproportionately associated with cybercrime. This is in contrast to Han et al. [98], who compared the geolocation of victim IP addresses with the target population of phishing kits and found that many received most of their victims from a single country. This was later confirmed by Bijmans et al. [15] by examining phishing kit installation times and their manuals.

The increased use of URL shorteners also allowed for a new method of delivering phishing URLs. In 2008, their abuse was noticed by McGrath et al. [158], albeit not very large – only 217 cases. More than ten years later, just 31 short URLs were discovered in the dataset of Peng et al. [192]. Many more were found by Le Page et al. [140] who used URL shortening services to compare the life cycle of phishing and malware attacks. Analysis of `bit.ly` short URLs showed that phishing short URLs have a high click-through rate but a short uptime and are most active 4 hours before being reported as malicious.

MONETIZATION

Just one study examined the monetization of phished credentials. Oest et al. [183] found that 7% of real visitors with an active account at the targeted organization suffered a fraudulent transaction, on average, five days after being phished. Additionally, 63% of the compromised accounts would later appear in a public dump, on average, almost a week later, suggesting that criminals first monetize phished credentials themselves before selling them off.

PHISHING TAKEAWAYS

From Table 6.4, we learn that the research focus has shifted towards the left – from execution and deployment towards the development component in the value chain. Yet, only one study, Oest et al. [183], yields insights into the monetization of phishing attacks. As a result, insights into how criminals make a profit from phishing attacks remain mostly anecdotal. Over time, more active measurements involving a crawler have been deployed, in contrast to earlier work that focused more on passive (domain) datasets. Consequently, due to changing attacker TTP, analysis of live phishing Web pages is increasingly included in research over time. The increased use of HTTPS by phishers has made their pages more believable [183], at the same time allowing for new possibilities to detect them through Certificate Transparency logs by defenders [15, 126]. We observe a similar pattern for the increased use of cloaking in phishing kits, which researchers have exploited by designing techniques to use it for detection [126, 261]. A closer look at the used data sources reveals the prevalent use of some data sources, such as PhishTank or APWG eCX, which could introduce biases [41]. For example, attacks mimicking PayPal are extensively studied, which raises the question of generalizing these results [98, 192]. Only two studies scoped their measurement to a company [183] or a country [15], while multiple studies suggested the localized nature of phishing [15, 98]. Lastly, just as with booter service measurements, extensive scanning from the security industry [98, 183] hampers robust measurements.

6.5. RATs MEASUREMENTS

A Remote Access Trojan (RAT) is a type of malware that allows an attacker to take over a victim's computer [74]. Typically, this includes access to audio and video interfaces, as well as logging of mouse movement and keyboard input. RATs require manual interaction from an attacker and are not designed to execute and exfiltrate automatically, unlike traditional malware [258]. This makes RATs a preferred choice for targeted attacks [73, 74]. Monetary value is created through victim extortion or reselling initial access. A typical value chain for such an attack is depicted in Figure 6.3, in which we identify four components. In *development*, cybercriminals search for developers and software to achieve their remote access. The required infrastructure to successfully operate the software is set up during *deployment*. Next, in the *execution*, an attacker delivers a *stub*, which can be controlled using a *control panel* or *controller*, hosted at a domain and server under the attacker's control. Lastly, profits are made through the acquired access in the *monetization* component. Although RATs have been around since 1999 [258], they have not been extensively studied. We could identify five large-scale technical measurements published between 2017 and 2022, which we list in Table 6.6 and further detail in the following paragraphs.

DATASETS

VirusTotal is the starting point for most RAT research. Through the years, consecutive studies collected more samples (stubs) for their analyses. First, Farinholt et al. [74] examined 19k samples in 2017, then Rezaeirad et al. [201] with 27k samples in 2018, and Farinholt et al. [73] used 146k samples in 2020. Only two studies did not use VirusTotal as a source for measurements. Significant effort was spent by Yang et al. [258] to search underground forums for RATs manually. Faulkenberry et al. [75] combined malware domains from the GT Malware Passive DNS feed with authoritative DNS data. Three studies employed Internet scanning to discover RAT artifacts [73, 74, 201] and two [74, 201] designed honeypots to study interactions. Three studies [73, 74, 201] focused predominantly on two RAT families, namely *DarkComet* or *njRAT*.

DEVELOPMENT

An overview of RAT characteristics by Yang et al. [258] comprises static and dynamic analysis on 53 RATs. A stub was generated for each RAT, and it was found that high-level programming languages (such as C# and VB.NET) are the most popular, as they require only a few or no runtime dependencies. 90% of these RAT stubs targeted solely Windows computers. Additionally, analysis of both the stubs and controller panels revealed that over 80% of the RATs were able to log keystrokes, set up a remote shell, download and execute files, and enable the camera. Oftentimes, these functionalities were implemented similarly across different RATs. Both Farinholt et al. [74] and Farinholt et al. [73] did not study the development of RATs, but through their analysis of the deployment and execution of DarkComet, they did discover several facts about its development. Analysis by Farinholt et al. [74] revealed that DarkComet stubs contain a campaign ID to manage infections, a password to encrypt communications, and a list of controller IP addresses. Follow-up work by Farinholt et al. [73] detailed how to discover DarkComet controllers and download their victim databases.

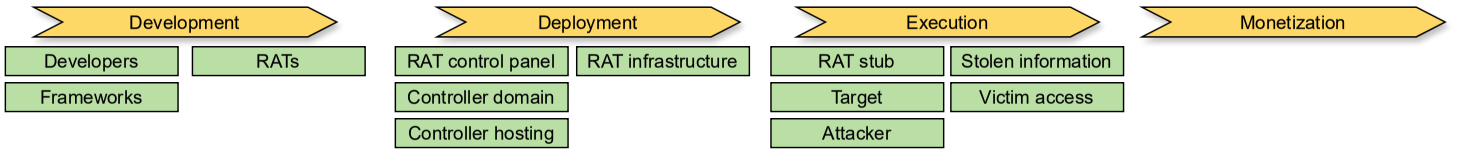


Figure 6.3: RAT value chain with its components at the top and the required resources at the bottom.

Table 6.6: Overview of large-scale RAT measurements. Measurements can be active (*A*), passive (*P*), or both (hybrid, *H*).

Authors	Year	Value chain				Data sources							Misc.				Time frame
		Development	Deployment	Execution	Monetization	VirusTotal	DNS	Internet scans	Hacker forums	RAT databases	Honeypots	Total samples	Measurement	DarkComet	njRAT	Other RATs	
Farinholt et al. [74]	2017		●	●		●	●	●			●	19k	*H*	✓			2016
Rezaeirad et al. [201]	2018		●	●		●		●			●	27k	*H*	✓	✓		2016 – 2017
Farinholt et al. [73]	2020		●	●		●		●		●		146k	*A*	✓			2016 – 2019
Faulkenberry et al. [75]	2022		●				●					245k	*P*	✓	✓	✓	2017 – 2021
Yang et al. [258]	2022	●							●			53	*P*	✓	✓	✓	1999 – 2016

DEPLOYMENT

Different approaches have been employed to gain insights into RAT infrastructure. A live overview of DarkComet was obtained by Farinholt et al. [74] by extracting controller domains and IPs from stubs and through continuous Internet-wide scanning for specific banner responses. 175 online controllers were found at any given time (9,877 in total during an eight-month monitoring period). More controller activity was observed on the weekends compared to the rest of the week. Both Farinholt et al. [74] and Farinholt et al. [73] found Turkey and Russia to be hosting many DarkComet controllers, whereas Rezaeirad et al. [201] reported on the prevalence of North Africa, the Middle East, Brazil, and Russia as hosting locations. Many controller domains use a dynamic DNS (DDNS) service to rapidly change IPs. A user types mapping of these IPs suggests that roughly 90% of controllers are hosted on residential IP networks, likely with limited OPSEC [74]. Two strategies were deployed by Rezaeirad et al. [201] to examine RAT controller operations. Through *RAT-Hole*, a honeypot to mimic a RAT controller, and *RAT-Scan*, to mimic a victim searching for its controller. The latter periodically resolved discovered domains and IPs from sandbox executions and identified 4,584 njRAT and 2,032 DarkComet controller IPs within a seven-month measurement period. The use of DDNS, as discovered by Farinholt et al. [74], was leveraged by Rezaeirad et al. [201] to register 6,897 expired DDNS domains previously used by RAT controllers. RAT-Hole revealed that the majority of traffic towards the honeypot originated from scanners and sandbox executions – a limited number of connections arose from victims. Analysis of downloaded DarkComet configurations by Farinholt et al. [73] discovered 3,518 IP addresses used by 1,162 RAT controllers in a 213-day measurement period. The average uptime of such controllers was 484 days, with some being functional for over three years. A global infrastructure of 399K IPs in 151 countries spread over 202 malware families during a four-year measurement period was observed by Faulkenberry et al. [75]. Various RATs were observed in use in 2022, although DarkComet and njRAT were the most prevalent. Additionally, much interest from scanners soon after a domain is listed as malicious was observed, inflating infection population counts if not properly filtered for.

EXECUTION – ATTACKERS

Monitoring attacker behavior in a controlled environment (e.g., a sandbox) reveals common attacker actions on a target machine. Executing 1,165 DarkComet samples in a honeypot learned that operators commonly access the webcam (61%), steal stored passwords (43%), or explore the victim's file system (40%) [74]. An average session lasted four minutes, 45% of the sessions were motivated by access to a human user – e.g., for harassment or extortion – and at least 58% of RAT operators were motivated by access to user credentials. Analysis by Rezaeirad et al. [201] showed that 43% of controller domains received only a single victim, 90% received at most 20 victims, and just 5% received over 40 victims. This suggests that some attackers widely distribute their malware, whereas others operate more targeted. A similar disparity was reported by Farinholt et al. [73], who found a median of two victims per controller, with ten outliers amassing over a thousand victims each. Lastly, Farinholt et al. [73] found that only 16% of operators use a VPN to hide their tracks.

EXECUTION – VICTIMS

Two studies examined RAT victims. Rezaeirad et al. [201] by letting victims connect to expired DDNS domains and Farinholt et al. [73] by scrutinizing downloaded controller databases. Victim connections found by Rezaeirad et al. [201] showed that most victim IPs are static, and more than half of them have a webcam, making them susceptible to extortion via camera recordings. Infections last long, as 90 days after the controller domain expiration and registration by the researchers, 40% of domains were still receiving victim connections. Almost every country was home to RAT victims, with Brazil being the most prevalent. Correlating the controller and victim locations revealed that they are often located in the same country. This finding was later confirmed by Farinholt et al. [73], who found that more than 74% of attackers with limited victims are located in the same country as most of their victims. Leveraging downloaded databases by Farinholt et al. [73] revealed a total of 57,805 victims in a five-year measurement period. Several steps were taken to validate this number and to allow for comparison with the work of Rezaeirad et al. [201], yielding an overestimation of the number of victims by 40%. Lastly, DarkComet was found to have collected 79,142 keystrokes and recorded 60 hours of activity over 9.6 days for each victim on average.

RATS TAKEAWAYS

Academic measurements focus predominantly on the deployment and execution of RATs, as shown in Table 6.6. The capabilities of stubs, attacker actions, and infrastructure have been well studied, but how RATs end up at victims or how infections are monetized remains unknown. The prevalent use of VirusTotal as a source for RAT stubs stands out and could introduce a measurement bias, as its database mostly depends on user submissions. This leads to the question of whether the stubs uploaded to VirusTotal are, in fact, a representative sample. Additionally, the discovered RAT characteristics are primarily based on two RATs (DarkComet or njRAT), whereas a great variety of RATs are listed by Faulkenberry et al. [75]. To what extent the findings generalize to other types of RATs is unknown. Attacker analysis distinguished two types of attackers: the ones operating on a large scale and targeted attackers operating locally [73, 201], which allows for more in-depth analyses of differences in TTP. Finally, just as with booters and phishing, scanners deployed by the security community make it hard to establish robust observations of victim traffic [75, 201].

6.6. MEASUREMENT CHARACTERISTICS

Before we report on the results of our workshop with LEAs to elicit their assessment of various studies, we summarize the findings from our survey. When scholars measure cybercrime, they measure mostly its deployment and execution, not the monetization component in the value chain, as illustrated by Tables 6.2, 6.4, and 6.6. Many measurements leverage the same datasets or methods, like the use of AmpPot in booter measurements [117, 132, 181], PhishTank in phishing measurements [49, 52, 58, 126, 140, 158, 167, 192, 241], and VirusTotal in RAT measurements [73, 74, 201]. The prevalent use of such datasets could introduce measurement biases. For example, a community-vetted database like PhishTank does not contain every phishing URL but does include all known PayPal phishing URLs due to their collaboration [41]. Which, consequently, results in an

overestimation of PayPal's popularity as a target among phishers. And, although differentiation in temporal and geographical characteristics is mentioned in measurements of every type of cybercrime in our research [15, 73, 181], just two of the 38 studies had their focus adjusted to a specific geographical area or company. Internet-scale measurements provide a nice panoramic view of the global cybercrime landscape. Yet, studies have found phishers operating in one geographical or language area – often the same as theirs [15], gamers launching attacks at their neighboring rivals through booter services [181], and RAT operators targeting victims locally [201]. Such localities can easily get lost in an analysis on an Internet scale, where a geographical focus is absent. Therefore, the takeaways of such analyses represent an average across countries, not within a country, making them less actionable for LEAs operating within local jurisdictions. From the attacker analyses across the examined cybercrimes, we can differentiate two types of actors: cybercriminal groups performing large-scale untargeted attacks – e.g., originating from countries known for cybercrime [183, 240] – and individuals performing small-scale attacks locally (e.g., many RAT operators with few victims [201]). A clear differentiation between these attackers is essential in measurements. Just as geographical diversity, including both groups in one measurement, can result in senseless averages – as these groups rely on very distinct TTP. Lastly, multiple studies across the three cybercrimes mention the amount of scanning by other researchers that hamper robust measurements [132, 183, 201], which should be accounted for properly.

6.7. MEASUREMENTS FOR LAW ENFORCEMENT

As detailed in previous sections, scholars have spent significant effort measuring cybercrime. However, little is known about what kind of measurements (i.e., data, methods, and analyses) best cater to LEA needs. Some papers specifically mention how their measurements could assist LEAs in policing cybercrime [73, 167], but does law enforcement agree? To answer our second research question, we engage with LEA professionals in the Netherlands during a workshop with two objectives. First, we want to elicit their assessment of the added value of specific measurements to see which characteristics align with LEA needs. Second, we wanted their insights into what innovative measurements can cater to LEA needs. Dutch LEAs have been involved in several high-profile cybercrime interventions, such as the takedown of QakBot [243], Webstresser [67], and Bestmixer [69]. Such experiences make them a good partner for this study.

6.7.1. WORKSHOP APPROACH

We invited a diverse group of seven LEA professionals consisting of four participants from the Dutch National Police (ranging from analysts to project managers – all in dedicated cybercrime units), two from the Dutch Public Prosecution Service tasked with cybercrime cases, and one from the cybercrime unit of the Dutch Fiscal Information and Investigation Service (FIOD). Their experience was evenly spread and ranged from one to 15+ years. All participants gave verbal informed consent to participate in our study under the condition that their names or any other PII would not be used in any publication. They also agreed to record and transcribe the workshop. To collect their perspectives on what makes a study more or less aligned with their needs, we present them with

specific studies rather than abstract overviews of a large set of studies. Exposing them to all 38 studies would not be feasible; thus, we selected three studies with different characteristics in terms of data, methods, and findings for each of the three crime types – nine studies in total. In the workshop, we presented 100-word summaries of each study without mentioning their title and authors. Each summary detailed the data source(s), methods, and main findings. Participants were asked to rate each study on three criteria, using seven-point Likert scales. First, the participants were asked about *understanding the phenomenon*: To what extent does this paper add to your understanding of the measured cybercrime? Second, about the *connection to police work*: To what extent does this study connect to your daily work as a law enforcement professional? And third, about the *actionable perspectives*: To what extent do elements of this study offer clear and actionable perspectives? Additionally, we asked participants to reflect on the study in a five-minute discussion. This open-ended approach allowed for a reflexive process of inductive reasoning to discover the characteristics of studies that align with LEA needs. One researcher distilled such characteristics from the transcriptions and audio recordings, and the other researcher who attended the workshop agreed with its findings. In the second part of the workshop, we asked our participants how to advance cybercrime measurements by bringing new measurement ideas forward in a brainstorming session. To allow for fresh ideas, we structured this around three different cybercrimes than the ones already discussed, namely *ransomware*, *online stolen data markets* (e.g., data from phishing or data breaches), and *bulletproof hosting* (hosting providers that do not comply with LEA inquiries when hosting malicious content). For each of these crimes, the participants were asked to write down ideas for new measurements on Post-its. These Post-its were collected per cybercrime on a large sheet of paper. After reading all the ideas, every participant was given 10 points per cybercrime, which they were asked to divide. The highest-scoring ideas were discussed to capture the rationale behind them.

6.7.2. WORKSHOP RESULTS

The next section first details the results of the workshop, which involved previous scientific measurements, followed by the results of the second part of the workshop, which revolved around innovative measurements that can cater to LEA needs.

BOOTERS

As shown in Table 6.7, all studies were rated moderately well on contributing to a better understanding of the phenomenon. The work of Santanna et al. [211] was met with skepticism, and questions were raised regarding the relevancy of this work, as it was published eight years ago. Additionally, the finding that a third of all purchased attacks at booters are not executed was deemed not interesting as "criminals scamming criminals is not our priority", according to Participant 2. The responses to the measurements presented by Kopp et al. [131] were more positive. Ironically, multiple participants agreed with Participant 3, who stated: "It is great to see how little effect these police interventions have. This should be taken into account when designing other interventions". Participant 1 added that "taking down booters is just a game of whack-a-mole, it's better to aim higher". However, this person also added that "police actions are also normative, especially with types of crime that are difficult to control", indicating that such actions send

Table 6.7: Scores assigned by LEA participants, ranging from 1 (totally not agreed) to 7 (totally agreed) ($N = 7$).

	Reference	*Understanding*	*Connection*	*Action*
Booters	Santanna et al. [211]	4.0	2.2	2.8
	Brunt et al. [26]	4.3	3.8	3.9
	Kopp et al. [131]	4.8	4.0	3.8
Phish.	Peng et al. [192]	4.0	2.5	3.1
	Oest et al. [183]	5.0	3.3	3.1
	Bijmans et al. [15]	5.0	4.3	4.1
RATs	Farinholt et al. [74]	4.9	3.3	3.3
	Rezaeirad et al. [201]	4.1	3.3	3.0
	Farinholt et al. [73]	4.7	4.3	4.9

a message to other criminals even though direct results are limited. This study inspired Participant 5 to state that "it would be nice to collaborate with academics whenever we act. They could measure before and after our actions to analyze the effects." Insights into monetization of booters by Brunt et al. [26] gave our participants concrete and actionable perspectives. The popularity of a certain payment service provider can spark future cooperation, especially when such a provider is willing to work with law enforcement. According to Participant 4, this shows that public-private partnerships to police cybercrime are effective, as "profits diminished despite customers changing payment methods" according to Kopp et al. [131].

PHISHING

All presented studies were rated moderately well on contributing to a better understanding of the phenomenon but differed in being well-connected to police work and whether or not they offered an action perspective, as shown in Table 6.7. This discrepancy is well illustrated by the discussion that followed upon the life cycle measurements by Oest et al. [183]. This paper illustrates the short life cycle of phishing URLs, being online for only 21 hours on average. Participant 1 from the Dutch police said: "It is shocking to see these short time frames, as we can never act in this time frame, it's simply too short." The measurements of Oest et al. [183] and Bijmans et al. [15] strengthened some participants in their beliefs that phishing can not be stopped through criminal investigations alone, but also by taking preventive actions or through public-private partnerships. Various participants appreciated the local focus in Bijmans et al. [15] because its findings directly apply to their work. Additionally, the "insights into what phishing kits are popular could steer our investigations towards the developers of the most popular kits.", according to Participant 2. The methods and results in the work of Peng et al. [192] did not align well with police practice, according to our participants. Purposely leaking credentials and observing their evolution did not contribute to anything LEAs can act upon. Despite an idea for a public-private partnership ("if PayPal is really that prevalent, we should cooperate with them", participant 5), it offered no useful insights for criminal investigations.

RATs

All presented studies featured analyses of either one or two different RATs, which made Participant 2 raise the question: "there are hundreds of different RATs active at the moment, did the researchers know how prevalent this RAT was?". Besides this question, all papers were rated moderately well regarding both phenomenon understanding, as shown in Table 6.7, yet the work of Farinholt et al. [73] scored significantly better on the actionable perspective. The geographic distributions of both attackers and defenders reported by Farinholt et al. [73] were well-received: "It is always interesting to see the numbers. Such an overview shows that victim notification is possible" (Participant 5), referring to the victim counts per country according to Farinholt et al. [73]. Additionally, the long RAT controller uptime finding showed that "there is input for criminal investigations, because of the long uptimes. In the 400 days that some RATs remain active, we can easily do a full investigation!" (Participant 3). Participant 4 from the FIOD noted that "we often think about cybercrime on a global scale, this paper shows that victim and perpetrator are much closer to each other, which makes it more worthwhile to investigate", referring to the fact that prosecuting a Dutch perpetrator is much easier than a foreign one. Additionally, the analysis of perpetrator OPSEC gave relevant hints for LEA action. The low use of VPNs by RAT controllers could be exploited for identification. The findings of Rezaeirad et al. [201] showed that dependencies of malicious actors on legitimate services could be exploited for law enforcement investigations or in public-private partnerships. Regarding the popularity of one VPN provider, Participant 1 stated that "the use of commercial products, like these VPN services, makes them ideal targets for public-private partnerships."

6.7.3. ADVANCING MEASUREMENTS

For ransomware, participants wanted to learn about money laundering within the ecosystem. Since the victim payments are transacted in cryptocurrencies, what do criminals do next to convert their assets to fiat currencies? The second most-voted measurement was related to victims, emphasizing both geographical and sectoral analysis. Does ransomware tend to target specific organizations, or is it mostly opportunistic, capitalizing on initial access? Such insights could assist in designing preventive measures. Exact profit calculations or the means of initial access were less popular. As data breaches and stolen credentials enable many different forms of cybercrime [239], our participants expressed interest in learning more about the data types. Is every piece of data equally valuable, or are some more interesting – and more expensive – than others? And, to what extent does it influence the price of stolen data? The total volume of leaked credentials offered in the underground economy was less interesting to the participants. To learn which hosting providers can be considered bulletproof, our participants question the ratio of malicious and benign content that would classify a provider as bulletproof. Additionally, they are curious to know how such providers advertise in the underground economy. Customers are interesting as well, and our participants expressed the desire to understand both their background and payment methods.

To summarize, the LEA participants valued scientific measurements, as Participant 5 stated: "we fail in keeping a close eye on what science does, we should hire more researchers that could introduce these scientific insights into our daily work." The high scores related to the *"understanding the phenomenon"* column in Table 6.7 illustrate this. However, the lower scores in the other two columns emphasize that although scientific measurements can help to better understand cybercrime, they can not directly assist in combating it, as they do not offer concrete action perspectives. The few higher-scoring studies generated actionable perspectives related to geographical differentiation [15, 73], development [15], monetization [26] or the effects of interventions [26, 131]. Measurements of solely the deployment or execution components were less valued. Accurate numbers related to their jurisdiction allow LEAs to act, monetization insights allow for public-private partnerships with, for instance, payment service providers, whereas development insights could steer criminal investigations toward the developers of malicious software instead of their many clients downstream. Findings originating from these scientific measurements could be leveraged to set up public-private partnerships with companies (ab)used by criminals. Brainstorming on innovative measurements highlighted the wish to gain predominantly insights into the monetization of cybercrime, illustrated by the desire to know more about money laundering in ransomware, stolen data price mechanics, and bulletproof hosting advertising.

6.8. DISCUSSION

In this section, we discuss the inherent limitations of our work and mark interesting avenues for future work.

LIMITATIONS

We identify three limitations in our research that could have influenced our findings: our selection of cybercrimes, paper collection, and the recruitment of workshop participants. First, we selected only three cybercrimes to survey the academic field. As mentioned in § 6.2, we selected booter services, phishing, and RATs based on three years of Europol reporting [68, 70, 71]. We generalize our conclusions for all cybercrime research based on our survey of studies covering only these three cybercrimes, which could introduce a bias. Given the large differences in value chains across these cybercrimes, we, however, don't think our conclusions would be vastly different if we had selected other cybercrimes. Second, although we took a systematic approach to survey past cybercrime measurements, as elaborated upon in § 6.2, it is possible that we have missed or incorrectly discarded studies as part of our selection process. Lastly, the workshop with law enforcement professionals included only a limited number of participants. Although these participants came from a variety of agencies and had different roles, the group could have been biased.

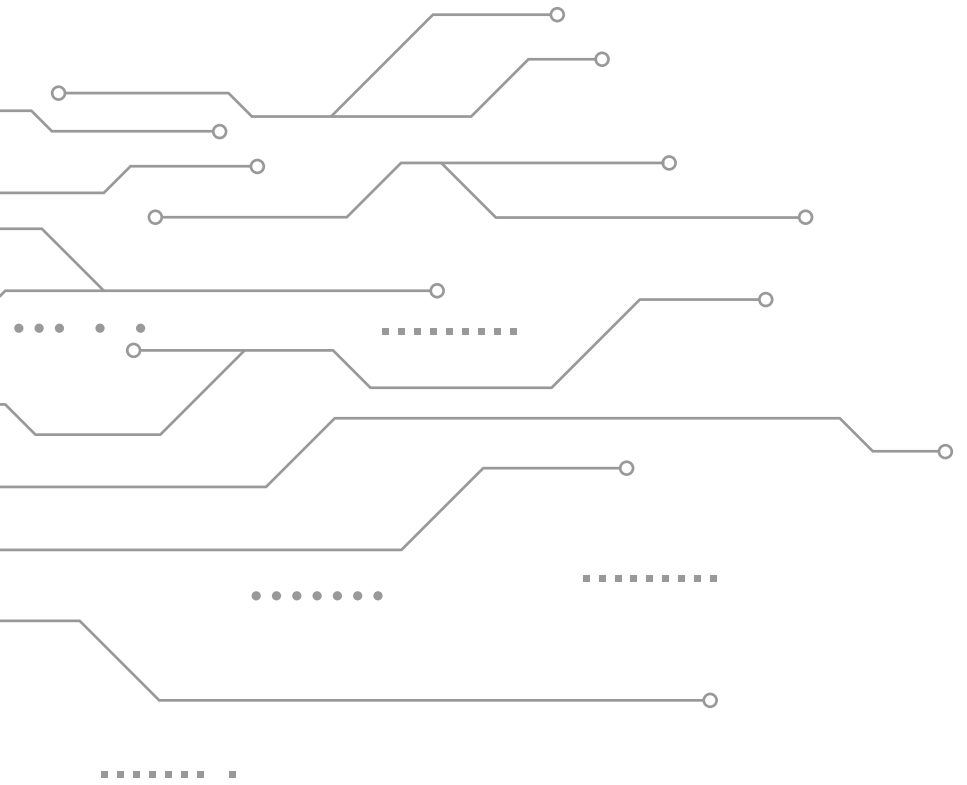
FUTURE WORK

We envision three avenues for future work from the results of this combined survey and user study. First, we encourage researchers to perform more large-scale measurement studies with a sole focus on cybercrime measurements instead of creating detection

methodologies. Since one of the pillars of science is to build on the shoulders of previous researchers, we promote research that takes already published detection methodology from earlier work to perform robust measurements of cybercrime. Second, in performing such measurements, we argue that well-demarcated studies in terms of jurisdiction, geography, or language could aid in better connecting scientific research to police operations. We acknowledge that such research requires good relations with local law enforcement – which can be difficult to achieve – and that such research is not appropriate at every university, nor approved by every institutional research board. Finally, comparative studies between the aforementioned demarcations have the potential to demonstrate differences in actor TTP and in the law enforcement efforts to police them. Such future work would help improve our understanding of cybercrime across the globe and discover what approaches work to combat cybercrime.

6.9. CONCLUSIONS

Combining our survey of large-scale technical measurements and the workshop with LEA professionals allows us to answer our research questions and find the nexus between cybercrime science and policing. In short, we find a mismatch between LEA needs and academic measurements. Most measurements focus on the deployment and execution of cybercrime, whereas LEAs desire to learn more about development and monetization. Although the deployment and execution components are paramount to measure, analyses on the source or how profits are made would also be beneficial, especially for LEAs to build an intervention repertoire. They provide clues that could bring investigators closer to the people facilitating these types of cybercrime, allowing for proactive and disruptive infrastructural policing with longer-lasting effects [48]. That is, monetization insights allow for public-private partnerships with, for instance, payment service providers, whereas development insights could steer criminal investigations toward the developers of malicious software instead of their many clients downstream. As noted by Cui et al. [52], taking down individual phishing websites is far less efficient than policing phishing kit developers, which is what LEAs – given their scarce resources – should focus on. Law enforcement doesn't prioritize investigations into hundreds of individual phishing pages, but this changes as soon as all these websites can be associated with one attacker [167]. To arrive at this conclusion, one needs large-scale technical measurements, which academics could design. Such measurements need a geographical focus, which is essential to LEAs as they operate in conjunction with local jurisdictions. Also, LEA interventions lean on attacker differentiation, as policing individuals requires a different enforcement repertoire than organized cybercriminal groups. Aligning academic and LEA opportunities, considering all value chain components, actor differentiation, and geographical diversity, augments the nexus between large-scale technical measurements and cybercrime policing.



7

CONCLUSION

This dissertation studied large-scale socio-technical Internet measurements of profit-driven cybercrime through various measurement lenses like value chains, life cycles, and campaign analysis. A total of five peer-reviewed studies have been presented in Chapters 2 to 5 and were all aimed at answering the following research question:

How to perform socio-technical measurements of cybercrime to assist in cybercrime governance?

In this final chapter, we summarize the empirical findings from Chapters 2-5 and combine it with the work in Chapter 6 to answer our main research question. Next, we consider the impact of our work and its implications for cybercrime governance. Finally, we propose future research directions that leverage the findings in this dissertation.

7.1. EMPIRICAL FINDINGS

Chapters 2 to 5 presented the results derived from empirical studies involving socio-technical measurements of cybercrime. The main takeaways from these studies are summarized in the following paragraphs and structured around the research gaps identified in § 1.2.

INNOVATIONS IN INTERNET MEASUREMENTS FOR EMERGING CYBERCRIMES

In Chapter 2, we estimated the prevalence of (organized) cryptojacking on websites. Utilizing previous detection techniques developed by Konoth et al. [127], we performed two large-scale measurements of cryptojacking focused on the prevalence of (organized) illicit cryptomining on domains. First, we analyzed the 1.7M most popular domains listed in three top lists to identify organized campaigns. We found 204 campaigns, from which we conclude that the size of organized cryptojacking in large campaigns is heavily underestimated by previous academic research and that criminals have chosen third-party software – such as WordPress – as their attack vector for spreading cryptojacking infections efficiently. In the second measurement, after crawling a random sample

of 49M domains (~20% of the Internet at the time), we conclude that cryptojacking is present on 0.011% of all domains and that adult content is the most prevalent category of websites affected. Looking at the different TLDs, we find that Russian, Brazilian, and Spanish zones are home to a disproportionate number of cryptojacking domains. We show that measurements through top lists like the Alexa Top 1M produce significant overestimates of the problem in terms of prevalence, which we found to be almost six times lower. The discontinuation of Coinhive in March 2019 and the diminishing value of Monero in 2019 – 2021 stressed the importance of monitoring organized cryptojacking campaigns, as cyber criminals will find new ways to spread their cryptojacking infections to remain profitable.

The third chapter examined organized cryptojacking on compromised Internet infrastructure and investigated how cybercriminals set up large-scale campaigns. Through a firmware vulnerability in MikroTik routers, criminals installed illicit cryptomining scripts on these devices, allowing them to rewrite outgoing user traffic and embed cryptomining code in every outgoing Web connection – a MITM attack. Using operator NetFlows, semiweekly Internet crawls, and network telescope back-scatter, we scrutinized cybercriminal activity over a period of 10 months. We found a total of 1.4M routers to be infected at any moment in time, approximately 70% of all MikroTik devices deployed worldwide, ranging from individual installations to campaigns involving thousands of routers. The most affected geographic locations were Brazil and Indonesia, where such devices responded at 29% and 35% of all publicly accessible IP addresses of the largest operators, respectively. Our results show that cryptojacking through MITM attacks is highly lucrative, estimated to exceed $1,200,000 per month in revenue for the ten top-grossing actors. Curiously, we find that innovation and the first-mover advantage do not pay off in terms of revenue made. The highest-grossing actors are not the ones creating new monetization options, deploying sophisticated infrastructure, or creating the largest deployment, but those finding the most productive niche where they can operate relatively undisturbed.

JURISDICTION-FOCUSED INTERNET MEASUREMENTS OF CYBERCRIME

In this chapter, we studied the role of phishing kits within the Dutch phishing ecosystem by leveraging the use of TLS certificates by phishers and the activity of cybercriminals in public Telegram groups. Through a collection of phishing kits shared on this platform and found in open server directories, we constructed fingerprints to measure their prevalence on websites in the wild. Crawling the domains referenced in newly issued TLS certificates enabled us to capture the end-to-end life cycle of these phishing domains – from domain registration and kit deployment to take-down. With this novel method, we identified 1,363 Dutch phishing domains that deployed these phishing kits in a four-month time period, with, on average, 31 phishing domains online every day, waiting for victims to arrive. Most of these domains are online for only 24 hours, but half of them (much) longer. Our analysis of the deployed phishing kits revealed that only a few different kits are in use and that many phishers were building their campaigns on the same framework, *uAdmin*. We discovered that phishers increased their luring capabilities by using multi-stage decoy pages to trick victims into disclosing their credentials.

Relating our findings to data from the APWG showed that our methodology has the potential to detect phishing websites swiftly and that it covers a complementary spectrum of phishing domains.

HOW COMPANIES HANDLE ABUSE TO FIGHT CYBERCRIME

In the fourth study in Chapter 5, we performed an empirical analysis on ground-truth data from a hosting provider to study the abuse handling within such a company in a way never seen before. Through a unique collaboration with law enforcement in the Netherlands, we were granted access to the operational back-end of a hosting provider with a reputation for abuse, which allowed us to investigate what mechanisms in the anti-abuse ecosystem influenced its anti-abuse actions. By gathering and analyzing 1.3M abuse reports and 9,227 unique client notifications, we find significant differences between notification rates among senders and abuse categories. We find that notification rates highly depend on the source and the type of abuse report and less on the involved client. Governance instruments like blocklisting, de-peering, and law enforcement inquiries that could directly hurt business continuity affect client notifications, whereas individual abuse reporting is easily ignored. We find a mismatch between the severity of certain abuse types and their corresponding anti-abuse governance instruments. For instance, spam is considered less harmful than DDoS attacks. Yet, the spam ecosystem is well monitored by Spamhaus, whereas such a governance instrument lacks for DDoS attacks – that relies more on individual reports (by victims). Through longitudinal analysis, we find that government interventions do affect abuse follow-up rates, as CSAM-related abuse reports were better taken care of when governmental pressure started to rise.

7.2. ACADEMIC MEASUREMENTS FOR LAW ENFORCEMENT

The empirical findings emerging from the studies detailed in the previous section convey novel insights into cybercrime measurement innovations. This section outlines our reflections on how socio-technical cybercrime measurements can assist law enforcement.

First, we find that an important and often overlooked aspect of Internet measurements examining cybercrime is that such measurements should aim to measure *crime*. Crime involves individuals (or groups) with a clear (profit-driven) intent and corresponding decision-making processes that balance risk and reward. Such characteristics of crime are well known to criminologists and well incorporated in their research, but are often not well reflected in the measurements performed by computer scientists. This might result from Internet measurements capturing mostly external signals – such as the use of Web-based resources – which can undoubtedly be used to measure parts of cybercriminal operations, yet internal (ground-truth) data is quintessential to assess the criminal intent of a cybercriminal actor.

Additionally, we find that Internet measurements of cybercrime often lack a clearly defined geographical demarcation, which is important to LEAs as they operate in conjunction with local jurisdictions. Although the Internet is a worldwide network that does not respect international borders, its users do live in local communities, and so do cybercriminals [254]. This aspect is overlooked in many large-scale Internet measurements stemming from computer science. Yet, they do matter in performing socio-technical

measurements studying cybercrime, as we have observed phishing criminals lure in victims within the same linguistic region (Chapter 4), routers being infected with cryptojacking malware concentrated in certain locations as a result of local ISP preferences (Chapter 3), and hosting companies operating just within the boundaries of local legislation (Chapter 5). Moreover, our survey of previous work in Chapter 6 has shown that many studies are driven by their access to datasets rather than their wishes to investigate cybercriminal TTP in depth. Selecting the correct dataset to measure cybercrime prevalence remains difficult. As shown in Chapter 2, measurements of the prevalence of cybercrime (in this case, cryptojacking) can differ significantly between the one million most popular domains according to Alexa [3] and a random sample of domains gathered from DNS zone files. To assess to what extent Internet users are suffering from a type of cybercrime – which is, ultimately, the goal of such a measurement – through a top-list measurement is thus unsuitable. Analyzing a top list per country or examining a random sample would result in findings that are much closer to reality.

Next, we find that all measurement lenses introduced in § 1.1.2 can assist in designing measurements that apply directly to law enforcement needs, albeit in different ways. In writing this dissertation, and especially in Chapter 6, it has become apparent that value chains are a useful lens to determine what measurements exactly aim to measure. Having a clear idea of which component in the value chain is the subject of measurement and considering how easily a resource can be outsourced or replaced prevents jumping to incorrect conclusions. Value chain analysis can uncover choke points by identifying scarce resources (e.g., interventions aimed at those resources would incur the most effect) and provide a source for campaign analysis where overlapping resources can be clustered and attributed to the same actor (group). An example of this is the use of sitekeys in the cryptojacking investigations in Chapter 2 and 3, but not the binaries used to perform these illicit cryptojacking operations. Similarly, phishing kits cannot be attributed to the same actor since there is a lively underground market that allows multiple actors to use the same phishing kit for their criminal endeavors. Life cycle analysis has the capability to show the (im)possibilities for governance interventions. For example, the found short uptime for phishing domains in Chapter 4 informs law enforcement that the deployment component in the value chain – where domains play a vital role – might not be the best place to intervene. In contrast, the long uptime of RAT controller domains delineates the possibilities for such interventions within the RAT value chain.

Performing campaign analysis has the potential to provide valuable insights into the degree of organization within a studied cybercrime, as we demonstrated in Chapters 2 and 3. Insights into the degree of organization and the different types of cybercriminals can assist law enforcement in designing effective governance strategies. A lack of actor differentiation in a measurement that includes both low-skilled actors (scripts-kiddies) and high-skilled actors (criminal groups or even nation-state actors) generates results that reflect neither of those two kinds of actors. Many measurements, however, lack attacker differentiation, which is what LEAs lean on to design different policing strategies for individuals than for organized cybercriminal groups. When a cybercrime originates from many individuals, a completely different governance strategy is required as opposed to a large cybercriminal group that is responsible for the majority of cybercrime.

Lastly, we find a misalignment between the focus of academic cybercrime measure-

ments and law enforcement needs, which we discovered through the study presented in Chapter 6. Here, we found most measurements to focus on the deployment and execution components of cybercrime, whereas LEAs desire to learn more about development and monetization. Although the deployment and execution components are paramount to measure, analyses on the source (i.e., the developer) or how profits are made are essential for LEAs to build an intervention repertoire.

In short, cybercrime measurement approaches that consider the intent behind criminal acts, that are clearly delineated in terms of geographical scope and the type of actor studied, and that incorporate campaign analysis to assess not only the prevalence but also the degree of organization, such measurements can provide valuable support to law enforcement efforts in combating cybercrime. Structuring academic analyses around value chains, life cycles, and campaign analysis creates safeguards for researchers to assess whether they measure what they want to measure.

7.3. SOCIETAL IMPACT

Besides the scientific impact of our findings highlighted in Chapters 2 to 6, the work in this dissertation also made a societal impact, although sometimes in an indirect manner. The following paragraphs highlight the impact of our studies outside academia.

AFTERMATH OF COINHIVE-BASED CRYPTOJACKING

Chapters 2 and 3 discussed cryptojacking on websites and compromised infrastructure, and the corresponding papers were published right after a spike in Monero (XMR) prices. Such high XMR prices fuel browser-based cryptojacking attacks. Coinhive, considered the most dominant provider offering browser-based cryptomining scripts, ceased operations in March 2019. Over a year later, in May 2020, Troy Hunt – the operator of `haveibeenpwned.com` – obtained the domains previously owned by Coinhive [105]. While writing a blog post about it, he stumbled upon the two papers on cryptojacking in this dissertation and reached out. We answered some of his questions, which helped him write this blog post. In this analysis, he found many websites still requesting non-existent resources related to Coinhive, which he started logging. This underlined our findings as he agreed that most of the websites requesting such resources were infected through the third-party software attack vector instead of being initiated by the website owner. To urge website owners to remove the defunct references to Coinhive from their website, he made every request to the Coinhive domains to return a modal popping up in the browser window, explaining that the visited website tried to perform illicit cryptomining. The blog was well-received, and we are proud that such a renowned cybersecurity figure referred to both our papers and conference talks.

Most of the 1.7M infected MikroTik routers we studied in Chapter 2 were already cleaned up during the course of our research – see Figure 3.10. By the end of our measurement period, thousands of routers were still serving cryptomining scripts. In an Interpol-led action against cryptojacking on compromised routers in South East Asia, collaborating law enforcement agencies cleaned up 15,000 of such routers in June 2019 [111]. Although this number sounds like many, our research has found that this is only a tiny fraction, as we saw over 1.7M routers infected with cryptojacking scripts

in our measurement period. Most of these routers were cleaned up months before the Interpol operation started. Despite these cleaning efforts, there are still hundreds of routers that have (partially defunct) cryptomining scripts included in their HTTP responses, as we found in Censys as of October 2024.

PHISHING MEASUREMENTS ASSISTING LAW ENFORCEMENT

Geographically demarcated measurements of cybercrime can assist law enforcement in their work to police cybercrime, as we have argued throughout this concluding chapter. Besides the scientific work highlighted in this dissertation, we have put this argument into practice by actively collaborating with Dutch law enforcement from 2019 to 2024. The measurement of phishing in the Netherlands, detailed in Chapter 4, has been conducted at TNO in a multi-year knowledge-building program in which TNO and the Dutch National Police collaborate to improve policing in the digital era. The software created to perform this research was further developed to assist police officers in their daily work, resulting in the application now known as "BigPhish". This application identified and examined phishing domains similarly to the methods described in our research and allowed users to perform in-depth searches and observe trends in phishing kit usage over time. In July 2021, Dutch law enforcement arrested the developer of one of the most used phishing kits at the time [196], which BigPhish had been tracking for several months. To assess the spread of this phishing kit throughout the phishing ecosystem in the Netherlands, police filed a claim to obtain all 400 domains where this kit was identified. The disclosed information was utilized in court to illustrate the scale of its operations. In doing so, we confirmed our hypothesis that scientific measurements can really assist in law enforcement operations. BigPhish continued to have an effect on cybercrime policing outside the Netherlands. A public prosecutor from Bavaria, Germany, approached us to start a pilot with BigPhish in Germany. In close collaboration with German public prosecutors and police officers, BigPhish was adjusted to operate within a different country and tested for several months. Ultimately, this resulted in the signing of a memorandum of understanding between TNO, CFLW, and the Bavarian Ministry of Justice, which agreed to fight phishing collaboratively [13].

The Belgian police took matters into their own hands by building a similar system as BigPhish all by themselves, purely based on the methodology described in our paper [15]. This allowed them to automatically send out thousands of notice-and-takedown requests to providers hosting identified phishing websites and to cluster related phishing domains into campaigns, hereby guiding the allocation of their investigative resources to key criminal actors [25]. Their "Phish Nemo" was even awarded a prize for being the most innovative police project in Belgium in December 2024 [38]

7.4. GOVERNANCE IMPLICATIONS

Given the knowledge we obtained from the five academic studies in this dissertation on how to perform socio-technical measurements of cybercrime, we address the implications our work has on governance in the following paragraphs. We structure our thoughts on governance implications around three entities in the cybercrime-fighting ecosystem: governments, intermediaries, and anti-abuse organizations.

GOVERNMENTS

The first entity within the cybercrime-fighting ecosystem we address is the government. Both local and central governments design policies to address cybercrime and combat its prevalence. The work in this dissertation highlights that such policy creation should align with robust (scientific) measurements. We argue that new laws, regulations, or (law enforcement) interventions should be built upon solid statistics and envision the process of policy creation as follows. First, a well-performed socio-technical measurement of a certain type of cybercrime is executed with clear geographical demarcations taken into account. The results from this measurement are then analyzed, and goals are set, together with policymakers, that future governance instruments should achieve. A SMART (specific, measurable, achievable, relevant, and time-bound) formulated goal could, for example, be to lower the amount of phishing domains targeting Dutch citizens within two years or to reduce the amount of CSAM imagery hosted on Dutch servers within one year. Given such a well-formulated goal, policymakers, together with academic researchers, brainstorm on possible governance interventions. Taking the example of phishing, analysis in Chapter 4 pointed out that most phishing originates from only a few phishing kits, which are made by only a few developers. Additionally, the value chain analysis in Chapter 6 showed that playing a game of whack-a-mole with phishing domains is not effective, as this resource can easily be replaced. Therefore, a possible governance intervention would then be to deprioritize investigations into individual phishing domains and focus on finding and prosecuting the few phishing kit developers. Civil servants or law enforcement professionals operationalize these governance ideas afterward into policy, and researchers wait for them to come into force. After a set period of time, researchers redo their measurements and discuss the outcomes with government officials to assess the effect of the implemented governance intervention.

An important addition to this measurement-based governance development process is the observation that governance actions may also be predominantly normative, i.e., aimed at setting out guidelines for what one should do. Normative governance actions express a social norm on cybercrime but do not necessarily have direct effects on it. Such actions can be divided into proactive and reactive normative governance actions. Reactive normative governance strategies would be to indict foreign hackers, as the U.S. Department of Justice did in October 2024 with two Russian nationals who attacked multiple companies to deploy ransomware [245]. Such indictments rarely lead to arrests or reduce ransomware attacks. Yet, they show the public that the government is taking these crimes seriously and demonstrates its capabilities to find the perpetrators. A proactive normative governance strategy is, for example, the use of Google Ads to make DDoS buyers think twice before they purchase their attacks on booter websites [166]. Here, the law enforcement deployed targeted online ad campaigns to raise cybercrime awareness among individuals searching for DDoS-for-hire services. Although the results of this study showed mixed results – i.e., the advertisement campaigns were not considered significantly effective in reducing DDoS attacks in the short term [166] – it could have influenced some criminals not to start their malpractices. To conclude, governments proposing governance interventions should do so based on robust measurements and clearly defined goals, even if the desired outcome is to implement normative actions.

INTERMEDIARIES

Since the Internet is a decentralized network of many different parties, various intermediaries facilitate both the good parts as well as the bad parts of the Web. Throughout this dissertation, we observed the government as just one of the many parties within the ecosystem to fight cybercrime. However, some of those parties are much more capable of intervening due to their unique information position within the ecosystem. Companies operating infrastructure abused by cybercriminals are in an excellent position to measure, cluster, and disrupt cybercriminal operations. However, such companies have little incentive to invest in such activities, as such efforts will more likely cost money than turn a profit. Based on the research presented in this dissertation, we argue that governance strategies that divide the responsibility to address cybercrime between the facilitating companies and law enforcement would improve the status quo in the fight against cybercrime. This kind of governing, characterized by Garland [81] as a *responsibilization strategy*, involves the government seeking to act upon cybercrime by acting indirectly and encouraging actions from non-state organizations. This can be done by stimulating new forms of behavior or by stopping established habits. Assessing whether this responsibility can be enforced through legislation comes afterward. Through the research presented in this dissertation, we identify several parties that could well be involved in such governance. For example, the responsibility to act upon illicit cryptomining can be shared with both the Monero mining pool operators and legitimate businesses offering cryptomining scripts such as Coinhive. Additionally, as we found the third-party software attack vector the most prominent in cryptojacking campaigns in Chapter 3, part of the responsibility to act upon this cybercrime could also be assigned to the maintainers of major CRM systems like WordPress or Drupal. In the case of phishing – or any other kind of cybercrime that relies on hosting malicious content on Web pages – we would identify the hosting providers as key players who have a shared responsibility to act upon this content.

However, current legislation explicitly limits the responsibility of hosting providers for the content on their infrastructure. Either this legislation has to be changed, or other governance means should address this. An example of the latter would be to increase market transparency through public benchmarking. Government-funded, scientifically executed benchmarks of cybercrime-facilitating markets can move companies to change their policies. As we have observed in Chapter 5, publishing a benchmark of hosting providers involved in hosting CSAM does result in a company changing its stance and efforts to remove such content from its network. However, companies that are willingly facilitating cybercrime are insusceptible to such naming-and-shaming practices.

Either way, the effects of a responsibilization governance strategy would be twofold. First, dividing the responsibility to combat cybercrime between law enforcement and facilitators would incur higher transaction costs for cybercriminals, making their cybercrime malpractices less profitable. Although many resources in cybercrime can be acquired from an abundance of suppliers, as soon as a majority of them feel the responsibility to act upon cybercrime themselves, criminal resources are taken down faster. Obviously, this will then drive cybercriminals into suppliers that do not feel this responsibility, which brings us to the second effect of this governance strategy. By ignoring the responsibility to combat cybercrime, a facilitator can immediately be labeled as a criminal

entity. By criminalizing such negligence, law enforcement can focus on those few misbehaving facilitators rather than chasing the bad customers of benign businesses. A governance approach to do this would be to make cybercrime-facilitating companies more responsible for assisting law enforcement. When every company operating within the same business area is responsible for supporting law enforcement in cybercrime-related matters, it will be part of the regular business. Companies that fail to adhere to the required assistance can then be punished by civil law, incurring more negative impacts on company profits. The latter approach, no matter how exactly executed, builds on strong collaboration between law enforcement, government bodies, and private companies. Such public-private collaborations require investment from the public parties specifically, but they are essential to making this happen.

ANTI-ABUSE ORGANIZATIONS

The last entity we discuss is the group of non-state organizations or individuals wanting to fight cybercrime, either driven by profit or altruism. Just as the Internet is a decentralized worldwide operating network, so are its rules and (non-state) guardians. Ever since the beginning of the Internet, most of its rules and the overarching code of conduct have been primarily voluntary. Conforming to an industry standard like RFC2142 for abuse mailboxes [108] or incorporating guidelines set by working groups like the M^{3}AAWG on how to handle abuse complaints [152] are currently by no means enforced. This is in contrast to rules related to the functionality of the Internet. Non-compliance with functional regulations of the Internet results in crippled connections (e.g., protocols like BGP or DNS only work when networks are properly set up), while a lax stance regarding abusive content or weak security policies is tolerated, as we have shown in Chapter 5. This led to a variety of initiatives trying to enforce the latter, ranging from large foundations to individuals. This fragmented source of abuse reporting makes it labor-intensive for abuse departments to take in, verify, and handle these complaints. As we have observed in Chapter 5, follow-up actions are more likely to be taken when a renowned anti-abuse organization reports a standardized and vetted abuse complaint instead of an individual dropping an email. Standardized reporting would accelerate abuse handling, and industry-wide blocklist operators have the ability to sanction non-compliance. At first sight, an initiative like the Digital Services Act (DSA) [65] that institutionalized the use of trusted flaggers for abuse handling can be considered a step in the right direction. Within the DSA, such parties are appointed by national governments and are considered experts at detecting certain types of illegal content online. Notices originating from trusted flaggers must be treated with priority as they are expected to be more accurate than notices submitted by an average user. Although we support the appointment of parties whose reports must be treated with higher priority, we consider the nation-focused approach a potential pitfall for this regulation. Even though we stated that geographically demarcated measurements are pivotal for law enforcement agencies operating in conjunction with local jurisdiction (in Chapter 6), abuse *reporting* should not be done at such a local level. Especially for certain types of cybercrime that are considered illegal in every European country, it would make no sense to install 27 separate reporting entities. Creating one European entity that could later grow into a worldwide reporting agency for abuse would be a much better solution.

7.5. FUTURE WORK

Each chapter in this dissertation has provided suggestions for future research. In this section, we look beyond these individual recommendations and explore other broader research directions that have emerged from our work.

MULTI-DISCIPLINARY RESEARCH OF THE PATHWAYS INTO CYBERCRIME

As has been put forward by this dissertation, especially in § 7.2, we embrace closer collaboration between Internet measurements and criminology. Several scholars are already joining forces from computer science and criminology, yet future research can advance this collaboration by focusing on a part of the criminal landscape that has not been touched upon by Internet measurements: the perpetrators. Although it is common to study (convicted) criminals in criminology, computer scientists have not been involved in such studies. An interesting research would be to combine the insights from criminology, such as an offender study, with the digital traces offenders leave behind online, which can be collected and analyzed by computer scientists. Combining the qualitative analyses from criminologists and the quantitative results from computer scientists would allow for discovering how external signals relate to personal cybercriminal decision-making. A starting point for this would be to study how cybercriminals make their way into cybercrime by discovering their pathways. Using data scraped from online forums, social media, or criminal chat applications, combined with interviews of the studied offenders, would illuminate how these criminals start their careers and where choke points for disruption can be identified.

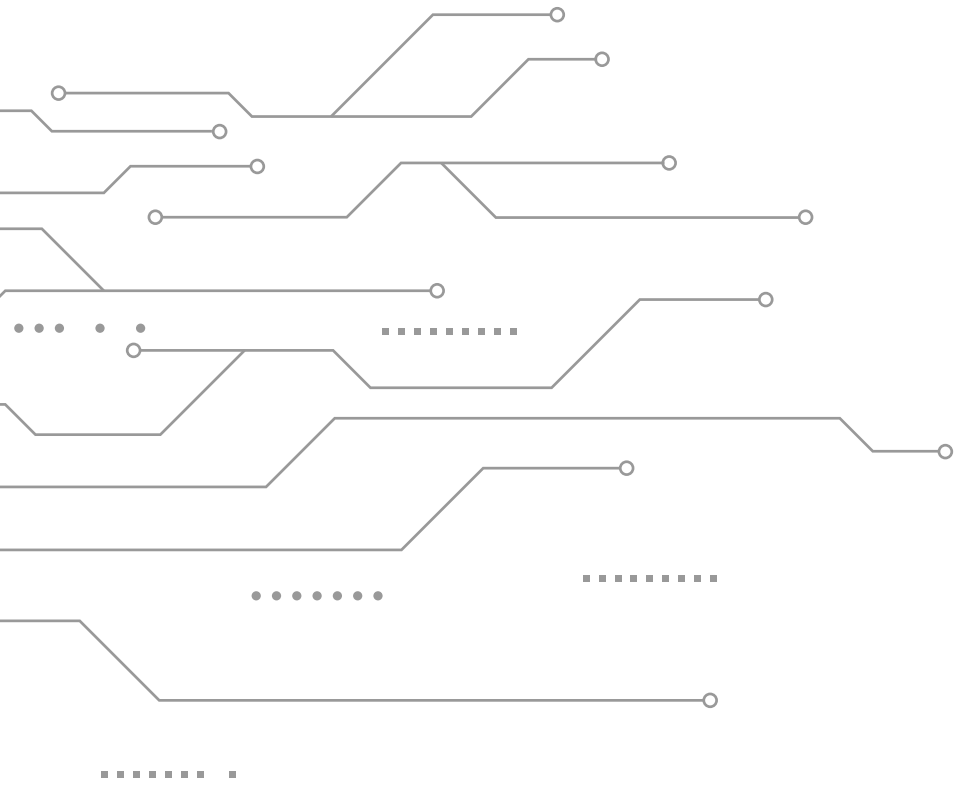
MEASURING INTERVENTION EFFECTS THROUGH COLLABORATIONS

Law enforcement interventions on booter services have been subject to scientific measurements performed together with law enforcement in the past, like the work of Collier et al. on booters [47]. Such research was met with great enthusiasm by the participants in our workshop with LEA detailed in Chapter 6. An avenue for future work would be to perform similar measurements of both other types of cybercrime and involve other kinds of governance interventions, not just those by law enforcement. Close collaboration with the involved governance body is key here. When setting up the measurement, academics should confer with the involved party and together define the key performance indicators that make the proposed intervention successful or not. Then, before the intervention takes place, a baseline must be gathered that is later used to compare to the situation after the intervention. The introduction of the Digital Services Act (DSA) [65] in 2022 would have been a prime target for this. This regulation initiated the use of trusted notifiers, whose notices have to be taken care of in a timely manner. Collaboration between legislators and academics in measuring the effects of these trusted notifiers would identify whether or not abuse notices originating from such trusted flaggers were handled faster as soon as they obtained that status.

LONGITUDINAL CROSS-COUNTRY MEASUREMENTS OF CYBERCRIME

As has been put forward by this dissertation, and especially in Chapter 6, we identified geographical focus in large-scale technical measurements of cybercrime as a key factor

for aligning academic measurements with law enforcement needs. Since LEAs operate in conjunction with local jurisdictions, their view on cybercrime and their options to combat it differ from country to country. To study the differences in cybercriminal TTP as well as the effects of law enforcement actions, a suggestion for future work would be to study cybercrime across countries or legislations. A comparative study between the aforementioned demarcations has the potential to demonstrate differences in actor TTP and in the law enforcement efforts to police them. Such studies are common in criminology, as have been put forward by the works of Moneva & Leukfeldt [166], but are scarce in the field of Internet measurements. Future studies would help improve our understanding of cybercrime across the globe and discover what approaches work to combat cybercrime. Performing such measurements longitudinally – over a longer period of time – would also observe criminals moving from country to country due to the effects of law enforcement or government interventions.



BIBLIOGRAPHY

[1] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules in Large Databases. In *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*. Morgan Kaufmann, 487–499. http://www.vldb.org/conf/1994/P487.PDF

[2] Naci Akdemir, Bülent Sungur, and Bürke Başaranel. 2020. Examining the Challenges of Policing Economic Cybercrime in the UK. *Güvenlik Bilimleri Dergisi* (2020), 113–134.

[3] Alexa. 2018. Top 1M sites. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip

[4] Sumayah Alrwais, Xiaojing Liao, Xianghang Mi, Peng Wang, Xiaofeng Wang, Feng Qian, Raheem Beyah, and Damon McCoy. 2017. Under the Shadow of Sunshine: Understanding and Detecting Bulletproof Hosting on Legitimate Service Provider Networks. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, San Jose, CA, USA, 805–823. https://doi.org/10.1109/SP.2017.32

[5] Ross Anderson, Chris Barton, Rainer Böhme, Richard Clayton, Carlos Gañan, Tom Grasso, Michael Levi, Tyler Moore, Stefan Savage, and Marie Vasek. 2019. Measuring the Changing Cost of Cybercrime. In *The 18th Annual Workshop on the Economics of Information Security (WEIS 2019)*. Boston, Massachusetts.

[6] Ross Anderson, Rainer Böhme, Richard Clayton, and Tyler Moore. 2008. *Security Economics and The Internal Market*. Technical Report. Report to the European Network and Information Security Agency.

[7] Ross Anderson, Rainer Böhme, Richard Clayton, Michel J. G. van Eeten, Michael Levi, Tyler Moore, and Stefan Savage. 2013. Measuring the Cost of Cybercrime. In *The Economics of Information Security and Privacy*. Springer Berlin Heidelberg, Berlin, Heidelberg, 265–300. https://link.springer.com/10.1007/978-3-642-39498-0_12

[8] Anti-Phishing Working Group. 2020. Phishing Activity Trends Report: 2nd Quarter 2020. https://docs.apwg.org/reports/apwg_trends_report_q2_2020.pdf

[9] Anti-Phishing Working Group. 2023. The APWG eCrime Exchange (eCX). https://apwg.org/ecx/

[10] Jart Armin and Cath Everett. 2010. *Top 50 Bad Hosts & Networks 2010 Q1*. Technical Report. Hostexploit. 23 pages. http://hostexploit.com/downloads/top_50_bad_hosts_201003.pdf

[11] Rowland Atkinson and John Flint. 2001. Accessing hidden and hard-to-reach populations: Snowball research strategies. *Social research update* 33, 1 (2001), 1–4.

[12] Banken.nl. 2019. Banken.nl: Marktaandeel. https://www.banken.nl/bankensector/marktaandeel

[13] Bayerisches Staatsministerium der Justiz. 2024. Bayern Setzt Auf "Big-Phish" Im Kampf Gegen Phishing von Sensiblen Bankdaten. https://www.justiz.bayern.de/presse-und-medien/pressemitteilungen/archiv/2024/85.php

[14] Rasika Bhalerao, Maxwell Aliapoulios, Ilia Shumailov, Sadia Afroz, and Damon McCoy. 2019. Mapping the Underground: Supervised Discovery of Cybercrime Supply Chains. In *2019 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, Pittsburgh, PA, USA, 1–16. https://doi.org/10.1109/eCrime47957.2019.9037582

[15] Hugo Bijmans, Tim Booij, Anneke Schwedersky, Aria Nedgabat, and Rolf van Wegberg. 2021. Catching Phishers By Their Bait: Investigating the Dutch Phishing Landscape through Phishing Kit Detection. In *30th USENIX Security Symposium*. USENIX Association, Virtual Event, 3757–3774. https://www.usenix.org/conference/usenixsecurity21/presentation/bijmans

[16] Hugo L. J. Bijmans, Tim M. Booij, and Christian Doerr. 2019. Inadvertently Making Cyber Criminals Rich: A Comprehensive Study of Cryptojacking Campaigns at Internet Scale. In *28th USENIX Security Symposium*. USENIX Association, Santa Clara, CA, USA, 1627–1644. https://www.usenix.org/conference/usenixsecurity19/presentation/bijmans

[17] Hugo L. J. Bijmans, Tim M. Booij, and Christian Doerr. 2019. Just the Tip of the Iceberg: Internet-Scale Exploitation of Routers for Cryptojacking. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM, London, United Kingdom, 449–464. https://doi.org/10.1145/3319535.3354230

[18] Hugo L. J. Bijmans, Michel J. G. van Eeten, and Rolf S. van Wegberg. 2025. A Measured Response – On the Nexus of Large-Scale Technical Measurements and Cybercrime Policing. In *The 24th Workshop on the Economics of Information Security*. Tokyo, Japan. http://kmlabcw.iis.u-tokyo.ac.jp/weis/2025/doc/proceedings/WEIS2025_paper_11.pdf

[19] Leyla Bilge and Tudor Dumitras. 2012. Before we knew it: an empirical study of zero-day attacks in the real world. In *the ACM Conference on Computer and Communications Security, CCS'12*. ACM, Raleigh, NC, USA, 833–844. https://doi.org/10.1145/2382196.2382284

[20] Marzieh Bitaab, Haehyun Cho, Adam Oest, Penghui Zhang, Zhibo Sun, Rana Pourmohamad, Doowon Kim, Tiffany Bao, Ruoyu Wang, Yan Shoshitaishvili, Adam

Doupe, and Gail-Joon Ahn. 2020. Scam Pandemic: How Attackers Exploit Public Fear through Phishing. In *2020 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, Boston, MA, USA, 1–10. https://doi.org/10.1109/eCrime51433.2020.9493260

[21] Norbert Blenn, Vincent Ghiëtte, and Christian Doerr. 2017. Quantifying the Spectrum of Denial-of-Service Attacks through Internet Backscatter. In *Proceedings of the 12th International Conference on Availability, Reliability and Security (ARES)*. ACM, Reggio Calabria, Italy, 21:1–21:10. https://doi.org/10.1145/3098954.3098985

[22] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[23] Aaron Blum, Brad Wardman, Thamar Solorio, and Gary Warner. 2010. Lexical feature based phishing URL detection using online learning. In *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security*. 54–60.

[24] Casey Breen, Cormac Herley, and Elissa M. Redmiles. 2022. A Large-Scale Measurement of Cybercrime Against Individuals. In *CHI '22: CHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA, 29 April 2022 - 5 May 2022*, Simone D. J. Barbosa, Cliff Lampe, Caroline Appert, David A. Shamma, Steven Mark Drucker, Julie R. Williamson, and Koji Yatani (Eds.). ACM, 122:1–122:41. https://doi.org/10.1145/3491102.3517613

[25] Fred Breuls. 2023. Uniek Datalab van de Limburgse Federale Politie Blokkeerde Preventief al 1.000 Phishingwebsites. https://www.vrt.be/vrtnws/nl/2023/10/19/uniek-datalab-van-de-limburgse-federale-politie-blokkeerde-preve/

[26] Ryan Brunt, Prakhar Pandey, and Damon McCoy. 2017. Booted: An Analysis of a Payment Intervention on a DDoS-for-Hire Service. In *Workshop on the Economics of Information Security (WEIS)*. San Diego, 6–26.

[27] Calidog. 2021. Certstream. https://certstream.calidog.io/

[28] Calidog. 2021. Certstream-python. https://github.com/CaliDog/certstream-python

[29] Domhnall Carlin, Philip O'kane, Sakir Sezer, and Jonah Burgess. 2018. Detecting cryptomining using dynamic analysis. In *2018 16th annual conference on privacy, security and trust (PST)*. IEEE, 1–6.

[30] Centraal Bureau voor de Statistiek. 2019. The Netherlands on the European scale: Internet. https://longreads.cbs.nl/european-scale-2019/internet/

[31] Centraal Bureau voor de Statistiek. 2022. Cybersecuritymonitor 2021. https://www.cbs.nl/nl-nl/longread/rapportages/2022/cybersecuritymonitor-2021

[32] Centraal Bureau voor de Statistiek (CBS). 2022. Minder Traditionele Criminaliteit, Meer Online Criminaliteit. https://www.cbs.nl/nl-nl/nieuws/2022/09/minder-traditionele-criminaliteit-meer-online-criminaliteit

[33] Nils Christie. 1997. Four Blocks against Insight: Notes on the Oversocialization of Criminologists. *Theoretical Criminology* 1, 1 (1997), 13–23.

[34] Nilesh Christopher. 2018. Hackers mined a fortune from Indian websites. https://economictimes.indiatimes.com/small-biz/startups/newsbuzz/hackers-mined-a-fortune-from-indian-websites/articleshow/65836088.cms

[35] Justyna Joanna Chromik, Jose Jair Santanna, Anna Sperotto, and Aiko Pras. 2015. Booter Websites Characterization: Towards a List of Threats. In *33rd Brazilian Symposium on Computer Networks and Distributed Systems (SBRC)*. IEEE, Vitória, Brazil, 445–458.

[36] Robert B Cialdini. 1984. *Influence: The new psychology of modern persuasion*. Morrow.

[37] Catalin Cimpanu. 2018. A mysterious grey-hat is patching people's outdated MikroTik routers. https://www.zdnet.com/article/a-mysterious-grey-hat-is-patching-peoples-outdated-mikrotik-routers/

[38] Circle of Police Leadership. 2024. Persbericht: CPL Awards Uitgereikt Voor Opvallend Politiewerk. https://cdn.webdoos.io/circleofpoliceleadership/Persbericht%20NL%20CPL%20Awards%202024_Ine.pdf

[39] Cisco. 2018. Cisco Umbrella 1 Million. http://s3-us-west-1.amazonaws.com/umbrella-static/top-1m.csv.zip

[40] Thomas Claburn. 2018. Crypto-jackers enlist Google Tag Manager to smuggle alt-coin miners. https://www.theregister.co.uk/2017/11/22/cryptojackers_google_tag_manager_coin_hive/

[41] Richard Clayton, Tyler Moore, and Nicolas Christin. 2015. Concentrating Correctly on Cybercrime Concentration. In *14th Annual Workshop on the Economics of Information Security (WEIS)*. Delft, The Netherlands, 16.

[42] Coin-Have. 2018. Coinhave – Monero JavaScript Mining. https://coin-have.com/

[43] Coinhive. 2017. First Week Status Report. https://coinhive.com/blog/en/status-report

[44] Coinhive. 2018. CoinHive - Monero Mining Club. https://coinhive.com/

[45] Coinhive. 2019. Blog: Discontinuation of Coinhive. https://coinhive.com/blog/en/discontinuation-of-coinhive

[46] Coinhive. 2019. Coinhive blog: AuthedMine – Non-Adblocked. https://coinhive.com/blog/en/authedmine

[47] Ben Collier, Daniel R. Thomas, Richard Clayton, and Alice Hutchings. 2019. Booting the Booters: Evaluating the Effects of Police Interventions in the Market for Denial-of-Service Attacks. In *Proceedings of the Internet Measurement Conference*. ACM, Amsterdam Netherlands, 50–64. https://doi.org/10.1145/3355369.3355592

[48] Ben Collier, Daniel R. Thomas, Richard Clayton, Alice Hutchings, and Yi Ting Chua. 2022. Influence, Infrastructure, and Recentering Cybercrime Policing: Evaluating Emerging Approaches to Online Law Enforcement through a Market for Cybercrime Services. *Policing and Society* 32, 1 (Jan. 2022), 103–124. https://doi.org/10.1080/10439463.2021.1883608

[49] Marco Cova, Christopher Kruegel, and Giovanni Vigna. 2008. There Is No Free Phish: An Analysis of "Free" and Live Phishing Kits. In *2nd USENIX Workshop on Offensive Technologies (WOOT)*, Vol. 8. USENIX Association, San Jose, CA, USA, 1–8.

[50] Cryptoloot.com. 2018. CryptoLoot - Earn More From Your Traffic. https://crypto-loot.com/

[51] Alejandro Cuevas, Fieke Miedema, Kyle Soska, Nicolas Christin, and Rolf van Wegberg. 2022. Measurement by Proxy: On the Accuracy of Online Marketplace Measurements. In *Proceedings of the 31st USENIX Security Symposium*. Boston, MA, USA. https://www.usenix.org/system/files/sec22-cuevas.pdf

[52] Qian Cui, Guy-Vincent Jourdan, Gregor V. Bochmann, Russell Couturier, and Iosif-Viorel Onut. 2017. Tracking Phishing Attacks Over Time. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, Perth Australia, 667–676. https://doi.org/10.1145/3038912.3052654

[53] Tiffany Curtiss. 2016. Computer Fraud and Abuse Act Enforcement: Cruel, Unusual, and Due for Reform. *Washington Law Review* 91, 4 (2016), 1813–1850.

[54] Avisha Das, Shahryar Baki, Ayman El Aassal, Rakesh Verma, and Arthur Dunbar. 2020. SoK: A Comprehensive Reexamination of Phishing Research From the Security Perspective. *IEEE Communications Surveys & Tutorials* 22, 1 (2020), 671–708. https://doi.org/10.1109/COMST.2019.2957750

[55] Son Dinh, Taher Azeb, Francis Fortin, Djedjiga Mouheb, and Mourad Debbabi. 2015. Spam Campaign Detection, Analysis, and Investigation. *Digital Investigation* 12 (March 2015), S12–S21. https://doi.org/10.1016/j.diin.2015.01.006

[56] Domaintools.com. 2019. Domain Count Statistics for TLDs. http://research.domaintools.com/statistics/tld-counts/

[57] MML Domenie, ER Leukfeldt, JA Van Wilsem, J Jansen, and W Ph Stol. 2013. *Slachtofferschap in een gedigitaliseerde samenleving: Een onderzoek onder burgers naar e-fraude, hacken en andere veelvoorkomende criminaliteit.* Boom Lemma Uitgevers.

[58] Vincent Drury, Luisa Lux, and Ulrike Meyer. 2022. Dating Phish: An Analysis of the Life Cycles of Phishing Attacks and Campaigns. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*. ACM, Vienna Austria, 1–11. https://doi.org/10.1145/3538969.3538997

[59] Benoit Dupont. 2017. Bots, Cops, and Corporations: On the Limits of Enforcement and the Promise of Polycentric Regulation as a Way to Control Large-Scale Cybercrime. *Crime, Law and Social Change* 67, 1 (Feb. 2017), 97–116. https://doi.org/10.1007/s10611-016-9649-z

[60] Richard Joseph Durbin. 2023. STOP CSAM Act. https://www.congress.gov/bill/118th-congress/senate-bill/1199/text

[61] Zakir Durumeric, David Adrian, Ariana Mirian, Michael D. Bailey, and J. Alex Halderman. 2015. A Search Engine Backed by Internet-Wide Scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS'15)*. ACM, Denver, CO, USA, 542–553. https://doi.org/10.1145/2810103.2813703

[62] Eric Enge. 2019. Mobile vs Desktop Traffic in 2019. https://www.stonetemple.com/mobile-vs-desktop-usage-study/

[63] Shayan Eskandari, Andreas Leoutsarakos, Troy Mursch, and Jeremy Clark. 2018. A First Look at Browser-Based Cryptojacking. In *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, London, 58–66. https://doi.org/10.1109/EuroSPW.2018.00014

[64] European Commission. 2013. *Joint Communication to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions. Cybersecurity Strategy of the European Union: An Open, Safe and Secure Cyberspace.* Technical Report. European Commission: Brussels, Brussels. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=join:JOIN_2013_071

[65] European Parlement & the European Council. 2022. Regulation (EU) 2022/2065 of the European Parliament and of the Council of 19 October 2022 on a Single Market For Digital Services and Amending Directive 2000/31/EC (Digital Services Act). http://data.europa.eu/eli/reg/2022/2065/oj

[66] Europol. 2017. *Internet Organised Crime Threat Assessment (IOCTA) 2017.* Technical Report. Publications Office of the European Union, Luxembourg. https://www.europol.europa.eu/cms/sites/default/files/documents/iocta2017.pdf

[67] Europol. 2018. World's Biggest Marketplace Selling Internet Paralysing DDoS Attacks Taken down. https://www.europol.europa.eu/media-press/newsroom/news/world%E2%80%99s-biggest-marketplace-selling-internet-paralysing-ddos-attacks-taken-down

[68] Europol. 2019. *Internet Organised Crime Threat Assessment (IOCTA) 2019*. Technical Report. Publications Office of the European Union, Luxembourg. https://www.europol.europa.eu/publications-events/main-reports/internet-organised-crime-threat-assessment-iocta-2019

[69] Europol. 2019. Multi-Million Euro Cryptocurrency Laundering Service Bestmixer.Io Taken Down. https://www.europol.europa.eu/media-press/newsroom/news/multi-million-euro-cryptocurrency-laundering-service-bestmixerio-taken-down

[70] Europol. 2021. *Internet Organised Crime Threat Assessment (IOCTA) 2021*. Technical Report. Publications Office of the European Union, Luxembourg. https://www.europol.europa.eu/publications-events/main-reports/internet-organised-crime-threat-assessment-iocta-2021

[71] Europol. 2023. *Internet Organised Crime Threat Assessment (IOCTA) 2023*. Technical Report. Publications Office of the European Union, Luxembourg. https://www.europol.europa.eu/publication-events/main-reports/internet-organised-crime-assessment-iocta-2023

[72] Europol Cybercrime Centre EC3. 2016. Cybercrime dependencies map. https://www.europol.europa.eu/publications-events/publications/cybercrime-dependencies-map

[73] Brown Farinholt, Mohammad Rezaeirad, Damon McCoy, and Kirill Levchenko. 2020. Dark Matter: Uncovering the DarkComet RAT Ecosystem. In *Proceedings of The Web Conference 2020*. ACM, Taipei Taiwan, 2109–2120. https://doi.org/10.1145/3366423.3380277

[74] Brown Farinholt, Mohammad Rezaeirad, Paul Pearce, Hitesh Dharmdasani, Haikuo Yin, Stevens Le Blond, Damon McCoy, and Kirill Levchenko. 2017. To Catch a Ratter: Monitoring the Behavior of Amateur DarkComet RAT Operators in the Wild. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, San Jose, CA, 770–787. https://doi.org/10.1109/SP.2017.48

[75] Aaron Faulkenberry, Athanasios Avgetidis, Zane Ma, Omar Alrawi, Charles Lever, Panagiotis Kintis, Fabian Monrose, Angelos D. Keromytis, and Manos Antonakakis. 2022. View from Above: Exploring the Malware Ecosystem from the Upper DNS Hierarchy. In *Proceedings of the 38th Annual Computer Security Applications Conference*. ACM, Austin, TX, USA, 240–250. https://doi.org/10.1145/3564625.3564646

[76] FBI. 2022. Internet Crime Report 2022. https://www.ic3.gov/Media/PDF/AnnualReport/2022_IC3Report.pdf

[77] Federal Government of the United States. 1986. Computer Fraud and Abuse Act (CFAA).

[78] Jason Franklin, Adrian Perrig, Vern Paxson, and Stefan Savage. 2007. An inquiry into the nature and causes of the wealth of internet miscreants. In *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*. ACM, 375–388. https://doi.org/10.1145/1315245.1315292

[79] Evgeniy Gabrilovich and Alex Gontmakher. 2002. The homograph attack. *Commun. ACM* 45, 2 (2002), 128. https://doi.org/10.1145/503124.503156

[80] Sujata Garera, Niels Provos, Monica Chew, and Aviel D. Rubin. 2007. A Framework for Detection and Measurement of Phishing Attacks. In *Proceedings of the 2007 ACM Workshop on Recurring Malcode*. ACM, Alexandria Virginia USA, 1–8. https://doi.org/10.1145/1314389.1314391

[81] D. Garland. 1996. The Limits of the Sovereign State: Strategies of Crime Control in Contemporary Society. *British Journal of Criminology* 36, 4 (Jan. 1996), 445–471. https://doi.org/10.1093/oxfordjournals.bjc.a014105

[82] Guang-Gang Geng, Xiao-Dong Lee, Wei Wang, and Shian-Shyong Tseng. 2013. Favicon - a clue to phishing sites detection. In *2013 APWG eCrime Researchers Summit*. IEEE, 1–10.

[83] Vincent Ghiette and Christian Doerr. 2018. How Media Reports Trigger Copycats: An Analysis of the Brewing of the Largest Packet Storm to Date. In *ACM SIGCOMM Workshop on Traffic Measurements for Cybersecurity (WTMC)*. 8–13. http://www.cyber-threat-intelligence.com/publications/WTMC-2018-memcached.pdf

[84] GitHub.com. 2018. hoshsadiq/adblock-nocoin-list. https://github.com/hoshsadiq/adblock-nocoin-list

[85] GitHub.com. 2019. leonidackov901/leonidackov901.github.io. https://github.com/leonidackov901/leonidackov901.github.io

[86] Max Goncharov. 2015. *Criminal Hideouts for Lease: Bulletproof Hosting Services*. Technical Report 28. TrendMicro. https://www.trendmicro.com/vinfo/pl/security/news/cybercrime-and-digital-threats/bulletproof-hosting-services-cybercriminal-hideouts-for-lease

[87] Google. 2019. minerBlock. https://chrome.google.com/webstore/detail/minerblock/emikbbbebcdfohonlaifafnoanocnebl?hl=en

[88] Google. 2019. No Coin - Block miners on the web! https://chrome.google.com/webstore/detail/no-coin-block-miners-on-t/gojamcfopckidlocpkbelmpjcgmbgjcl

[89] Google. 2021. Certificate Transparency. https://www.certificate-transparency.org/

[90] Google. 2021. Safe Browsing - Google Safe Browsing. https://safebrowsing.google.com/

[91] Google. 2022. Threat Analysis Group (TAG) - The latest on our efforts to counter government-backed attacks. https://blog.google/threat-analysis-group/

[92] Google. 2023. Certificate Transparency - Working Together to Detect Maliciously or Mistakenly Issued Certificates. https://certificate.transparency.dev/

[93] Google. 2023. Computer Security & Cryptography - Top Publications. https://scholar.google.com/citations?view_op=top_venues&vq=eng_computersecuritycryptography

[94] Google. 2024. About the YouTube Priority Flagger Program. https://support.google.com/youtube/answer/7554338?hl=en

[95] Sarah Gordon and Richard Ford. 2006. On the Definition and Classification of Cybercrime. *Journal in Computer Virology* 2 (08 2006), 13–20. https://doi.org/10.1007/s11416-006-0015-z

[96] Harm Griffioen, Kris Oosthoek, Paul Van Der Knaap, and Christian Doerr. 2021. Scan, Test, Execute: Adversarial Tactics in Amplification DDoS Attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Virtual Event Republic of Korea, 940–954. https://doi.org/10.1145/3460120.3484747

[97] Anti-Phishing Working Group. 2021. APWG - Unifying the global response to cybercrime. https://apwg.org/

[98] Xiao Han, Nizar Kheir, and Davide Balzarotti. 2016. PhishEye: Live Monitoring of Sandboxed Phishing Kits. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Vienna Austria, 1402–1413. https://doi.org/10.1145/2976749.2978330

[99] Crane Hassold. 2017. The Mobile Phishing Threat You'll See Very Soon: URL Padding. https://info.phishlabs.com/blog/the-mobile-phishing-threat-youll-see-very-soon-url-padding

[100] John. J. Hoffman, Steve C. Lee, and Jeffrey S. Jacobson. 2015. New Jersey Division of Consumer Affairs Obtains Settlement with Developer of Bitcoin-Mining Software Found to Have Accessed New Jersey Computers Without Users' Knowledge or Consent. https://nj.gov/oag/newsreleases15/pr20150526b.html

[101] Geng Hong, Zhemin Yang, Sen Yang, Lei Zhang, Yuhong Nan, Zhibo Zhang, Min Yang, Yuan Zhang, Zhiyun Qian, and Haixin Duan. 2018. How You Get Shot in the Back: A Systematical Study about Cryptojacking in the Real World. In

Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. ACM, Toronto Canada, 1701–1713. https://doi.org/10.1145/3243734.3243840

[102] Danny Yuxing Huang, Maxwell Matthaios Aliapoulios, Vector Guo Li, Luca Invernizzi, Elie Bursztein, Kylie McRoberts, Jonathan Levin, Kirill Levchenko, Alex C. Snoeren, and Damon McCoy. 2018. Tracking Ransomware End-to-end. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, San Francisco, CA, 618–631. https://doi.org/10.1109/SP.2018.00047

[103] Danny Yuxing Huang, Hitesh Dharmdasani, Sarah Meiklejohn, Vacha Dave, Chris Grier, Damon McCoy, Stefan Savage, Nicholas Weaver, Alex C Snoeren, and Kirill Levchenko. 2014. Botcoin: Monetizing stolen cycles.. In *NDSS*, Vol. 2014. 1–16.

[104] Keman Huang, Michael Siegel, and Stuart Madnick. 2019. Systematically Understanding the Cyber Attack Business: A Survey. *Comput. Surveys* 51, 4 (July 2019), 1–36. https://doi.org/10.1145/3199674

[105] Troy Hunt. 2021. I Now Own the Coinhive Domain. Here's How I'm Fighting Cryptojacking and Doing Good Things with Content Security Policies. https://www.troyhunt.com/i-now-own-the-coinhive-domain-heres-how-im-fighting-cryptojacking-and-doing-good-things-with-content-security-policies/

[106] Eric M Hutchins, Michael J Cloppert, Rohan M Amin, et al. 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research* 1, 1 (2011), 80.

[107] IETF. 1987. Domain names - implementation and specification. RFC 1035. https://doi.org/10.17487/RFC1035

[108] IETF. 1997. Mailbox Names for Common Services, Roles and Functions. https://www.ietf.org/rfc/rfc2142.txt

[109] Colin C. Ife, Toby Davies, Steven J. Murdoch, and Gianluca Stringhini. 2019. Bridging Information Security and Environmental Criminology Research to Better Mitigate Cybercrime. https://doi.org/10.48550/ARXIV.1910.06380

[110] International Narcotics Control Board. 2022. Materials and Equipment. https://www.incb.org/incb/en/precursors/materials-and-equipment.html

[111] Interpol. 2020. Interpol-Led Action Takes Aim at Cryptojacking in Southeast Asia. https://www.interpol.int/News-and-Events/News/2020/INTERPOL-led-action-takes-aim-at-cryptojacking-in-Southeast-Asia

[112] Luca Invernizzi, Kurt Thomas, Alexandros Kapravelos, Oxana Comanescu, Jean Michel Picod, and Elie Bursztein. 2016. Cloak of Visibility: Detecting When Machines Browse a Different Web. In *IEEE Symposium on Security and Privacy,*

SP 2016, San Jose, CA, USA, May 22-26, 2016. IEEE Computer Society, 743–758. https://doi.org/10.1109/SP.2016.50

[113] ISACA. 2019. *State of Cybersecurity 2019 Part 2: Current Trends in Attacks, Awareness and Governance*. Technical Report. ISACA, Schaumburg, IL, USA. https://www.isaca.org/-/media/files/isacadp/project/isaca/why-isaca/surveys-and-reports/state-of-cybersecurity-2019-part-2_res_eng_0619

[114] Jan Vermeulen. 2019. The Big South African IP Address Heist – How Millions Are Made on the "Grey" Market. https://mybroadband.co.za/news/internet/318205-the-big-south-african-ip-address-heist-how-millions-are-made-on-the-grey-market.html

[115] Franklin Jason, Paxson Vern, Perrig Adrian, and Savage Stefan. 2007. An Inquiry into the Nature and Causes of the Wealth of Internet Miscreants. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*. ACM, Alexandria Virginia USA, 375–388. https://doi.org/10.1145/1315245.1315292

[116] Mohammad Hanif Jhaveri, Orcun Cetin, Carlos Gañán, Tyler Moore, and Michel Van Eeten. 2017. Abuse Reporting and the Fight Against Cybercrime. *Comput. Surveys* 49, 4 (Dec. 2017), 1–27. https://doi.org/10.1145/3003147

[117] Mattijs Jonker, Alistair King, Johannes Krupp, Christian Rossow, Anna Sperotto, and Alberto Dainotti. 2017. Millions of Targets under Attack: A Macroscopic Characterization of the DoS Ecosystem. In *Proceedings of the 2017 Internet Measurement Conference*. ACM, London United Kingdom, 100–113. https://doi.org/10.1145/3131365.3131383

[118] JSEcoin. 2019. JSEcoin: Digital currency - Designed for the Web. https://jsecoin.com/

[119] Lakshmana Rao Kalabarige, Routhu Srinivasa Rao, Ajith Abraham, and Lubna Abdel Kareim Gabralla. 2022. Multilayer Stacked Ensemble Learning Model to Detect Phishing Websites. *IEEE Access* 10 (2022), 79543–79552. https://doi.org/10.1109/ACCESS.2022.3194672

[120] Mohammad Karami and Dammon McCoy. 2013. Rent to Pwn: Analyzing Commodity Booter DDoS Services. *login Usenix Magazine* 38, 6 (2013). https://www.usenix.org/publications/login/december-2013-volume-38-number-6/rent-pwn-analyzing-commodity-booter-ddos

[121] Mohammad Karami, Youngsam Park, and Damon McCoy. 2016. Stress Testing the Booters: Understanding and Undermining the Business of DDoS Services. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, Montréal Québec Canada, 1033–1043. https://doi.org/10.1145/2872427.2883004

[122] Dennis Keil. 2018. WP Monero Miner - Home. https://www.wp-monero-miner.com/

[123] Erin Kenneally and David Dittrich. 2012. The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. *SSRN Electronic Journal* (2012). https://doi.org/10.2139/ssrn.2445102

[124] Amin Kharraz, Zane Ma, Paul Murley, Charles Lever, Joshua Mason, Andrew Miller, Nikita Borisov, Manos Antonakakis, and Michael Bailey. 2019. Outguard: Detecting In-Browser Covert Cryptocurrency Mining in the Wild. In *The World Wide Web Conference on - WWW '19*. ACM Press, San Francisco, CA, USA, 840–852. https://doi.org/10.1145/3308558.3313665

[125] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Rosa Romero Gómez, Nikolaos Pitropakis, Nick Nikiforakis, and Manos Antonakakis. 2017. Hiding in Plain Sight: A Longitudinal Study of Combosquatting Abuse. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*. ACM, 569–586. https://doi.org/10.1145/3133956.3134002

[126] Brian Kondracki, Babak Amin Azad, Oleksii Starov, and Nick Nikiforakis. 2021. Catching Transparent Phish: Analyzing and Detecting MITM Phishing Toolkits. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Virtual Event Republic of Korea, 36–50. https://doi.org/10.1145/3460120.3484765

[127] Radhesh Krishnan Konoth, Emanuele Vineti, Veelasha Moonsamy, Martina Lindorfer, Christopher Kruegel, Herbert Bos, and Giovanni Vigna. 2018. MineSweeper: An In-depth Look into Drive-by Cryptocurrency Mining and Its Defense. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Toronto Canada, 1714–1730. https://doi.org/10.1145/3243734.3243858

[128] Maria Konte, Roberto Perdisci, and Nick Feamster. 2015. ASwatch: An AS Reputation System to Expose Bulletproof Hosting ASes. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. ACM, London United Kingdom, 625–638. https://doi.org/10.1145/2785956.2787494

[129] Daniel Kopp, Christoph Dietzel, and Oliver Hohlfeld. 2021. DDoS Never Dies? An IXP Perspective on DDoS Amplification Attacks. In *Passive and Active Measurement - 22nd International Conference (Lecture Notes in Computer Science)*, Vol. 12671. IEEE, Virtual Event, 284–301. https://doi.org/10.1007/978-3-030-72582-2_17

[130] Daniel Kopp, Eric Strehle, and Oliver Hohlfeld. 2021. CyberBunker 2.0 - A Domain and Traffic Perspective on a Bulletproof Hoster. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Virtual Event Republic of Korea, 2432–2434. https://doi.org/10.1145/3460120.3485352

[131] Daniel Kopp, Matthias Wichtlhuber, Ingmar Poese, Jair Santanna, Oliver Hohlfeld, and Christoph Dietzel. 2019. DDoS Hide & Seek: On the Effectiveness of a Booter Services Takedown. In *Proceedings of the Internet Measurement Conference*. ACM, Amsterdam Netherlands, 65–72. https://doi.org/10.1145/3355369.3355590

[132] Lukas Krämer, Johannes Krupp, Daisuke Makita, Tomomi Nishizoe, Takashi Koide, Katsunari Yoshioka, and Christian Rossow. 2015. AmpPot: Monitoring and Defending Against Amplification DDoS Attacks. In *Research in Attacks, Intrusions, and Defenses (RAID) (Lecture Notes in Computer Science)*, Vol. 9404. Springer, Kyoto Japan, 615–636.

[133] Neal Krawetz. 2021. Looks Like It - The Hacker Factor Blog. http://www.hackerfactor.com/blog/index.php?/archives/432-Looks-Like-It.html

[134] Brian Krebs. 2010. Naming and Shaming 'Bad' ISPs. https://krebsonsecurity.com/2010/03/naming-and-shaming-bad-isps/

[135] Brian Krebs. 2016. The Reincarnation of a Bulletproof Hoster. https://krebsonsecurity.com/2016/08/the-reincarnation-of-a-bulletproof-hoster

[136] Brian Krebs. 2018. Krebs on Security - Who and What is Coinhive. https://krebsonsecurity.com/2018/03/who-and-what-is-coinhive/

[137] Brian Krebs. 2021. Arrest, Raids Tied to 'U-Admin' Phishing Kit. https://krebsonsecurity.com/2021/02/arrest-raids-tied-to-u-admin-phishing-kit/

[138] Mohit Kumar. 2017. This Phishing Attack is Almost Impossible to Detect On Chrome, Firefox and Opera. https://thehackernews.com/2017/04/unicode-Punycode-phishing-attack.html

[139] Robert Layton, Paul A. Watters, and Richard Dazeley. 2010. Automatically determining phishing campaigns using the USCAP methodology. In *2010 eCrime Researchers Summit, eCrime 2010, Dallas, TX, USA, October 18-20, 2010*. IEEE, 1–8. https://doi.org/10.1109/ecrime.2010.5706698

[140] Sophie Le Page, Guy-Vincent Jourdan, Gregor V. Bochmann, Jason Flood, and Iosif-Viorel Onut. 2018. Using URL Shorteners to Compare Phishing and Malware Attacks. In *2018 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, San Diego, CA, 1–13. https://doi.org/10.1109/ECRIME.2018.8376215

[141] Let's Encrypt. 2021. Let's Encrypt - Free SSL/TLS Certificates. https://letsencrypt.org/

[142] E. R. Leukfeldt. 2014. Cybercrime and Social Ties: Phishing in Amsterdam. *Trends in Organized Crime* 17, 4 (Nov. 2014), 231–249. https://doi.org/10.1007/s12117-014-9229-5

[143] Rutger Leukfeldt, Sander Veenstra, and Wouter Stol. 2013. High volume cyber crime and the organization of the police: The results of two empirical studies in the Netherlands. *International Journal of Cyber Criminology* 7, 1 (2013), 1.

[144] K. Levchenko, A. Pitsillidis, N. Chachra, B. Enright, M. Felegyhazi, C. Grier, T. Halvorson, C. Kanich, C. Kreibich, He Liu, D. McCoy, N. Weaver, V. Paxson, G. M. Voelker, and S. Savage. 2011. Click Trajectories: End-to-End Analysis of the Spam Value Chain. In *2011 IEEE Symposium on Security and Privacy*. IEEE, Berkeley, CA, 431–446. https://doi.org/10.1109/SP.2011.24

[145] Yukun Li, Zhenguo Yang, Xu Chen, Huaping Yuan, and Wenyin Liu. 2019. A stacking model using URL and HTML features for phishing webpage detection. *Future Gener. Comput. Syst.* 94 (2019), 27–39. https://doi.org/10.1016/j.future.2018.11.004

[146] Yun Lin, Ruofan Liu, Dinil Mon Divakaran, Jun Yang Ng, Qing Zhou Chan, Yiwen Lu, Yuxuan Si, Fan Zhang, and Jin Song Dong. 2021. Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages. In *30th USENIX Security Symposium (USENIX Security 21)*. 3793–3810.

[147] Pim Lindeman. 2019. Criminelen handelen op berichtenapp: 'Heb je geld? Ik heb een pistool voor 3k'. https://www.ad.nl/dossier-weekend/criminelen-handelen-op-berichtenapp-heb-je-geld-ik-heb-een-pistool-voor-3k~ab66bdd0/

[148] Baojun Liu, Chaoyi Lu, Zhou Li, Ying Liu, Hai-Xin Duan, Shuang Hao, and Zaifeng Zhang. 2018. A Reexamination of Internationalized Domain Names: The Good, the Bad and the Ugly. In *48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2018, Luxembourg City, Luxembourg, June 25-28, 2018*. IEEE Computer Society, 654–665. https://doi.org/10.1109/DSN.2018.00072

[149] Jingqiang Liu, Zihao Zhao, Xiang Cui, Zhi Wang, and Qixu Liu. 2018. A Novel Approach for Detecting Browser-Based Silent Miner. In *Third IEEE International Conference on Data Science in Cyberspace, DSC 2018, Guangzhou, China, June 18-21, 2018*. IEEE, 490–497. https://doi.org/10.1109/DSC.2018.00079

[150] Joeri Loggen and Rutger Leukfeldt. 2022. Unraveling the crime scripts of phishing networks: an analysis of 45 court cases in the Netherlands. *Trends in Organized Crime* (2022), 1–21.

[151] Stephen Lynch. 2020. OpenDNS Unveils 'NLPRank,' a New Model for Advanced Threat Detection. https://umbrella.cisco.com/blog/opendns-unveils-nlprank-a-new-model-for-advanced-threat-detection

[152] M3AAWG. 2015. Anti-Abuse Best Common Practices for Hosting and Cloud Service Providers. https://www.m3aawg.org/sites/default/files/document/M3AAWG_Hosting_Abuse_BCPs-2015-03.pdf

[153] Dhia Mahjoub. 2017. Behaviors and Patterns of Bulletproof and Anonymous Hosting Providers. https://www.usenix.org/conference/enigma2017/conference-program/presentation/mahjoub

[154] Majestic. 2018. Majestic Million CSV now free for all, daily. http://downloads.majestic.com/majestic_million.csv

[155] Samuel Marchal, Kalle Saari, Nidhi Singh, and N. Asokan. 2016. Know Your Phish: Novel Techniques for Detecting Phishing Sites and Their Targets. In *36th IEEE International Conference on Distributed Computing Systems, ICDCS 2016, Nara, Japan, June 27-30, 2016*. IEEE Computer Society, 323–333. https://doi.org/10.1109/ICDCS.2016.10

[156] MaxMind, Inc. 2019. MaxMind GeoIP2. https://www.maxmind.com/en/geoip2-services-and-databases/

[157] Kieren McCarthy. 2018. CBS's Showtime caught mining crypto-coins in viewers' web browsers. https://www.theregister.co.uk/2017/09/25/showtime_hit_with_coinmining_script/

[158] D Kevin McGrath and Minaxi Gupta. 2008. Behind Phishing: An Examination of Phisher Modi Operandi. In *First USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, Vol. 8. USENIX Association, San Francisco, CA, USA, 4.

[159] Xavi Mendez. 2019. Wfuzz - The Web Fuzzer. https://github.com/xmendez/wfuzz/

[160] Meta. 2023. Bringing Local Context to Our Global Standards. https://transparency.meta.com/policies/improving/bringing-local-context/

[161] Simon Migliano. 2018. The Dark Web is Democratizing Cybercrime. https://hackernoon.com/the-dark-web-is-democratizing-cybercrime-75e951e2454

[162] MineCryptoNight. 2019. MineCryptoNight - Making mining profits great again! https://minecryptonight.net/

[163] Mineralt. 2019. Developer API Documentation and Reference. https://support.mineralt.io/support/solutions/articles/36000047274-js-miner-usage-and-api-reference

[164] David Molnar, Serge Egelman, and Nicolas Christin. 2010. This Is Your Data on Drugs: Lessons Computer Security Can Learn from the Drug War. In *Proceedings of the 2010 New Security Paradigms Workshop*. ACM, Concord Massachusetts USA, 143–149. https://doi.org/10.1145/1900546.1900566

[165] Monero Ocean. 2019. Monero ocean – FAQ. https://moneroocean.stream/#/help/faq

[166] Asier Moneva and Rutger Leukfeldt. 2023. The Effect of Online Ad Campaigns on DDoS-attacks: A Cross-national Difference-in-differences Quasi-experiment. *Criminology & Public Policy* 22, 4 (Nov. 2023), 869–894. https://doi.org/10.1111/1745-9133.12649

[167] Tyler Moore and Richard Clayton. 2007. Examining the Impact of Website Take-down on Phishing. In *Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit*, Vol. 269. ACM, Pittsburgh Pennsylvania USA, 1–13. https://doi.org/10.1145/1299015.1299016

[168] Mozilla Foundation. 2014. asm.js - Working Draft — 18 August 2014. http://asmjs.org/spec/latest/

[169] Margi Murphy. 2018. YouTube shuts down hidden cryptojacking adverts. https://www.telegraph.co.uk/technology/2018/01/29/youtube-shuts-hidden-crypto-jacking-adverts/

[170] Troy Mursch. 2018. Cryptojacking malware Coinhive found on 30,000 websites. https://badpackets.net/cryptojacking-malware-coinhive-found-on-30000-websites/

[171] Troy Mursch. 2018. Over 100,000 Drupal websites vulnerable to Drupalgeddon 2 (CVE-2018-7600). https://badpackets.net/over-100000-drupal-websites-vulnerable-to-drupalgeddon-2-cve-2018-7600/

[172] Marius Musch, Christian Wressnegger, Martin Johns, and Konrad Rieck. 2019. Thieves in the Browser: Web-based Cryptojacking in the Wild. In *Proceedings of the 14th International Conference on Availability, Reliability and Security, ARES 2019, Canterbury, UK, August 26-29, 2019*. ACM, 4:1–4:10. https://doi.org/10.1145/3339252.3339261

[173] Satoshi Nakamoto. 2009. Bitcoin: A Peer-to-Peer Electronic Cash System.

[174] Namecheap. 2018. What payment methods do you accept for domain registrations? https://www.namecheap.com/support/knowledgebase/article.aspx/35/7/what-payment-methods-do-you-accept-for-domain-registrations

[175] National CSIRT Cyprus. 2019. MikroTik Routers Compromised in Cryptojacking Campaign. https://csirt.cy/mikrotik-routers-compromised-in-cryptojacking-campaign/

[176] Betaalvereniging Nederland. 2019. Veel meer valse SMS-berichten, zogenaamd van banken. https://www.betaalvereniging.nl/actueel/nieuws/veel-meer-valse-sms-berichten-zogenaamd-van-banken/

[177] NetLab360. 2018. Quick summary about the Port 8291 scan. https://blog.netlab.360.com/quick-summary-port-8291-scan-en/

[178] NIST National Vulnerability Database. 2018. NVD - CVE-2018-14847 Detail. https://nvd.nist.gov/vuln/detail/CVE-2018-14847

[179] Arman Noroozian, Michael Ciere, Maciej Korczynski, and Michel van Eeten. 2017. Inferring the Security Performance of Providers from Noisy and Heterogenous Abuse Datasets. In *16th Workshop on the Economics of Information Security (WEIS)*. San Diego, CA, USA.

[180] Arman Noroozian, Jan Koenders, Eelco van Veldhuizen, Carlos H Ganan, Sumayah Alrwais, Damon McCoy, and Michel van Eeten. 2019. Platforms in Everything: Analyzing Ground-Truth Data on the Anatomy and Economics of Bullet-Proof Hosting. In *28th USENIX Security Symposium*. USENIX Association, Santa Clara, CA, USA, 1341–1356.

[181] Arman Noroozian, Maciej Korczyński, Carlos Hernandez Gañan, Daisuke Makita, Katsunari Yoshioka, and Michel Van Eeten. 2016. Who Gets the Boot? Analyzing Victimization by DDoS-as-a-Service. In *Research in Attacks, Intrusions, and Defenses - 19th International Symposium, RAID 2016 (Lecture Notes in Computer Science)*, Vol. 9854. Springer, Paris, France, 368–389.

[182] Adam Oest, Yeganeh Safei, Adam Doupe, Gail-Joon Ahn, Brad Wardman, and Gary Warner. 2018. Inside a Phisher's Mind: Understanding the Anti-Phishing Ecosystem through Phishing Kit Analysis. In *2018 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, San Diego, CA, 1–12. https://doi.org/10.1109/ECRIME.2018.8376206

[183] Adam Oest, Penghui Zhang, Brad Wardman, Eric Nunes, Jakub Burgis, Ali Zand, Kurt Thomas, Adam Doupé, and Gail-Joon Ahn. 2020. Sunrise to Sunset: Analyzing the End-to-end Life Cycle and Effectiveness of Phishing Attacks at Scale. In *29th USENIX Security Symposium (USENIX Security 2020)*. USENIX Association, Virtual Event, 361–377. https://www.usenix.org/conference/usenixsecurity20/presentation/oest-sunrise

[184] Ufuoma Ogono. 2018. Monero Cryptojacking: Monero Cryptocurrency Mining Malware Disrupts Government Site. https://smartereum.com/35507/monero-cryptojacking-monero-cryptocurrency-mining-malware-disrupts-government-site-monero-news-today/

[185] Openbaar Ministerie. 2020. FIOD Doet Onderzoek Naar Bedrijf Dat 'Bulletproof' Internetdiensten Aanbiedt. https://www.om.nl/actueel/nieuws/2020/09/25/fiod-doet-onderzoek-naar-bedrijf-dat-%E2%80%98bulletproof%E2%80%99-internetdiensten-aanbiedt

[186] OpenPhish. 2023. OpenPhish - Phishing Intelligence. https://openphish.com/

[187] Charlie Osborne. 2018. MikroTik routers enslaved in massive Coinhive cryptojacking campaign. https://www.zdnet.com/article/mikrotik-routers-enslaved-in-massive-coinhive-cryptojacking-campaign/

[188] Palo Alto Networks. 2022. Unit 42 Threat Analysis Group. https://unit42.paloaltonetworks.com/

[189] Panagiotis Papadopoulos, Panagiotis Ilia, and Evangelos P. Markatos. 2018. Truth in Web Mining: Measuring the Profitability and Cost of Cryptominers as a Web Monetization Model. *CoRR* abs/1806.01994 (2018). arXiv:1806.01994 http://arxiv.org/abs/1806.01994

[190] Sergio Pastrana and Guillermo Suarez-Tangil. 2019. A First Look at the Crypto-Mining Malware Ecosystem: A Decade of Unrestricted Wealth. In *Proceedings of the Internet Measurement Conference*. ACM, Amsterdam, The Netherlands, 73–86. https://doi.org/10.1145/3355369.3355576

[191] Jordan Pearson. 2019. Starbucks Wi-Fi Hijacked People's Laptops to Mine Cryptocurrency. https://motherboard.vice.com/en_us/article/gyd5xq/starbucks-wi-fi-hijacked-peoples-laptops-to-mine-cryptocurrency-coinhive

[192] Peng Peng, Chao Xu, Luke Quinn, Hang Hu, Bimal Viswanath, and Gang Wang. 2019. What Happens After You Leak Your Password: Understanding Credential Sharing on Phishing Sites. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*. ACM, Auckland New Zealand, 181–192. https://doi.org/10.1145/3321705.3329818

[193] Kirsty Phillips, Julia C. Davidson, Ruby R. Farr, Christine Burkhardt, Stefano Caneppele, and Mary P. Aiken. 2022. Conceptualizing Cybercrime: Definitions, Typologies and Taxonomies. *Forensic Sciences* 2, 2 (April 2022), 379–398. https://doi.org/10.3390/forensicsci2020028

[194] PhishTank. 2021. PhishTank - Out of the Net, into the tank. https://www.phishtank.com/

[195] PhishTank. 2023. PhishTank - Join the Fight against Phishing. https://phishtank.org/

[196] Politie. 2021. Ontwikkelaar Phishing Software Opgepakt. https://www.politie.nl/nieuws/2021/juli/22/11-ontwikkelaar-en-verkoper-phishing-software-opgepakt.html

[197] The Spamhaus Project. 2020. The Spamhaus Project - The Top 10 Most Abused TLDs. https://www.spamhaus.org/statistics/tlds/

[198] Proxy Lists 24. 2019. Proxy Lists 24 - Daily Free Proxy Server Lists. http://www.proxyserverlist24.top/

[199] Julian Rauchberger, Sebastian Schrittwieser, Tobias Dam, Robert Luh, Damjan Buhov, Gerhard Pötzelsberger, and Hyoungshick Kim. 2018. The Other Side of the Coin: A Framework for Detecting and Analyzing Web-based Cryptocurrency Mining Campaigns. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*. ACM, Hamburg Germany, 1–10. https://doi.org/10.1145/3230833.3230869

[200] Carin MM Reep-van den Bergh and Marianne Junger. 2018. Victims of cybercrime in Europe: a review of victim surveys. *Crime science* 7, 1 (2018), 1–15.

[201] Mohammad Rezaeirad, Brown Farinholt, Hitesh Dharmdasani, Paul Pearce, Kirill Levchenko, and Damon McCoy. 2018. Schrödinger's RAT: Profiling the Stakeholders in the Remote Access Trojan Ecosystem. In *27th USENIX Security Symposium*. USENIX Association, Baltimore, MD, USA, 1043–1060.

[202] Markus Riek and Rainer Böhme. 2018. The Costs of Consumer-Facing Cybercrime: An Empirical Exploration of Measurement Issues and Estimates. *Journal of Cybersecurity* 4, 1 (Jan. 2018). https://doi.org/10.1093/cybsec/tyy004

[203] Rijksoverheid. 2020. Grapperhaus Spreekt ICT-bedrijven Aan Op Kinderporno Op Hun Servers. https://www.rijksoverheid.nl/actueel/nieuws/2020/04/30/grapperhaus-spreekt-ict-bedrijven-aan-op-kinderporno-op-hun-servers

[204] RIPE NCC. 2024. RIPEstat Announced Prefixes. https://stat.ripe.net/docs/02.data-api/announced-prefixes.html

[205] Juan D. Parra Rodriguez and Joachim Posegga. 2018. RAPID: Resource and API-Based Detection Against In-Browser Miners. In *Proceedings of the 34th Annual Computer Security Applications Conference, ACSAC 2018, San Juan, PR, USA, December 03-07, 2018*. ACM, 313–326. https://doi.org/10.1145/3274694.3274735

[206] Robby Roks and Nahom Monshouwer. 2020. F-gamers die 'mapsen','swipen'en 'bonken': een netnografisch onderzoek naar fraude en oplichting op Telegram Messenger. *Justitiele Verkenningen* 46, 2 (2020).

[207] Jan Rüth, Torsten Zimmermann, Konrad Wolsing, and Oliver Hohlfeld. 2018. Digging into Browser-based Crypto Mining. In *Proceedings of the Internet Measurement Conference 2018*. ACM, Boston MA USA, 70–76. https://doi.org/10.1145/3278532.3278539

[208] Muhammad Saad, Aminollah Khormali, and Aziz Mohaisen. 2018. End-to-End Analysis of In-Browser Cryptojacking. *CoRR* abs/1809.02152 (2018). arXiv:1809.02152 http://arxiv.org/abs/1809.02152

[209] Nicolas van Saberhagen. 2013. CryptoNote v 2.0. https://cryptonote.org/whitepaper.pdf

[210] Jose Jair Santanna, Romain Durban, Anna Sperotto, and Aiko Pras. 2015. Inside Booters: An Analysis on Operational Databases. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, Ottawa, ON, Canada, 432–440. https://doi.org/10.1109/INM.2015.7140320

[211] Jose Jair Santanna, Roland van Rijswijk-Deij, Rick Hofstede, Anna Sperotto, Mark Wierbosch, Lisandro Zambenedetti Granville, and Aiko Pras. 2015. Booters – An

Analysis of DDoS-as-a-service Attacks. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, Ottawa, ON, Canada, 243–251. https://doi.org/10.1109/INM.2015.7140298

[212] Emily Schechter. 2021. Evolving Chrome's security indicators. https://blog.chromium.org/2018/05/evolving-chromes-security-indicators.html

[213] Quirin Scheitle, Oliver Hohlfeld, Julien Gamba, Jonas Jelten, Torsten Zimmermann, Stephen D. Strowes, and Narseo Vallina-Rodriguez. 2018. A Long Way to the Top: Significance, Structure, and Stability of Internet Top Lists. In *Proceedings of the Internet Measurement Conference 2018*. ACM, Boston, MA, USA, 478–493. https://doi.org/10.1145/3278532.3278574

[214] Jim AM Schiks, Steve GA van de Weijer, and E Rutger Leukfeldt. 2022. High tech crime, high intellectual crime? Comparing the intellectual capabilities of cybercriminals, traditional criminals and non-criminals. *Computers in Human Behavior* 126 (2022), 106985.

[215] Mathew J. Schwartz. 2018. Cryptojackers Keep Hacking Unpatched MikroTik Routers. https://www.bankinfosecurity.com/cryptominers-keep-hacking-unpatched-mikrotik-routers-a-11627

[216] Jerome Segura. 2018. A look into Drupalgeddon's client-side attacks. https://blog.malwarebytes.com/threat-analysis/2018/05/look-drupalgeddon-client-side-attacks/

[217] Selenium. 2021. WebDriver: Documentation for Selenium. https://www.selenium.dev/documentation/en/webdriver/

[218] Shodan. 2019. Shodan - The search engine for Internet-connected devices. https://www.shodan.io/

[219] Craig A Shue, Andrew J Kalafut, and Minaxi Gupta. 2012. Abnormally Malicious Autonomous Systems and Their Internet Connectivity. *IEEE/ACM Transactions on Networking* 20, 1 (2012), 220–230.

[220] SimilarWeb. 2019. Similar Web. Website Traffic Statistics & Market Intelligence. https://www.similarweb.com

[221] Camelia Simoiu, Ali Zand, Kurt Thomas, and Elie Bursztein. 2020. Who is targeted by email-based phishing and malware?: Measuring factors that differentiate risk. In *IMC '20: ACM Internet Measurement Conference, Virtual Event, USA, October 27-29, 2020*. ACM, 567–576. https://doi.org/10.1145/3419394.3423617

[222] Geoffrey Simpson and Tyler Moore. 2020. Empirical Analysis of Losses from Business-Email Compromise. In *2020 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, Boston, MA, USA, 1–7. https://doi.org/10.1109/eCrime51433.2020.9493250

[223] Geoffrey Simpson, Tyler Moore, and Richard Clayton. 2020. Ten Years of Attacks on Companies Using Visual Impersonation of Domain Names. In *2020 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, Boston, MA, USA, 1–12. https://doi.org/10.1109/eCrime51433.2020.9493251

[224] Gilberto Atondo Siu, Ben Collier, and Alice Hutchings. 2021. Follow the Money: The Relationship between Currency Exchange and Illicit Behaviour in an Underground Forum. In *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, Vienna, Austria, 191–201. https://doi.org/10.1109/EuroSPW54576.2021.00027

[225] Liudmila Sivetc and Mariëlle Wijermars. 2021. The Vulnerabilities of Trusted Notifier-Models in Russia: The Case of Netoscope. *Media and Communication* 9, 4 (2021), 27–38.

[226] Slushpool. 2018. Stratum Mining Protocol. https://slushpool.com/help/topic/stratum-protocol/

[227] Zhanna Malekos Smith, Eugenia Lostri, and James A. Lewis. 2020. The Hidden Costs of Cybercrime. https://www.mcafee.com/enterprise/en-us/assets/reports/rp-hidden-costs-of-cybercrime.pdf

[228] SonicWall. 2018. Massive cryptojacking campaign compromised ̃200,000 MikroTik routers. https://securitynews.sonicwall.com/xmlpost/massive-cryptojacking-campaign/

[229] Spamhaus Project. 2024. Spamhaus - Strengthening Trust and Safety across the Internet. https://www.spamhaus.org/

[230] Brett Stone-Gross, Christopher Kruegel, Kevin Almeroth, Andreas Moser, and Engin Kirda. 2009. FIRE: FInding Rogue nEtworks. In *2009 Annual Computer Security Applications Conference*. IEEE, Honolulu, Hawaii, USA, 231–240. https://doi.org/10.1109/ACSAC.2009.29

[231] Karthika Subramani, William Melicher, Oleksii Starov, Phani Vadrevu, and Roberto Perdisci. 2022. PhishInPatterns: Measuring Elicited User Interactions at Scale on Phishing Websites. In *Proceedings of the 22nd ACM Internet Measurement Conference*. ACM, Nice France, 589–604. https://doi.org/10.1145/3517745.3561467

[232] Samaneh Tajalizadehkhoob, Rainer Böhme, Carlos Gañán, Maciej Korczyński, and Michel Van Eeten. 2018. Rotten Apples or Bad Harvest? What We Are Measuring When We Are Measuring Abuse. *ACM Transactions on Internet Technology* 18, 4 (July 2018), 49:1–49:25. https://doi.org/10.1145/3122985

[233] Choon Lin Tan, Kang-Leng Chiew, KokSheik Wong, and San-Nah Sze. 2016. PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder. *Decis. Support Syst.* 88 (2016), 18–27. https://doi.org/10.1016/j.dss.2016.05.005

[234] Telegram. 2021. Telegram FAQ: So how do you encrypt data? https://telegram.org/faq#q-so-how-do-you-encrypt-data

[235] Tenable. 2018. MikroTik RouterOS Vulnerabilities: There's More to CVE-2018-14847. https://www.tenable.com/blog/mikrotik-routeros-vulnerabilities-there-s-more-to-cve-2018-14847

[236] The Monero Project. 2018. Monero: What is Monero (XMR)? https://www.getmonero.org/get-started/what-is-monero/

[237] The Pirate Bay. 2017. The Pirate Bay-Miner. https://thepiratebay.org/blog/242

[238] Daniel R. Thomas, Richard Clayton, and Alastair R. Beresford. 2017. 1000 Days of UDP Amplification DDoS Attacks. In *2017 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, Scottsdale, AZ, 79–84. https://doi.org/10.1109/ECRIME.2017.7945057

[239] Kurt Thomas, Danny Yuxing Huang, David Wang, Elie Bursztein, Chris Grier, Thomas J Holt, Christopher Kruegel, Damon McCoy, Stefan Savage, and Giovanni Vigna. 2015. Framing Dependencies Introduced by Underground Commoditization. In *14th Annual Workshop on the Economics of Information Security (WEIS)*. Delft, The Netherlands.

[240] Kurt Thomas, Frank Li, Ali Zand, Jacob Barrett, Juri Ranieri, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, Angelika Moscicki, Daniel Margolis, Vern Paxson, and Elie Bursztein. 2017. Data Breaches, Phishing, or Malware?: Understanding the Risks of Stolen Credentials. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Dallas Texas USA, 1421–1434. https://doi.org/10.1145/3133956.3134067

[241] Ke Tian, Steve T. K. Jan, Hang Hu, Danfeng Yao, and Gang Wang. 2018. Needle in a Haystack: Tracking Down Elite Phishing Domains in the Wild. In *Proceedings of the Internet Measurement Conference 2018*. ACM, Boston MA USA, 429–442. https://doi.org/10.1145/3278532.3278569

[242] Ivan Torroledo, Luis David Camacho, and Alejandro Correa Bahnsen. 2018. Hunting Malicious TLS Certificates with Deep Neural Networks. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security, CCS 2018, Toronto, ON, Canada, October 19, 2018*. ACM, 64–73. https://doi.org/10.1145/3270101.3270105

[243] U.S. Attorney's Office. 2023. Qakbot Malware Disrupted in International Cyber Takedown. https://www.justice.gov/usao-cdca/pr/qakbot-malware-disrupted-international-cyber-takedown

[244] U.S. Department of Justice. 2018. Criminal Charges Filed in Los Angeles and Alaska in Conjunction with Seizures Of 15 Websites Offering DDoS-For-Hire Services. https://www.justice.gov/opa/pr/criminal-charges-filed-los-

angeles-and-alaska-conjunction-seizures-15-websites-offering-ddos

[245] U.S. Department of Justice. 2024. Russian National Indicted for Series of Ransomware Attacks. https://www.justice.gov/opa/pr/russian-national-indicted-series-ransomware-attacks

[246] Erik van de Sandt. 2019. *Deviant security: the technical computer security practices of cyber criminals*. Ph.D. Dissertation. University of Bristol, UK. https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.774525

[247] Amber van der Heijden and Luca Allodi. 2019. Cognitive Triaging of Phishing Attacks. In *28th USENIX Security Symposium (USENIX Security 19)*. USENIX Association, Santa Clara, CA, 1309–1326. https://www.usenix.org/conference/usenixsecurity19/presentation/van-der-heijden

[248] Rolf van Wegberg, Samaneh Tajalizadehkhoob, Kyle Soska, Ugur Akyazi, Carlos Ganan, Bram Klievink, Nicolas Christin, and Michel van Eeten. 2018. Plug and Prey? Measuring the Commoditization of Cybercrime via Online Anonymous Markets. In *27th USENIX Security Symposium*. USENIX Association, Baltimore, USA, 1009–1026. https://www.usenix.org/conference/usenixsecurity18/presentation/van-wegberg

[249] Rolf Steven van Wegberg. 2020. *Outsourcing Cybercrime*. Ph.D. Dissertation. Delft University of Technology.

[250] Wenhao Wang, Benjamin Ferrell, Xiaoyang Xu, Kevin W. Hamlen, and Shuang Hao. 2018. SEISMIC: SEcure In-lined Script Monitors for Interrupting Cryptojacks. In *Computer Security*. Vol. 11099. Springer International Publishing, Cham, 122–142. http://link.springer.com/10.1007/978-3-319-98989-1_7

[251] WebAssembly.org. 2018. WebAssembly. https://webassembly.org/

[252] Webshrinker. 2019. Webshrinker APIs. https://www.webshrinker.com/apis/

[253] Websocket.org. 2018. HTML5 WebSocket - A Quantum Leap in Scalability for the Web. http://www.websocket.org/aboutwebsocket.html

[254] Barry Wellman. 2001. Computer Networks As Social Networks. *Science* 293, 5537 (Sept. 2001), 2031–2034. https://doi.org/10.1126/science.1065547

[255] Wordfence.com. 2017. WordPress Plugin Banned for Crypto Mining. https://www.wordfence.com/blog/2017/11/wordpress-plugin-banned-crypto-mining/

[256] x0rz. 2020. phishing_catcher. https://github.com/x0rz/phishing_catcher

[257] Jun (Jim) Xu, Jinliang Fan, Mostafa H. Ammar, and Sue B. Moon. 2001. On the design and performance of prefix-preserving IP traffic trace anonymization. In *Proceedings of the 1st ACM SIGCOMM Internet Measurement Workshop, IMW 2001*.

ACM, San Francisco, California, USA, 263–266. https://doi.org/10.1145/505202.505234

[258] Runqing Yang, Xutong Chen, Haitao Xu, Yueqiang Cheng, Chunlin Xiong, Linqi Ruan, Mohammad Kavousi, Zhenyuan Li, Liheng Xu, and Yan Chen. 2022. RATScope: Recording and Reconstructing Missing RAT Semantic Behaviors for Forensic Analysis on Windows. *IEEE Transactions on Dependable and Secure Computing* 19, 3 (May 2022), 1621–1638. https://doi.org/10.1109/TDSC.2020.3032570

[259] Jing Zhang, Zakir Durumeric, Michael Bailey, Mingyan Liu, and Manish Karir. 2014. On the Mismanagement and Maliciousness of Networks. In *Proceedings 2014 Network and Distributed System Security Symposium*, Vol. 14. Internet Society, San Diego, CA, 23–26. https://doi.org/10.14722/ndss.2014.23057

[260] Penghui Zhang, Adam Oest, Haehyun Cho, Zhibo Sun, Rc Johnson, Brad Wardman, Shaown Sarker, Alexandros Kapravelos, Tiffany Bao, Ruoyu Wang, Yan Shoshitaishvili, Adam Doupe, and Gail-Joon Ahn. 2021. CrawlPhish: Large-scale Analysis of Client-side Cloaking Techniques in Phishing. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, San Francisco, CA, USA, 1109–1124. https://doi.org/10.1109/SP40001.2021.00021

[261] Penghui Zhang, Zhibo Sun, Sukwha Kyung, Hans Walter Behrens, Zion Leonahenahe Basque, Haehyun Cho, Adam Oest, Ruoyu Wang, Tiffany Bao, Yan Shoshitaishvili, Gail-Joon Ahn, and Adam Doupé. 2022. I'm SPARTACUS, No, I'm SPARTACUS: Proactively Protecting Users from Phishing by Intentionally Triggering Cloaking Behavior. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Los Angeles CA USA, 3165–3179. https://doi.org/10.1145/3548606.3559334

[262] Yue Zhang, Jason I. Hong, and Lorrie Faith Cranor. 2007. Cantina: a content-based approach to detecting phishing web sites. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*. ACM, 639–648. https://doi.org/10.1145/1242572.1242659

[263] Tong Zhu, Yan Meng, Haotian Hu, Xiaokuan Zhang, Minhui Xue, and Haojin Zhu. 2021. Dissecting Click Fraud Autonomy in the Wild. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*. ACM, 271–286. https://doi.org/10.1145/3460120.3484546

ACKNOWLEDGEMENTS

The first time somebody offered me a PhD position, I turned down the offer after much consideration. However, pursuing a PhD would always stay in the back of my mind, so when Rolf approached me and offered me a part-time PhD position under his wing, I knew I had to seize the opportunity. Although both the administrative processes as well as some of the research did not go very smoothly from time to time, I am happy I embarked on this academic journey. Throughout it, I have been supported by the presence of brilliant people around me who helped, supported, or joyfully distracted me to finish the booklet that is currently in your hands. In this chapter, I would like to take the opportunity to thank a few of them.

To my co-promotor, former colleague, daily supervisor, and fellow cycling enthusiast, Rolf, thank you for offering me the PhD position to become your very first PhD graduate. I want to thank you for all our wonderful conversations, not just about the work in this dissertation but also about life, cycling, traveling, Italy, food, and coffee. Your guidance throughout this journey has always been outstanding. I really want to express my gratitude for the time you made available to brainstorm, review my work, and throw some last-minute sprinkles over my texts. Your motto of "let me fix stuff so that you can do the research" should be an example to other promotores.

Michel, thank you for believing in me as a researcher from the start, for your guidance throughout my PhD, and for your sharp look at my analyses during every meeting. I am still impressed by your detailed knowledge regarding the research all your PhDs are involved in. Although it's a shame we never got to sing "Watskeburt" together, I enjoyed the many informal moments we had, either during the Thursday drinks, the yearly TPM festival, during our biweeklies, or the occasional talks we had with me standing in your doorway.

Christian, as my first ever supervisor in academia, I want to thank you for teaching me the foundations of writing scientific papers. Your never-ending curiosity, extensive knowledge of cybersecurity and computer networks, and drive to make a good story from scientific research have been inspiring.

Tim, the second author on three of the five studies that are part of this thesis, but above all, a friend with whom I can enjoy my spare time and work exceptionally well. Thank you for your significant contributions to this booklet and for being my paranymph during my defense. I am very pleased I dragged you along to TNO to become my study mate, friend, and colleague for a while. We shared plenty of good memories, like your first sip of coffee, our 'illegal' coding sessions at my place during the COVID-19 lockdowns that always ended with a beer and many snacks, and running and screaming like little girls during an epic escape room.

Lars, thank you for the wonderful conversations we have every time we meet, on topics related to cybersecurity, but also on stock investments, houses, coffee, life, and traveling. I remember sharing my doubts about taking this PhD opportunity and the

good talks we had discussing all the pros and cons of it. I'm grateful for your advice to just do it. Choosing you as one of my paranymphs is my way to say thank you! Although we see each other almost every month – to play another round of Pandemic – I always feel we should meet more often. Let's make that happen.

Harm, thank you for being a fellow PhD and friend to whom I could whine about paper rejections and horrible reviewers, but also one with whom I could share my research ideas and brainstorm while drinking beer and eating lots of meat. I would have loved to have you on my defense committee, but I finished this dissertation too soon. Best of luck combining an academic career with your entrepreneurial endeavors.

My TNO colleagues, and especially TNO's cybercrime team featuring Thijmen, Anneke, Federico, Arne, Jacinta, Ivo, Naomi, Ignas, and the many different project managers we had in the last five years, thanks for supporting me throughout my PhD and the teamwork on awesome projects. Having a team like this that shares a common goal to assist the Dutch police in catching cybercriminals was amazing.

Esther, you deserve to be mentioned here. Thank you for all your efforts to make TNO contribute to my PhD. Although this process did not go without a hitch, you always supported me personally. Our trip to Munich in 2024 was the cherry on the cake for our collaboration.

Karin, my manager at TNO, you have supported my PhD since you became my manager; I am very grateful for that. Your mindset to make things possible rather than spot impossibilities was refreshing and helped me finish my PhD a lot sooner than I would have without your help.

A big thank you to my fellow PhDs in TPM's cybersecurity group – Aksel, Annebel, Arwa, Cécile, Elsa, Evi, Fieke, Gerbrand, Kelvin, Lorenz, Mathew, Max, Radu, Ronak, Sandra, Swaathi, Szu-Chun, Veerle, Xander, Yana – for being such nice peers throughout this academic adventure. I enjoyed our lunch conversations, the yearly Sinterklaas at Michel's house, the occasional Wednesday pita, and, of course, the Thursday drinks, although I know I attended too few of them.

A special thanks to the LEA-only (now cyber-cyber) group. Fieke, for being so enthusiastic, smart, and ambitious. I really hope you finish your PhD soon while making epic career moves within Dutch law enforcement. Kelvin, for fixing my broken VMs and servers and letting me enjoy your interesting food and drink choices. Cécile, for the lovely conversations we had sitting next to each other. Simone, for adding a touch of good-old student life into the often calm PhD days. Joyce, for making every meeting with Rolf in your shared office even more fun.

Alice, I would like to thank you for hosting me at Cambridge University. It was an absolute pleasure working with you and your team. Having a quiet place to focus on writing the final parts of this dissertation was very appreciated. Also, a special thanks to Tina, Konstantinos, Yanna, Mariella, Markus, Anna, Ivy, Jack, and Anh for the warm welcome, the board game evenings, inviting me for dinner at King's, the daily lunches at the West Hub followed by a round of *Timeguessr*, and for making me feel part of your excellent research group.

Daimen, thanks for being present at crucial moments in my life.

To my high-school friends, Bob, Donny, Evelien, Femke, Gijs, Jeroen, Jim, Joris, Marie, Petra, Remco, Rick, and Ruben, thanks for the much-needed distractions, the

yearly (running) dinners, Ardennes trips, and laughs every time we see each other.

To my Papyrii, Auke, Bram, Douwe, Freek, Glenn, Jasper, Jelle, Joep, Lennard, Ludo, Michael, and Robin, thanks for joining me in the adventure often referred to as 'student life'. I'm very grateful we kept this group of wonderful guys together after our studies and still meet regularly. I appreciated your constant questions regarding the status of my PhD. Thank you all for all the fun we had and will have in the future.

Although he is not around us anymore, I want to thank composer Simeon Ten Holt, as significant parts of this thesis have been written while listening to the endless piano variations featured in his masterpiece *Canto Ostinato*.

To my parents, although you have occasionally called one of my papers a 'trash paper' (*flut paper*), I know you didn't mean it like that and have always supported me throughout this PhD journey. I know I've made you proud. Thanks for the continuous support and love.

And last but not least, thank you, Emma, for being by my side during the second half of this project. Because of you, I got to work on my PhD at places I would never have imagined. Your faith in me and the proud look on your face when I finished a paper motivated me to finish this dissertation.

Hugo Bijmans
Leiden, August 2025

AUTHORSHIP CONTRIBUTIONS

This dissertation is founded on five peer-reviewed papers stemming from studies involving several co-authors. I led each study, but I benefited from valuable feedback and varying contributions from my collaborators. The following paragraphs outline the specific contributions of each co-author for every study.

For the first study (Chapter 2), my co-authors, Tim Booij and Christian Doerr, helped with writing the draft, proofreading, and polishing the text. Tim Booij analyzed the WebSocket proxy data featuring the operator NetFlows. Christian Doerr drew the illustration in Figure 2.1 and helped trim down the text to conform to the formatting requirements. I implemented the crawler, performed the data analyses, and did most of the writing.

In the second study (Chapter 3), my co-authors, Tim Booij and Christian Doerr, helped with writing the draft, proofreading, and polishing the text. Christian Doerr drew the illustrations in Figure 3.1 and in Figure 3.9 and created the timeline in Figure 3.2. Again, Tim Booij performed all the analyses involving the operator NetFlows. I performed all other data analyses and did most of the writing.

In the third study (Chapter 4), Tim Booij, Anneke Schwedersky, and I worked together to collect phishing kits from Telegram. Aria Nedgabat was responsible for building the first version of the domain crawler, which Tim Booij and I built upon to create the crawler infrastructure used in this study. Tim Booij performed the phishing kit family analysis and created Figure 4.7. Rolf van Wegberg wrote the public policy takeaways in § 4.10. Both Tim Booij and Rolf van Wegberg helped with writing the draft, proofreading, and polishing the text. I performed data analysis and did most of the writing.

For the fourth study (Chapter 6), my co-authors, Rolf van Wegberg and Michel van Eeten, helped structure the research and assisted in proofreading and polishing the text. Rolf van Wegberg supported me during the workshop, used his network to find suitable participants, and proposed the title of this study. I performed data analysis and did most of the writing.

For the fifth study (Chapter 5), I must first and foremost thank Rolf van Wegberg for obtaining access to the data. I would not have been able to perform this study without his tireless efforts to make such a unique data source available for scientific research. Both Michel van Eeten and Rolf van Wegberg helped me scope the study and assisted with proofreading and polishing the text. I performed all the data analysis and did most of the writing.

LIST OF PUBLICATIONS

7. **Bijmans, H.L.J.**, van Eeten, M.J.G. & van Wegberg, R.S. (2025). "Tickets to Hide: Scrutinizing the Anti-Abuse Ecosystem with Internal Abuse Data". *Under submission at NDSS'26*
6. Bekkers, L., Loggen, J., Keja, N., **Bijmans, H.**, & Leukfeldt, R. (2025). "Exploring cybercrime and traditional crime on Telegram in the Netherlands." *Under submission at Global Crime.*
5. **Bijmans, H.L.J.**, van Eeten, M.J.G. & van Wegberg, R.S. (2025). "A Measured Response – On the Nexus of Large-Scale Technical Measurements and Cybercrime Policing". In *Proceedings of the 24th Workshop on the Economics of Information Security (WEIS '25).*
4. **Bijmans, H.L.J.** & van Leuken, M.S.C. (2024). "No Time to Choose: Leveraging Internet Scans to Determine IoC Lifetimes." In *Proceedings of the 2024 IEEE International Conference on Big Data (BigData).*
3. **Bijmans, H.L.J.**, Booij, T.M., Schwedersky, A., Nedgabat, A. & van Wegberg, R.S. (2021). "Catching Phishers By Their Bait: Investigating the Dutch Phishing Landscape through Phishing Kit Detection". In *Proceedings of the 30th USENIX Security Symposium (USENIX Security '21).*
2. **Bijmans, H.L.J.**, Booij, T.M. & Doerr, C. (2019). "Just the Tip of the Iceberg: Internet-Scale Exploitation of Routers for Cryptojacking". In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS 2019).*
1. **Bijmans, H.L.J.**, Booij, T.M. & Doerr, C. (2019). "Inadvertently Making Cyber Criminals Rich: A Comprehensive Study of Cryptojacking Campaigns at Internet Scale". In *Proceedings of the 28th USENIX Security Symposium (USENIX Security '19).*

DATASETS

Table 7.1: Dataset availability.

Publication	Dataset(s)
Bijmans, H.L.J., Booij, T.M. & Doerr, C. (2019). "Inadvertently Making Cyber Criminals Rich: A Comprehensive Study of Cryptojacking Campaigns at Internet Scale". In *Proceedings of the 28th USENIX Security Symposium (USENIX Security '19)*.	Crawling data and software is publicly available at `https://www.cyber-threat-intelligence.com/cryptojacking-campaigns`. Operator NetFlows can not be shared.
Bijmans, H.L.J., Booij, T.M. & Doerr, C. (2019). "Just the Tip of the Iceberg: Internet-Scale Exploitation of Routers for Cryptojacking". In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS 2019)*.	Data collected from Censys can not be shared due to our agreements with Censys. Operator NetFlows can not be shared.
Bijmans, H.L.J., Booij, T.M., Schwedersky, A., Nedgabat, A. & van Wegberg, R.S. (2021). "Catching Phishers By Their Bait: Investigating the Dutch Phishing Landscape through Phishing Kit Detection". In *Proceedings of the 30th USENIX Security Symposium (USENIX Security '21)*.	Software is publicly available at `https://github.com/COSSAS/bigphish`, the collected dataset can not be shared.
Bijmans, H.L.J., van Eeten, M.J.G. & van Wegberg, R.S. (2025). "A Measured Response – On the Nexus of Large-Scale Technical Measurements and Cybercrime Policing". In *Proceedings of the 24th Workshop on the Economics of Information Security (WEIS '25)*.	Workshop transcripts, even anonymized, can not be shared. All reviewed literature is publicly available.
Bijmans, H.L.J., van Eeten, M.J.G. & van Wegberg, R.S. (2025). "Tickets to Hide: Scrutinizing the Anti-Abuse Ecosystem with Internal Abuse Data". *Under submission at NDSS'26*	Datasets from Dutch law enforcement can not be shared. IP Prefix announcements are publicly available through the RIPE NCC API.

About the author

Hugo Bijmans (1994) was born in Voorburg, the Netherlands. He joined the Delft University of Technology in 2012 to pursue a bachelor's degree in Systems Engineering, Policy Analysis, and Management (*Technische Bestuurskunde*). After completing this, he decided to switch to the field of Computer Science and successfully completed a bridging program. In 2016, he enrolled in the MSc Computer Science and specialized in cybersecurity. During his studies, he interned for four months at KPN's Security Operations Centre (SOC) to gain hands-on experience in cybersecurity. He obtained his MSc degree in 2019 with a thesis on cryptojacking, which was rewarded with a 10 out of 10. After graduation, he joined TNO, where he worked as a researcher on topics like cybercrime, monitoring and detecting cyber threats, and the automation of cybersecurity operations. He worked with various clients within the cybersecurity industry, including governments, financial institutions, and academia, to bring cybersecurity innovation into practice. In 2022, he rejoined Delft University of Technology as a part-time PhD candidate under the supervision of prof. dr. Michel van Eeten and dr. Rolf van Wegberg. During this PhD, Hugo studied socio-technical measurements of cybercrime and how insights gained from these measurements can improve governance. He has been involved in assisting the Cybercrime Science course, supervised one master's student during her final project, and presented his work at several international conferences. In 2024, he spent three weeks as a visiting PhD candidate at the Cambridge Cybercrime Centre in the United Kingdom.