From the Outside In

Predicting internal security incidents with external network data

Vermeer, M.

DOI
10.4233/uuid:23ca5781-44dd-449d-a5c4-d6485e84fc96

Publication date
2024

Document Version
Final published version

Citation (APA)
Vermeer, M. (2024). *From the Outside In: Predicting internal security incidents with external network data*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:23ca5781-44dd-449d-a5c4-d6485e84fc96

FROM THE OUTSIDE IN

PREDICTING INTERNAL SECURITY INCIDENTS WITH EXTERNAL NETWORK DATA

FROM THE OUTSIDE IN

PREDICTING INTERNAL SECURITY INCIDENTS WITH EXTERNAL NETWORK DATA

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus, Prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates
to be defended publicly on
11 July 2024 at 12:30

by

Mathew VERMEER

Master of Science in Computer Science, Delft University of Technology
born in Santiago de Cali, Colombia.

This dissertation has been approved by the promotors:

Dr.ir. C. Hernández Gañán
Prof. dr. M.J.G. van Eeten

Composition of the doctoral committee:

Rector Magnificus	Chairman
Dr.ir. C. Hernández Gañán	Delft University of Technology, promotor
Prof. dr. M.J.G. van Eeten	Delft University of Technology, promotor
Independent members:	
Prof.dr.ir. P.H.A.J.M. van Gelder	Delft University of Technology
Prof.dr.ir. R.M. van Rijswijk-Deij	University of Twente
Prof.dr. G. Smaragdakis	Delft University of Technology
Dr.ir. S. Tajalizadehkhoob	ICANN
Reserve member:	
Prof.dr. M.E. Warnier	Delft University of Technology

This research was partially supported by the Air Force Research Laboratory (AFRL) under agreement number FA8750-19-1-0152, and by the European Commission through the H2020 program in projects CyberSecurity4Europe (Grant No. #830929), and Safe-DEED (Grant No. #825225). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

Keywords: asset discovery, network intrusion detection system, security operations center, NIDS, SOC, security incident prediction

Printed by: Gildeprint

Cover image: Rowdy Smith

ISBN: 978-94-6496-149-2

An electronic version of this dissertation is available at
http://repository.tudelft.nl/

To my mom and dad

CONTENTS

SUMMARY

It goes without saying that the Internet is far from secure. As the number of Internet-connected devices increases, so do the number of cyberattacks we have to deal with. Numerous industry reports reveal significant upswings in software vulnerabilities year after year. These are issues plaguing enterprises of all sizes, within the public and private sector. In light of these findings, it becomes imperative for businesses, regardless of size, to prioritize cybersecurity and re-evaluate their current defense mechanisms against this evolving threat landscape.

The evolving cyber threat landscape has emphasized the importance of adopting proactive approaches to manage and mitigate cybersecurity risks. Organizations can take a great many steps to achieve this, but mainly choose security measures that revolve around compliance requirements and standardized methodologies and frameworks to improve overall security posture. Adapting and integrating these standardized methodologies is not merely a technical endeavor; it also entails considerable monetary and human resources, and time commitments. However, it remains unclear to which extent such investments have their desired effect. This is mainly because security is a latent property that cannot be measured directly, meaning that changes to it through means of security investment cannot be determined in a straightforward manner.

Alternative approaches have recently emerged that aim to measure security in a more direct manner. Instead of relying on self-reported data, internal or otherwise, firms gather externally accessible data such as network or server misconfigurations, as well as data concerning malicious activities within the organization's network (e.g., hosts listed in various abuse blacklists). Subsequently, a classifier is trained using this data, enabling it to predict, with a certain level of accuracy, which organizations are likely to experience (large-scale) breaches.

Still, while there seems to be a correlation between external measurements and breaches, causality between the two has yet to be confirmed by independent research, calling into question the feasibility of their endeavor. Furthermore, it is not clear how metrics derived from purely external measurements compare to the security level derived from internal measurements of an organization's network. This reveals the necessity of taking into account the internal state of networks when observing external security signals, instead of exclusively

relying on externally observable or publicly reported data breaches.

Using exclusively external network data treats organizations as black boxes. There is no way to tell whether these signals say anything about the internal state of the network. Furthermore, the predictions are very coarse-grained. Introducing internal network data from network intrusion detection systems (NIDSs) that is processed and examined by security operations centers (SOCs) into this problem can be used to study the links between external network signals and the internal state. NIDSs and the rulesets they employ are responsible for the detection of potentially malicious activity, and, therefore, serve as windows into the internal state of an organization's network.

This dissertation studies the feasibility of security incident prediction and risk estimation. It examines how external network scan data can be leveraged to infer information about the internal state of security of an organization's network. Thus, we formulate the following research question: *How can internal security incidents be predicted through the leverage of external network signals?* The following chapters present the multiple studies that altogether aim to address different portions of this overarching research question.

Chapter 2 compiles and systematizes novel asset discovery techniques. We presented a framework for asset discovery that proposes a syntax to make explicit the steps in the asset-discovery process and used it to systematize all asset discovery techniques designed or mentioned in literature published in 14 leading academic venues between 2015 and 2019. This framework and systematization provide a natural way for researchers to identify gaps in their study design, thereby creating opportunities to build on earlier efforts by broadening the set of assets discovered. Furthermore, they can help researchers select relevant techniques for the individual steps of our framework based on their needs. Finally, we illustrate how techniques can be combined and how to identify where gaps remain.

Chapter 3 aimed at shedding light on signature-based NIDSs and the rulesets they employ to detect threats. Using 13 years of NIDS rule management data and internal alert and incident data from a managed security service provider (MSSP), we analyze the evolution of said rules and rulesets, as well as how the rules influence the detection of threats and the investigation of security incidents. We find that, overall, barely a fifth of all rules are meaningfully updated throughout their lifespan (i.e., in a way that alters their detection capabilities), either to account for changes in the threat landscape, or to reduce the amount of generated alerts by making rules more specific to certain threats, thereby alleviating the workload on the SOC and its analysts. Furthermore, half a percent of rules are responsible for 80% of all alerts. Of all these alerts, we find only a 1.2% are significant enough to warrant closer investigation, of which only a fourth carry any risk to an organization.

Chapter 4 studies the organizational processes surrounding the management of signature-

based NIDSs and their respective rulesets. Although many of these processes are technical in nature, they are carried out manually by a team of professionals, and end up determining the types of security incidents and the degree to which they are detected and resolved. We find that there are a number of critical factors that dictate the manner in which rules are managed, and that there was significant consensus among participants regarding the importance of these factors when they are used to determine the quality of rules and rulesets. Still, we find that there is no single manner in which all SOCs are managed. The different factors need to be managed and balanced against each other within the context of, e.g., the SOC's rule management processes, analyst experience, resource availability, and customer expectations.

Chapter 5 investigates how a collection of external network signals can be used to predict the occurrence of internal network security incidents. To this end, we use nearly five years of external network scan data, malicious activity reports and abuse datasets, to train a model to predict internal security incidents. This system is able to predict with high accuracy the number of internal security incidents within a certain time span, thereby demonstrating the predictive power of external network data. Additionally, we find that this performance is highly dependent upon the sparsity of the input features extracted from the external data and abuse datasets. These findings allow organizations to identify troubling security signals ahead of time, thereby potentially be kick-starters for organizations to take a more proactive approach to their network security.

Finally, Chapter 6 presents the main findings and outcomes, and reflects on the governance implications that result from the creation and widespread use of risk estimation and security incident prediction technology. While previous work has demonstrated the effectiveness and predictive power of external network signals when observing and exploring large data breaches, none had examined such relationships with internal security incidents. We discovered a myriad of asset discovery techniques buried within recent academic literature that have not found themselves in the field of external network security measurements, despite their clear relevance. These additional discovery techniques can shed much more light on an organization's level of security. Moreover, the additional exposure discovered through these techniques can potentially translate into internal security incidents detected by an NIDS and investigated by a SOC. We find that the relationship between external network signals and the internal state of a network is significant enough for the former to have predictive power for the latter. However, the effectiveness of this prediction system depends on various factors that vary between organizations, and, hence, such systems need to be continually updated and retrained as each organization's threat landscape and security practices evolve. Nevertheless, our work successfully demonstrates the feasibility of this approach.

SAMENVATTING

Het spreekt voor zich dat het internet verre van veilig is. Naarmate het aantal met internet verbonden apparaten toeneemt, neemt ook het aantal cyberaanvallen waarmee we te maken krijgen toe. Talloze brancherapporten laten jaar na jaar aanzienlijke stijgingen in softwarekwetsbaarheden zien. Dit zijn problemen waarmee ondernemingen van elke omvang, binnen de publieke en private sector, te kampen hebben. In het licht van deze bevindingen wordt het voor bedrijven, ongeacht hun omvang, absoluut noodzakelijk om prioriteit te geven aan cyberbeveiliging en hun huidige verdedigingsmechanismen tegen dit evoluerende dreigingslandschap opnieuw te evalueren.

Het zich ontwikkelende landschap van cyberdreigingen heeft het belang benadrukt van het aannemen van een proactieve aanpak om cyberveiligheidsrisico's te beheren en te beperken. Organisaties kunnen een groot aantal stappen ondernemen om dit te bereiken, maar kiezen vooral voor beveiligingsmaatregelen die draaien om compliance-eisen en gestandaardiseerde methodologieën en frameworks om de algehele beveiligingsniveau te verbeteren. Het aanpassen en integreren van deze gestandaardiseerde methodologieën is niet alleen een technische onderneming; het vereist ook aanzienlijke financiële en menselijke middelen. Het blijft echter onduidelijk in hoeverre dergelijke investeringen het gewenste effect hebben. Dit komt vooral omdat veiligheid een latente eigenschap is die niet direct kan worden gemeten, wat betekent dat veranderingen daarin door middel van investeringen in veiligheid niet op een eenvoudige manier kunnen worden bepaald.

Er zijn recentelijk alternatieve benaderingen ontstaan die tot doel hebben de veiligheid op een directere manier te meten. In plaats van te vertrouwen op zelfgerapporteerde gegevens verzamelen bedrijven extern toegankelijke gegevens zoals verkeerde netwerk- of serverconfiguraties, evenals gegevens over kwaadwillige activiteiten binnen het netwerk van de organisatie (bijvoorbeeld hosts die op verschillende zwarte lijsten van misbruik staan). Vervolgens wordt met deze data een classifier getraind, waardoor deze met een zekere mate van nauwkeurigheid kan voorspellen welke organisaties waarschijnlijk te maken zullen krijgen met (grootschalige) inbreuken.

Hoewel er een verband lijkt te bestaan tussen externe metingen en inbreuken, moet de causaliteit tussen beide nog worden bevestigd door onafhankelijk onderzoek, waardoor de haalbaarheid van hun inspanningen in twijfel kan worden getrokken. Bovendien is het niet

duidelijk hoe statistieken die zijn afgeleid van puur externe metingen zich verhouden tot het beveiligingsniveau dat is afgeleid van interne metingen van het netwerk van een organisatie. Hieruit blijkt de noodzaak om bij het observeren van externe beveiligingssignalen rekening te houden met de interne staat van netwerken, in plaats van uitsluitend te vertrouwen op openbaar gerapporteerde datalekken.

Door uitsluitend externe netwerkgegevens te gebruiken, worden organisaties als zwarte dozen behandeld. Er is geen manier om te bepalen of deze signalen iets zeggen over de interne staat van het netwerk. Bovendien zijn de voorspellingen zeer breed. Het introduceren van interne netwerkgegevens van *network intrusion detection systems* (NIDS's) die worden verwerkt en onderzocht door *security operations centers* (SOC's) in dit probleem kan worden gebruikt om de verbanden tussen externe netwerksignalen en de interne staat te bestuderen. NIDS's en de regelsets die zij gebruiken zijn verantwoordelijk voor de detectie van potentieel kwaadaardige activiteiten en dienen daarom als vensters op de interne status van het netwerk van een organisatie.

Deze dissertatie bestudeert de haalbaarheid van het voorspellen van beveiligingsincidenten en risicoschatting. Het onderzoekt hoe externe netwerkscangegevens kunnen worden gebruikt om informatie af te leiden over de interne beveiligingsstatus van het netwerk van een organisatie. Daarom formuleren we de volgende onderzoeksvraag: *Hoe kunnen interne beveiligingsincidenten worden voorspeld door gebruik te maken van externe netwerksignalen?* In de volgende hoofdstukken worden de verschillende onderzoeken gepresenteerd die er gezamenlijk op gericht zijn verschillende delen van deze overkoepelende onderzoeksvraag te beantwoorden.

Hoofdstuk 2 verzamelt en systematiseert nieuwe technieken voor *asset discovery*. We presenteren een framework voor asset discovery dat een syntax voorstelt om de stappen in het asset-discovery proces expliciet te maken en gebruikte het om alle asset discovery technieken te systematiseren die zijn ontworpen of genoemd in de literatuur gepubliceerd in 14 toonaangevende academische locaties tussen 2015 en 2019. Dit framework en de systematisering bieden een natuurlijke manier voor onderzoekers om gaten in hun studieontwerp te identificeren, waardoor ze kunnen voortbouwen op eerder werk door het uitbreiden van de set van ontdekte assets. Bovendien kunnen ze onderzoekers helpen relevante technieken te selecteren op basis van hun behoeften. Tot slot illustreren we hoe technieken kunnen worden gecombineerd en hoe te identificeren waar technieken ontbreken.

Hoofdstuk 3 onderzoekt *signature-based* NIDS's en de regelsets die ze gebruiken om dreigingen te detecteren. Met behulp van 13 jaar aan NIDS-regelbeheergegevens en interne waarschuwings- en incidentgegevens van een *managed security service provider* (MSSP), analyseren we de evolutie van regels en regelsets, evenals hoe de regels de detectie van dreigingen en het onderzoek van beveiligingsincidenten beïnvloeden. We ontdekken dat, over het algemeen, nauwelijks een vijfde van alle regels zinvol wordt bijgewerkt gedurende hun levensduur (d.w.z. op een manier die hun detectiecapaciteiten verandert), hetzij om rekening te houden met veranderingen in het dreigingslandschap, of om de hoeveelheid gegenereerde waarschuwingen te verminderen door regels specifieker te maken voor bepaalde dreigingen. Hierdoor wordt de werkdruk op het SOC en zijn analisten verlicht. Bovendien is een half procent van de regels verantwoordelijk voor 80% van alle *alerts*. Van al deze alerts zijn slechts 1,2% belangrijk genoeg om aanvullend onderzoek te rechtvaardigen, waarvan slechts een vierde enig risico voor een organisatie met zich meebrengt.

Hoofdstuk 4 bestudeert de organisatorische processen rond het beheer van signature-based NIDS's en hun respectievelijke regelsets. Hoewel veel van deze processen technisch van aard zijn, worden ze handmatig uitgevoerd door een team van professionals, en bepalen ze uiteindelijk de soorten beveiligingsincidenten en de mate waarin ze worden gedetecteerd en opgelost. We ontdekken een aantal kritieke factoren die de manier dicteren waarop regels worden beheerd, en dat er aanzienlijke consensus was onder de deelnemers over het belang van deze factoren wanneer ze worden gebruikt om de kwaliteit van regels en regelsets te bepalen. Toch blijkt er geen één manier te zijn waarop alle SOC's worden beheerd. De verschillende factoren moeten worden afgewogen tegen elkaar binnen de context van bijvoorbeeld de regelbeheerprocessen van het SOC, analistervaring, beschikbaarheid van middelen en klantverwachtingen.

Hoofdstuk 5 onderzoekt hoe een verzameling van externe netwerksignalen kan worden gebruikt om het optreden van interne netwerkbeveiligingsincidenten te voorspellen. Hiertoe gebruiken we bijna vijf jaar aan externe netwerkscangegevens, rapporten van kwaadaardige activiteiten en internet abuse datasets om een model te trainen die interne beveiligingsincidenten kan voorspellen. Dit systeem kan met hoge nauwkeurigheid het aantal interne beveiligingsincidenten voorspellen binnen een bepaalde tijdspanne. Hiermee wordt de voorspellende kracht van externe netwerkgegevens aangetoond. Bovendien constateren we dat deze nauwkeurigheid sterk afhankelijk is van de schaarsheid van de *input features* die uit de externe gegevens en abuse-datasets worden gehaald. Deze bevindingen stellen organisaties in staat om verontrustende beveiligingssignalen vooraf te identificeren, waardoor ze mogelijk een proactievere benadering van hun netwerkbeveiliging kunnen nemen.

Tot slot presenteert Hoofdstuk 6 de belangrijkste bevindingen en resultaten van deze dissertatie. Verder reflecteert het op de bestuursimplicaties die voortvloeien uit de creatie en wijdverbreid gebruik van risicoschatting en voorspellingstechnologieën. Hoewel eerdere literatuur de effectiviteit en voorspellende kracht van externe netwerksignalen heeft aangetoond bij grote datalekken, heeft niemand dergelijke relaties met interne beveiligingsincidenten on-

derzocht. We ontdekken een groot aantal asset discovery technieken in recente academische literatuur die zich nog niet in het veld van externe netwerkbeveiligingsmetingen bevonden, ondanks hun duidelijke relevantie. Deze extra ontdekkingstechnieken kunnen veel meer licht werpen op het beveiligingsniveau van een organisatie. Bovendien kunnen de door deze technieken ontdekte assets potentieel worden vertaald in interne beveiligingsincidenten die door een NIDS gedetecteerd en door een SOC onderzocht worden. We stellen vast dat het verband tussen externe netwerksignalen en de interne staat van een netwerk significant genoeg is voor voorspellende kracht. De effectiviteit van dit voorspellingssysteem is echter afhankelijk van verschillende factoren die variëren tussen organisaties. Daarom moeten dergelijke systemen continu worden bijgewerkt en getraind naarmate het dreigingslandschap en de beveiligingspraktijken van elke organisatie evolueren. Desalniettemin demonstreert ons werk met succes de haalbaarheid van deze aanpak.

1

Introduction

In 2022 we witnessed an all-time high volume of cyberattacks, reaching nearly 1,200 attacks per organization in the last quarter of the year. This upswing indicates a 38% increase in cyberattacks on corporate networks in 2022 compared to the previous year, suggesting that we are currently in the midst of several converging cyber threat trends [46]. This is reflected in the number of published CVEs, rising 24% from 20,161 in 2021 to 25,059 in 2022 [61]. Not all CVEs are created equal, though, some being more severe than others. Thus, perhaps a more striking development in the same time span is the increase in the number of published CVEs with a critical CVSS score of 9 and higher: nearly 40%, from 3,596 to 5,008 [204].

However, these threats are not limited to the corporate giants. Disturbingly, nearly 43% of these cyberattacks targeted small and medium businesses [17]. The vulnerability of this sector becomes even more evident when considering that a mere 14% of these smaller businesses are adequately prepared to defend against such threats, underscoring the significant security gap present in this sector.

Adding to the concerns, IBM's recent data breach report suggests that the frequency of security lapses has become alarmingly common. A staggering 83% of organizations reported experiencing more than one data breach during 2022 [104]. Financial repercussions from these incidents have also intensified. The global average cost of a data breach witnessed an increase, climbing from $4.24 million in 2021 to $4.35 million in 2022 [104].

In light of these findings, it becomes imperative for businesses, regardless of size, to prioritize cybersecurity and re-evaluate their current defense mechanisms against this evolving threat landscape.

1.1. Background

Organizations are not defenseless against threats. Considerable effort is exerted to secure an organization from both external and internal threats. These efforts exist in different forms, such as procedural (compliance frameworks) and technical (antivirus), or reactive (incident detection and response) and proactive (penetration testing).

1.1.1. Frameworks and security

The evolving cyber threat landscape has emphasized the importance of adopting proactive approaches to manage and mitigate cybersecurity risks. Organizations can take a great many steps to achieve this. These can include the use of antivirus software and periodic penetration testing. A prominent example is red/blue or purple teaming. A red team comprises offensive security experts that attempt to penetrate an organization's security defenses. By contrast, a blue team defends against the red team's intrusion attempts. A purple team blends both approaches together, allowing for greater collaboration between members of both teams.

However, organizations mainly choose security measures that revolve around compliance requirements and standardized methodologies and frameworks [149] to improve overall security posture.

Such standardized methodologies and frameworks provide structured guidelines and best practices tailored to address a spectrum of potential threats and vulnerabilities. Frequently used frameworks include the NIST Cybersecurity Framework [157], CIS Controls (formerly SANS Critical Security Controls) [44], and ISO/IEC 27001 [109].

The NIST Cybersecurity Framework [157] is a product of the U.S. National Institute of Standards and Technology. It offers a comprehensive structure, guiding organizations in the direction of managing and reducing cybersecurity risks. The framework is divided into five core functions: Identify, Protect, Detect, Respond, and Recover. Each function provides a high-level, strategic view of an organization's management of cybersecurity risk, allowing for a broad perspective on organizational preparedness and response mechanisms.

The CIS Controls [44], developed by the Center for Internet Security, represent a more specific set of actions that organizations should adopt to enhance their cyber defense posture. Unlike the broader strategic focus of the NIST and ISO frameworks, the CIS controls provide organizations with a granular approach, outlining specific defensive actions to tackle an array of cyber threats. The controls encompass various areas, from inventory and control of hardware assets to maintenance, monitoring, and analysis of audit logs.

Lastly, ISO/IEC 27001 [109] provides an international standard focused on information security management. Rooted in a systematic approach, this framework emphasizes the management of sensitive company information. Organizations that adopt this standard are typically looking to ensure the confidentiality, integrity, and availability of their data. Key elements of ISO/IEC 27001 revolve around assessing risks, establishing controls, and continuous monitoring and improvement of the information security management system (ISMS).

The implementation of cybersecurity frameworks and other associated information security measures has become a significant financial investment. Adapting and integrating these standardized methodologies is not merely a technical endeavor; it also entails considerable monetary and human resources, and time commitments.

The magnitude of this investment is evident when looking at the broader industry trends. Financial forecasts highlight the rising commitment of businesses to bolster their cybersecurity defenses. Spending on information security and risk management products and services is on a sharp incline. The market's growth trajectory is set to continue, with a predicted rise of 11.3% in 2023, leading to the global market surpassing $188.3 billion [87].

Clearly, the intention is to improve an organization's security posture through investments in the organization's security. However, it remains unclear to which extent such investments have their desired effect. This is mainly because security is a latent property that cannot be measured directly, meaning that changes to it through means of security investment cannot be determined in a straightforward manner.

1.1.2. MEASUREMENT OF SECURITY

The prevailing industry standard for determining level of security is based on two different aspects: the behavior of other organizations, and degree of compliance to several standardized security frameworks and best practices. Specifically, if many organizations gravitate towards certain controls or frameworks, then they are assumed to be effective. For the compliance aspect, firms verify whether the organization adheres to a number of predefined security standards.

Alternative approaches have recently emerged that aim to measure security in a more direct manner. Instead of relying on use of controls or adherence to frameworks, they measure the resulting (malicious) activity that might be a consequence of all the technologies and frameworks put into place by an organization.

A number of new risk rating services have emerged that take a more direct approach to security and risk measurements. Such firms include SecurityScorecard [203], BitSight [29] and Quadmetrics [118] (recently acquired by FICO [79, 147]). These risk-rating services have their basis in research that has undergone validation through peer-reviewed publications, with the research conducted by Liu et al. [133] being a prominent instance. Instead of relying on self-reported data, internal or otherwise, from client companies, these firms firms gather externally accessible data such as misconfigured DNS, SMTP, and untrusted TLS certificates, as well as data concerning malicious activities within the organization's network (e.g., hosts listed in various abuse blacklists). Subsequently, a classifier is trained using this data, enabling it to predict, with a certain level of accuracy, which organizations are likely to experience breaches (based on the large-scale breaches recorded in the VERIS Community Database [238] and other public breach notification databases).

However, there are a number of issues that remain unaddressed. First and foremost, while there seems to be a correlation between external measurements and breaches, causality between the two has yet to be confirmed by independent research, calling into question the feasibility of their endeavor. Furthermore, it is not clear how metrics derived from purely external measurements compare to the security level derived from internal measurements of an organization's network.

This reveals the necessity of taking into account the internal state of networks when observing external security signals, instead of relying on externally observable or publicly reported data breaches. Such a window into the state of an organization's internal network can be found through controls that are installed to monitor said networks and detect threats: intrusion detection systems.

1.1.3. NIDSs and SOCs

One of the main lines of defense that organizations employ against internal and external threats is the intrusion detection system (IDS). These are systems that attempt to detect malicious activity that occurs within a network and raise alerts or drop network traffic when such activity is detected. IDSs are classified by their placement and by the techniques that are used for detection. As for the placement of an IDS, there are network-based, host-based, and application-based intrusion detection systems. A network-based IDS (NIDS) is named as such, because it is placed at a strategic point within a network and analyzes all network packets it receives to detect attacks. It is generally a single, independent system whose only purpose is to capture and analyze network traffic, allowing for the monitoring of large networks.

Furthermore, depending on its method of detection, an IDS can be signature-based or anomaly-based [146]: detecting malicious activity by means of rules that characterize known malicious activity, or statistical- or machine learning-based techniques to identify deviations from a network's standard behavior, respectively.

Signature-based IDSs find malicious activity by matching it with a predefined set of patterns or events that are characteristic of known attacks. Although this technique certainly is effective at detecting known attacks, it struggles at detecting novel attacks. This is because a signature for the novel attacks is not available yet. Anomaly-based IDSs work with the assumption that malicious activities behave differently than normal activities. By establishing a baseline for normal behavior, it tries to detect the malicious activities by identifying these differences. In this manner, it aims to detect not only previously-seen attacks, but novel attacks as well. This approach is hardly perfect, with false positives being one of the major drawbacks and point of contention [146].

Recent studies in network intrusion detection have increasingly leaned towards statistical and machine-learning approaches [148, 209, 220]. Due to the necessity of manual maintenance, both industry and academia have held the belief that traditional rule- and signature-based techniques will not be able to match the pace of rapidly advancing threats, thereby rendering them outdated. Nevertheless, these types of NIDSs can still be found performing their tasks dutifully and effectively. Even though industry players have long lamented their lack of quality [39], these systems remain a cornerstone of network security, as evidenced by their continued use throughout SOCs globally [8, 120].

Oftentimes, NIDSs are maintained and managed by a security operations center (SOC), which is composed of a team of security professionals and analysts. The SOC receives the output from, e.g., NIDSs, and processes the alerts, investigates potential security incidents, and provides incident response to maintain an organization's network security.

Naturally, alerts produced by NIDS—if managed properly—say something about the internal state of a network. There is much flexibility in the information that you can acquire from an NIDS, as it all depends on what a rule is designed to detect. Network intrusions are not the only occurrence that can be detected from analyzing network traffic. They can be designed to detect numerous sorts of unwanted behavior, such as unauthorized or out-of-date software use, suspicious web traffic originating from within the network, or other types of network policy violations. Furthermore, they can be built to detect threats from both internal and external sources. Such functionality is not only useful for reactive security in case of an actual breach, but for proactive security efforts such as red, blue, or purple teaming [59]. Thus, MSSPs and organizations themselves can use NIDS data for processes such as internal diagnostics and evaluations of network security and can launch investigations or deliberate with the client in order to correct its cause.

1.2. TOWARDS EMPIRICAL RISK ESTIMATION

As opposed to previous work [133], which bases its predictions and risk estimations on large, publicly-reported breaches to which organizations have fallen victim, internal SOC and

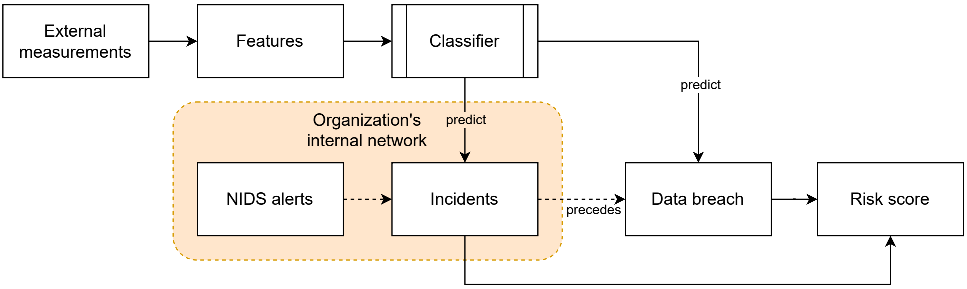


Figure 1.1: A simplified version of a risk measurement pipeline. Previous work [133] predicts the occurrence of large data breaches directly, thereby excluding the internal network items in the orange area from their designs, and depriving analysts of the ability to make observations about the internal state of their network.

NIDS data is much finer-grained and enables us to to look at the issues of incident prediction and risk estimation from a new perspective. What previous work fails to mention, however, is that these large-scale breaches are often the result of (persistent) malicious activity within an organization's network. Such malicious activity is likely recorded in the form of NIDS alert and security incident logs. Figure 1.1 illustrates this. This internal malicious activity is, thus, a precursor to large-scale data breaches. It not only provides information regarding successful breaches into an organization's network, but large collections of unsuccessful breach attempts and lower-level malicious activity that takes place periodically within any network. Hence, it provides a larger window into the internal state of its respective network. Assuming the validity of previous work and that these types of breaches reveal information regarding an organization's security posture, it stands to reason that the findings will be consistent when attempting this different, finer-grained approach.

Adequate risk estimation requires additional elements. We know that attacks are the principal cause of incidents. However, the extent to which attacks translate into actual incidents is moderated by an organization's security and its exposure [162]. An organization's exposure can refer, for instance, to the size of its IP space, and the number of assets it owns and/or operates. Thus, thoroughly measuring an organization's exposure is vital to empirical security and risk estimation.

As long as such accurate metrics are not developed and used in cybersecurity, securing people and organizations will remain practically guesswork. This dissertation aims to move us closer toward to that objective.

1.3. RESEARCH GAPS

As stated in Section 1.1.2, empirical risk measurement as performed by the aforementioned firms is substantiated by peer-reviewed work [121, 133]. Using exclusively external network data, they are able to infer security risk information. While the predictive power of these methods provides some evidence for their validity, they do treat the organization as a black box. There is no way to tell whether these signals say anything about the internal state of the network. Furthermore, the predictions are very coarse-grained: Liu et al. [133] distinguish organizations that were victim of large breaches from organizations that are not, while Kotzias et al [121] focus their analysis on a narrow range of signals (malware and potentially unwanted programs). Introducing internal network data from NIDSs that is processed and examined by SOCs into this problem can be used to study the links between external network signals and the internal state.

Indeed, NIDSs and SOCs play a major role in this research. An NIDS and the rulesets it employs are responsible for the detection of potentially malicious activity. They serve as the window into the internal state of an organization's network. However, NIDS alerts are known to be considerably noisy [8], and although we know the rules and their design is what is culpable for the noise, it is unknown what drives their quality. In fact, most of the literature related to NIDS and its rulesets revolves around system performance metrics such as CPU load, memory usage and packet loss [10, 63, 207, 246]. Additionally, managing rulesets and processing alerts is a human task that also ultimately influences what a SOC is able to "see" within a network. Thus, the non-technical processes that govern NIDS and ruleset management are inseparable from the technical processes, since the latter is rooted in the former. As of yet, we lack the knowledge regarding how SOCs ensure the quality of the rules such that networks can be maintained secure and malicious adversaries can be thwarted. This, in turn, allows us to understand how data from NIDSs and SOCs can be used to create a picture about the security posture of organizations.

Much of the recent Internet measurement and security literature mentions or develops novel methods of external asset discovery, which remains unused in this type of research. However, it seems that a large portion of these techniques are created in a vacuum, fading into obscurity as the novelty of the original paper diminishes. Still, these are techniques that can be combined, and many of them chained one after another to extract a wide array of an organization's network assets. As the focus of the previous security incident prediction and risk estimation work lies on external measurements, it is necessary to be able to create the most comprehensive view of an organization's Internet exposure. This makes the aforementioned literature (and asset discovery methods detailed within) highly relevant to this field of research, where previous work has focused on a limited collection of features and services [121, 133]. By compiling the largest possible collection of potentially relevant external signals, similar systems can be implemented that are able to create much more accurate risk estimations.

Altogether, we discover three main gaps in the current research: 1) we lack the knowledge regarding state-of-the-art asset discovery techniques that are relevant for comprehensive external network measurements, 2) we do not know the driving factors behind NIDS rule and alert quality, limiting the value they provide regarding the internal state of the network, and 3) it is unclear whether, and to which extent, external network signals are indicative of the internal state of a network. We will conduct four research activities to address the identified gaps.

Firstly, we will create a comprehensive compilation of state-of-the-art external asset discovery techniques, and specify how to chain said techniques to extract the desired network information. Secondly, we will analyze the rules used by NIDSs, what factors drive their quality, the alerts they trigger, and the security incidents that the SOC investigates. Thirdly, we examine the human aspects of SOC and NIDS management that determine how security professionals ensure effective SOC operation. We do this by means of semi-structured interviews that allow us to discover the heuristics and thought processes that SOC professionals employ. And lastly, we combine these observations and insights and create a predictive model that attempts to link external network signals to internal security events.

1.4. RESEARCH OBJECTIVE

This dissertation studies the feasibility of security incident prediction and risk estimation. It examines how external network scan data can be leveraged to infer information about the internal state of security of an organization's network.

Thus, we formulate the following research question:

> How can internal security incidents be predicted
> through the leverage of external network signals?

In this dissertation, we lay the focus towards comprehending the relationship between external network signals and the internal network state. Conducting this research requires thorough examination of both of the aforementioned facets through the leverage of a mixed-methods approach: discovering and compiling sufficient external network signals, and understanding both the technical and non-technical aspects of internal security event detection and response. Specifically, the scientific contribution of this dissertation lies in tying seemingly unrelated external network signals to the occurrence of internal security events detected within the network of organizations, thereby advancing the progression of empirical risk estimation techniques.

Due to the nature of the overarching problem, this research requires different methods of investigation. The exploration encompasses not only the technical aspects of cybersecurity but also extends to the various actors involved. Security analysts whose heuristics and thought processes affect the manner in which they examine security incidents, for instance.

Thus, this research is interdisciplinary from the offset. We look at how to develop a new technology within an organization and its IT infrastructure, taking into account the influence that the actors within that organization have on the systems being studied to create that technology. We, therefore, utilize quantitative and qualitative analysis in tandem. The chapters in this dissertation being based on research papers, each individual chapter will expand upon its portion of the research methodology.

1.5. DISSERTATION OUTLINE

This section provides an outline of the dissertation. It describes the four studies that were conducted, as well as the research questions posed. This section also presents an overview of the peer-reviewed papers linked to their corresponding study and chapter.

STUDY 1

Asset discovery is fundamental to any organization's cybersecurity efforts. Indeed, one must accurately know which assets belong to an IT infrastructure before the infrastructure can be secured. While practitioners typically rely on a relatively small set of well-known techniques, the academic literature on the subject is voluminous. We systematize asset discovery techniques by constructing a framework that comprehensively captures how network identifiers and services are found. We extract asset discovery techniques from recent academic literature in security and networking and place them into the systematized framework. We then demonstrate how to apply the framework to several case studies of asset discovery workflows, which could aid research reproducibility.

The research question addressed by this study is the following:

RQ1: How can an organization's assets and controls be accurately identified through external measurements?

STUDY 2

Ultimately, internal network security can only be measured by the occurrence of security incidents, or lack thereof. And in cases of signature-based NIDSs, SOCs and analysts can only detect incidents for which they write a rule for. However, as threats evolve, so, too, must an organization's defenses. Hence, we study the NIDS operated by a major MSSP. We use longitudinal datasets of the rulesets employed by the MSSP, the alerts that are triggered by the rules, and the resulting security incidents that are investigated by the SOC to track the evolution of rules and how such evolution affects detection performance and examine how rule management processes are designed to achieve security. We perform in-depth interviews with rule developers to corroborate the patterns found in our analysis. Finally, we identify several rule management practices that influence rule and ruleset efficacy.

The research question addressed by this study is the following:

RQ2: How do security companies develop and manage NIDS rules to optimize incident detection and improve security?

STUDY 3

Although existing literature has attempted to address the complications and shortcomings within SOCs' use of NIDSs (through automation and implementation of ML systems), none have proposed solutions that address the root of the problem: the design and maintenance of NIDS rules. Since this human aspect of security influences how effective a SOC is at detecting incidents, it is crucial to understand the processes that govern it. What does the organizational ruleset management process look like? What are the main factors and success criteria in managing and evaluating NIDS rules and rulesets? How can security professionals improve rule management and network incident monitoring workflows to optimize SOC processes? To understand these processes, we conduct interviews with professionals who work at MSSPs or other organizations that provide network monitoring as a service or conduct their own network monitoring internally. We discovered numerous critical factors, such as rule specificity and total number of alerts and false positives, that guide SOCs in their rule management processes. We present several recommendations that address these lower-level aspects to help improve alert quality and allow SOCs to better optimize workflows and use of available resources.

The research question addressed by this study is the following:

RQ3: How do security professionals manage network incident monitoring processes to achieve security?

STUDY 4

Recently, novel approaches have emerged that directly measure an organization's security using external network signals. While these approaches provide breach prediction and risk ratings inferred from external signals, no link is made to the internal state of organizations' networks. Through our partnership with an MSSP, we acquire data on security events from their clients' networks, and SOC analysts. For the same period, we collect external signals related to these networks. Employing a gradient booster classifier, we successfully predicted the number of high-risk security incidents with high accuracy. Our results confirm the predictive power of external signals for an organization's internal security state, although the sparsity of external signals significantly influences incident prediction.

The research question addressed by this study is the following:

RQ4: How can internal security incidents be predicted using external network signals?

Table 1.1: Dissertation outline

Chapter	Publication
Ch. 2	Vermeer, M., West, J., Cuevas, A., Niu, S., Christin, N., Van Eeten, M., Fiebig, T., Ganán, C. and Moore, T. 2021. SoK: A Framework for Asset Discovery: Systematizing Advances in Network Measurements for Protecting Organizations. In Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P '21), Vienna, Austria, 2021, 440-456, https://doi.org/10.1109/EuroSP51992.2021.00037
Ch. 3	Vermeer, M., van Eeten, M., Gañán, C. 2022. Ruling the Rules: Quantifying the Evolution of Rulesets, Alerts and Incidents in Network Intrusion Detection. In Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security (ASIA CCS '22). Association for Computing Machinery, New York, NY, USA, 799–814. https://doi.org/10.1145/3488932.3517412
Ch. 4	Vermeer, M., Kadenko, N., van Eeten, M., Gañán, C., and Parkin, S. 2023. Alert Alchemy: SOC Workflows and Decisions in the Management of NIDS Rules. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23). Association for Computing Machinery, New York, NY, USA, 2770–2784. https://doi.org/10.1145/3576915.3616581
Ch. 5	*Under submission at IEEE S&P 2025*: Vermeer, M., van Eeten, M., Gañán, C. Network Whispers: Deciphering Incident Predictions with External Signals.

2

External asset discovery

Asset discovery is fundamental to any organization's cybersecurity efforts. Indeed, one must accurately know which assets belong to an IT infrastructure before the infrastructure can be secured. While practitioners typically rely on a relatively small set of well-known techniques, the academic literature on the subject is voluminous. In particular, the Internet measurement research community has devised a number of asset discovery techniques to support many measurement studies over the past five years. In this paper, we systematize asset discovery techniques by constructing a framework that comprehensively captures how network identifiers and services are found. We extract asset discovery techniques from recent academic literature in security and networking and place them into the systematized framework. We then demonstrate how to apply the framework to several case studies of asset discovery workflows, which could aid research reproducibility. These case studies further suggest opportunities for researchers and practitioners to uncover and identify more assets than might be possible with traditional techniques.

2.1. Introduction

The bedrock of good security posture for any organization's IT security is accurate knowledge of which systems belong to its IT infrastructure. If a company does not know what systems and software it is using, it cannot ensure their security and, therefore, cannot secure the organization. All risk management strategies are predicated on having full visibility into the organization's assets. Hence, it is no coincidence that the first function in the NIST

Cybersecurity Framework is to “identify” and the first category within this function is “asset management” [158]. For the same reasons, the ISO/IEC 27001 information security standard [11] requires the identification of assets that are associated with information and information processing facilities, as well as keeping this overview updated.

While the need to identify assets is obvious, how to accomplish this identification in practice is not. Organizations consistently struggle to keep a complete inventory of their assets—and consistently fail to do so. (See [20] for a couple of particularly telling examples.) Even a medium-sized organization can easily deploy tens of thousands of systems, software platforms and applications. This asset inventory is constantly changing, with many changes unplanned or unrecorded. This is further amplified by the problem of “shadow IT”: IT systems that are “not known, accepted and supported” by an organization’s official IT department [189]. All of this means that any centralized inventory of assets, such as those prescribed in the Information Technology Infrastructure Library (ITIL) and ISO procedures, necessarily contain errors and omissions. Automated techniques to identify these gaps are therefore essential for defenders to adopt.

Asset discovery is not only a defensive activity, but also a crucial component of both a red team’s and attacker’s process. Adversaries first need to know *what* they are targeting and what they *want* to target, before they can start attacking systems. Furthermore, the information deficit in organizations, as outlined above, means that attackers can get a serious advantage by having an equally good or even better overview of an organization’s assets than the defenders on the inside of the network.

For both sides, attackers as well as defenders, numerous techniques have been researched and tools have been developed. Many of these techniques are integrated into various industry toolchains and handbooks. However, the field of Internet measurement also blossomed in recent years, and many new techniques have emerged that are not yet used or even known by practitioners involved in asset discovery. The techniques are often developed for other purposes, such as diagnosing network disruptions; they are often not explicitly associated with asset discovery; and they are fragmented across different academic venues and communities. Furthermore, in recent years, automation and commoditization of these tools has also become prevalent, with offensive and defensive uses alike. Automated tools and techniques to discover assets have been used to show just where asset management fails—think of exposed systems showing up in search engines like Shodan or in the scans of pen testers [9, 32]. While automated tools and services leveraging well-known asset discovery techniques are widely available, many defenders may still not be aware of the most recent techniques available to understand their network, and they may lack valuable information on what capabilities for asset discovery attackers potentially have.

In this paper, we fill this gap by surveying and systematizing new developments in asset discovery, *i.e.*, techniques that appeared in papers in 14 leading academic venues over the past five years. Our goal is twofold: First, we provide an overview of recent techniques. This exploration is time-consuming to acquire given that these techniques are spread out over different communities and are often not explicitly associated with the concept of asset discovery. In other words, we aim to lower the search costs for practitioners and researchers to find techniques that they might use or that might be used against them. Second, we provide a systematization of asset discovery techniques that illustrate how the different techniques can be chained to each other for the development of toolchains. One technique's outputs are another technique's inputs. We frequently also observe loops, in which an initial seed of assets is snowballed into an expanded set, which can then become the seed for the next cycle.

The objectives of the asset discovery process determine how to select and combine various techniques. These objectives will vary across the use cases of system administrators, security officers, red-teams, and actuaries. A pentester might only need to opportunistically discover a few vulnerable assets, preferably via passive techniques, to gain a foothold in the target organization. A system administrator, on the other hand, needs to find all Internet-reachable systems in order to secure them and has no reason to avoid active techniques. An actuary, determining the risk exposure of a potential cyberinsurance customer, needs a discovery toolchain that can measure assets' footprints reliably and consistently across large numbers of organizations and can therefore not rely on including the inside knowledge of system administrators in these organizations. One key aim of our systematization is to support professionals in these use cases to improve their asset discovery processes.

We scope our work in two key ways. First, as mentioned above, we focus on the Internet measurement techniques that are reported on in the last five years (2015–2019) in 14 academic venues. We therefore focus on techniques utilized in recent work. These methods are often novel, but not always: if an older, more established technique is still valuable, it could be utilized by the authors. Often, while asset identification is required to complete the research, the paper's novelty lies elsewhere. Hence, while we do not completely catalog all asset identification techniques, we do comprehensively identify how asset identification is being carried out by researchers today. The scoping around Internet-based techniques also means that we focus on active measurement techniques, *i.e.*, we exclude passive techniques that require a privileged vantage point inside a network. This corresponds to our goal of providing a comprehensive document for red-team members during engagements, and a selection of techniques for organizations to understand the capabilities and visibility outsiders can gain on their network.

We provide the following contributions:

- We develop a framework that systematizes asset discovery techniques.
- We comprehensively survey asset discovery techniques extracted from academic literature published in top networking and security conferences, as well as related venues, and map these techniques onto the framework.
- We demonstrate how the framework can be used to construct workflows where techniques can be chained together for asset discovery use cases.

Structure: We introduce our framework, important terminology, and our literature search and systematization Methodology in Section 2.2.1. Subsequently we present our results in Section 2.3, and illustrate our findings with case studies in Section 2.4. Finally, we discuss our findings and conclude in Section 2.5.

2.2. Systematization Methodology

In this section, we first define the necessary terminology and processes to delineate the scope of our subsequent literature review. Our definitions include what we consider an asset, what it means to *discover* an asset, and what the formal meta-process of discovering an asset looks like. Next, we describe our literature review methodology and the steps we took to conduct the survey presented in this work. Finally, we conclude the section by introducing the notation we use for presenting the results of our survey.

2.2.1. Terms and Definitions

Assets The first item to define is the meaning of *asset*. When we turn to existing standardization frameworks for a clear definition, we find that ISO/IEC 27001 in its 2005 revision defines asset very broadly as *"anything that has value to the organization."* Similarly, the NIST framework regards as assets both hardware (*"ID.AM-1: physical devices and systems within the organization"*) and software (*"ID.AM-2: software platforms and applications"*) [158]. These two definitions take two very different perspectives on the term "asset": The NIST framework focuses on "hard" assets, *i.e.*, physical devices, systems, software platforms and applications. Conversely, the ISO/IEC definition also includes "soft" assets, such as information pertaining to the employees of an organization like job titles and email addresses, or business and financial information. As our survey focuses on external *network* measurements and new techniques to identify publicly-visible IT assets, we adopt and rephrase the definition of NIST:

Definition 1 *We consider as assets: (i) all* network identifiers, e.g., *addresses, FQDNs, and contents of DNS zones, and (ii) the* network services *reachable via these network identifiers, defined by the protocol they are implementing, and any information they provide upon initial connect in their banners,* e.g., *implementation names and version numbers.*

We specifically consider as out-of-scope inferred properties of these assets, *e.g.*, whether a certain network service is vulnerable to an exploit (either based on reasoning about the version number or attempting the exploit), inferring whether multiple network identifiers point to the same physical or logical host, or whether multiple discovered network services constitute a joint application service.

Asset Discovery Next, we define asset *discovery*.

Definition 2 *The discovery of an asset means that the existence of an asset associated with a specific organization becomes, for the first time, known to an entity.*

This entails that the entity which discovers an asset has not been aware of its existence, and that the discovery of an asset is independent of third parties knowing about the asset prior to the discovery process.

Bootstrapping Next, the issue of asset discovery leads to the question of *whose* assets are to be discovered. Going from our definition of discovery, we want to discover assets that belong to *specific* entities. Depending on the asset discovery technique used, it might be necessary to first manually discover and select assets connected to an entity, before the technique can be applied. We call this step "bootstrapping" and define it as follows:

Definition 3 *Bootstrapping is the process of obtaining the initial seed of information that the asset discovery techniques require as input to discover assets.*

The information selected in the bootstrapping stage differs depending on the objective. For penetration tests or network monitoring, one searches a specific organization's assets. For benchmarking, risk profiling or risk prediction, the process might focus on the assets of a whole sector or group of organizations. In general and especially for adversarial asset discovery, the discoverer does not know a company's address space in advance. Even in the case of pen-testing, the organization might not have a complete understanding of all addresses where assets reside, because of shadow IT, outsourcing and cloud infrastructure.

Starting with network addresses is a common, but certainly not the only, technique for bootstrapping. It is relatively straightforward, as the organization name can be fed into common search engines and databases that associate it with the addresses and networks registered by the organization. Examples are querying databases that contain WHOIS, BGP [123, 134] or passive DNS [224] data for the relevant strings. Another example is using a general search engine to find web domains containing an organization's name [134]. Along the same lines, one can search for specific strings in databases of SSL/TLS certificates, as Bonkoski et al. do, to extract relevant domain names [33]. In case of a correct match, the certificate will contain domain name information on the organization of interest.

Bootstrapping techniques using string matching have to contend with incorrect identification. Organization names might not be unique, or overlap with other names. A match of the name with a WHOIS record might thus incorrectly attribute the IP range of a similarly-named organization to an organization in the asset discovery scope. At the same time, an organization could own several network identifiers that cannot be matched with their name. This, for example, occurs during mergers and acquisitions, when IP ranges that used to belong to an acquired entity are still registered under the old name. Depending on the use case for asset discovery, different strategies are needed to deal with incorrect identification. We will discuss these in Section 2.4.

The asset discovery process Finally, from our definition of assets, discovery, the initial bootstrapping step, and the premise of asset discovery via external measurement, we arrive at a model for the formal asset discovery process, shown in Figure 2.1:

Definition 4 *The general asset discovery process consists of an initial bootstrapping process, feeding network identifiers into a recursive process to discover more network identifiers (to be fed back into the process) and network services associated with these identifiers.*

The initial discovery of network identifiers is restricted to the organization of which assets should be discovered (0), which may then yield further associated network identifiers for investigation (1). The discovery of network services associated with network identifiers (2) is straightforward, *i.e.*, one just checks for open ports on the network address (if the identifier is an address) or resolves the identifier to a network address and then checks for open ports. In addition to basic information on the network service, the banners of the detected open ports may then reveal further network identifiers (3) or further network services (4). As per our asset definition above, we explicitly leave out-of-scope the discovery of known software vulnerabilities in discovered network services.

2.2.2. Literature Search Process

Based on our earlier definition of assets, and the asset discovery process, we can now search the scientific literature for methods and techniques informing or enabling the asset discovery process. We scope our literature search to the major publication venues of Computer Security, Network Measurement, and Network Operations from the past five years (*i.e.*, 2015–2019).

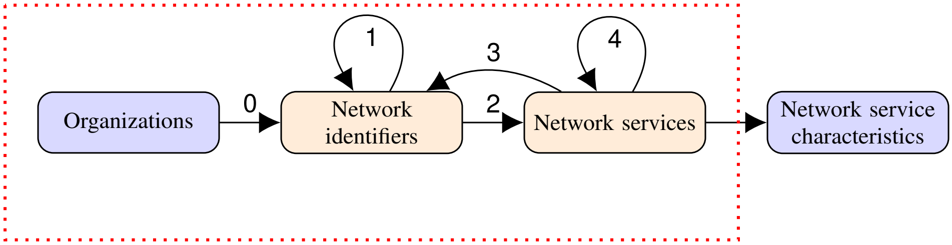


Figure 2.1: Framework for asset identification techniques. Each arrow represents a set of discovery techniques. We highlight the scope of our work with the red dotted line. The inference of software vulnerabilities, either by directly testing for them or by inferring them from version numbers, is the next step after asset discovery and explicitly out of scope for our survey. See Table 2.1 for examples for network identifiers and services.

Initially, four of the co-authors independently investigated 940 papers from five years of a leading security conference, ACM CCS, and a leading networking conference, ACM IMC. The researchers were tasked to identify papers that performed asset discovery. They then sought consensus by discussing conflicts, i.e., papers not included or excluded by all researchers. Because our research setting focused on achieving consensus, we opted to not explicitly calculate intercoder reliability, consistent with the recommendations of McDonald et al. [145].

Using the 32 selected papers, the entire team identified criteria for what constitutes asset discovery, and hence the exclusion and inclusion of papers, and applied those criteria to the remaining venues indicated in Table 2.2. We discarded papers clearly unrelated to the subject based on the papers' abstract. If the abstract and/or introduction of a paper indicated usage of an asset discovery process, we searched the rest of the sections. In cases where we did not find any asset discovery process, the paper was discarded.

In total, we found 93 papers that utilized asset discovery techniques. Notably, the venues from which we select most papers are PAM and ACM IMC. This aligns with our expectations, as both are venues focused on network measurements and novel measurement techniques. Other networking venues, as for example USENIX NSDI and ACM SIGCOMM saw generally fewer papers selected. This is related to those venues also featuring a major fraction of non-measurement networking papers, for example, high-performance networking related research. For security venues, the number of selected papers is again lower than for the purely network measurement focused venues. Again, this can be explained by the more diverse focus of these venues.

Table 2.1: Examples for "Network Identifiers" and "Network Services" in Figure 2.1 encountered during our literature survey. While the list of network identifiers is exhaustive, the list of network services is not. The list only contains the network services that are explicitly mentioned in the literature.

Network identifiers	**Network services**
IPv4 address	Web server
IPv6 address	Name server
Domain and subdomain	Proxy server
Autonomous System	Mail server
BGP prefix	SSH server
IPv4 prefix	FTP server
IPv6 prefix	Cryptocurrency clients
	VPN
	Honeypot
	CMS services
	...

Table 2.2: Overview of the 93 papers we selected from six major security and seven networking venues analyzed during the literature survey. IEEE/IFIP NOMS only takes place every second year.

	2015		**2016**		**2017**		**2018**		**2019**		**Total**	
Security	*Papers*	*Selected*	*Papers*	*Selected*	*Papers*	*Selected*	*Papers*	*Selected*	*Papers*	*Selected*	*Papers*	*Selected*
ACM CCS	128	6	137	2	151	2	134	0	177	2	727	12
IEEE S&P	55	0	55	0	60	1	62	2	84	0	316	3
ISOC NDSS	51	2	60	3	68	0	71	0	89	1	339	6
USENIX Security	67	1	72	2	85	2	100	4	113	1	437	10
RAID	28	0	21	2	21	0	32	1	37	1	139	4
IEEE/IFIP DSN	75	0	65	2	80	1	62	4	54	1	336	8
ACSAC	48	0	48	3	48	1	60	1	60	0	264	5
Networking												
ACM SIGCOMM	40	0	39	0	37	0	40	1	32	1	188	2
ACM IMC	43	2	46	6	42	3	43	3	39	6	213	20
USENIX NSDI	42	0	45	0	46	1	40	0	49	1	222	2
USENIX ATC	47	0	47	1	60	0	76	0	71	0	301	1
IEEE/IFIP NOMS	–	–	222	2	–	–	221	0	–	–	443	2
PAM	27	4	30	1	20	2	20	7	20	0	117	14
TMA	16	0	16	2	29	0	34	1	35	1	130	4
Total	667	15	903	26	747	13	995	24	860	15	**4,172**	**93**

2.2.3. Systematization Syntax and Structure

The papers that we select for the systematization contain tools or techniques that can be used for asset discovery. While some papers have an asset discovery tool or technique as the main contribution, many simply use discovery as a means to an end. To perform a meaningful systematization and comparison of techniques, we have to transform this heterogeneous body of literature to a uniform, comparable terminology.

We formalize each paper's contribution to asset identification by identifying the corresponding edge it represents in Figure 2.1. We base this on our definition of the general asset discovery process: all tools and techniques described in the selected papers work from an asset as input, be it a network identifier, or network service. This input is then used to execute any number of tasks, after which the tool or technique produces a certain output, which again is one or multiple assets. In general, these processes can be represented in the form, where **#** is the edge the process corresponds to:

$$\begin{array}{lccccc} \textbf{\#:} & \text{input asset} & \Longrightarrow & \text{discovery method} & \Longrightarrow & \text{output asset} \\ & \textit{network\_identifier} & & & & \textit{network\_identifier} \\ & \textit{network\_service} & & & & \textit{network\_service} \end{array}$$

Take as an example the usage of zone walking to enumerate hosts in an IPv6 reverse DNS zone [35]. This methodology relies on looking up an IPv6 prefix for an organization and returns the zone's entries as output. Both of these assets are network identifiers, hence it corresponds to Edge 1 from Figure 2.1. In this case, the discovery method can be represented in the following form:

$$\textbf{1:}\ \text{IPv6 prefix} \Rightarrow \text{NSEC3 zone walking} \Rightarrow \text{addresses.}$$

Each paper can include one or more such instances of techniques. Similarly, a paper can also include an asset discovery technique that can be decomposed into multiple different ones, each corresponding to an edge in Figure 2.1. Each sub-technique that results from this decomposition is individually systematized. Tables 2.3-2.6 apply this syntax to all of the literature discussed in Section 2.3. This framework allows us to create a clear overview of different techniques in the literature, and perform meaningful comparisons between them. Furthermore, it enables the easy chaining of techniques, as well as identifying opportunities to combine techniques that have not yet been pursued.

2.3. Asset discovery

In this section we present the systematization of techniques for asset discovery based on the model illustrated in Figure 2.1. While many papers necessarily must first complete the bootstrapping step mapping organizations of interest to network identifiers indicated in Edge 0, this step is invariably application-specific. Hence, we focus the remaining discussion on the more generally applicable steps 1–4 involving the discovery of network identifiers and services. Each edge 1–4 is discussed in the respective subsections 2.3.1–4. The literature is summarized using the systematization syntax in Tables 2.3, 2.4, 2.5, and 2.6, which is discussed in Section 2.3.5.

2.3.1. Discovering Network Identifiers from Network Identifiers – Edge (1)

Network identifiers signal the exposure of a company network. As such, a great part of the recent research has focused on developing new methods or extending well-known techniques to discover resources. We first discuss techniques that use known network identifiers to discover previously unknown identifiers. We group the techniques by the input asset, since they are each processed differently. We will first discuss methods that use domain names as an input, which are primarily concerned with DNS and passive DNS, as well as complications that arise due to the use of Cloud-based Security Providers. Next, we explain how onion URLs are processed differently than domains and IP addresses to discover network identifiers. We then examine efforts using IP addresses (v4 and v6) as input, which integrate closely with that of BGP prefixes, Autonomous System Numbers (ASNs), and the tools and datasets built around them. Finally, we describe techniques specific to IPv6 address discovery. Because full Internet scans are not feasible on IPv6 due to the much larger address space, new methods are required to find active IPv6 addresses. Note that it is common to see techniques which mirror others in terms of inputs and outputs.

When searching for network identifiers for a given organization, one common starting point is the known domain names for the given organization. The purpose of DNS is to resolve a given domain name to the server's IP address. The use of DNS to derive an IP address does not require special tooling. Thus, it provides a simple example of using one network identifier to learn of another network identifier. Knowing the DNS resolution for previous points in time is sometimes beneficial, as an asset may still exist at the former IP address even if a domain does no longer points to it. Taking a domain as input, Tajalizadehkhoob et al. [224] and Vissers et al. [242] suggest the usage of passive DNS databases. These databases contain logs of DNS responses received by different resolvers. The entries in the passive DNS database can be filtered using the provided domain name, revealing the IP addresses that at one point in time were associated with that domain. Passive DNS is not only useful for resolving IP addresses. Liu et al. [131] utilize a passive DNS database to perform a wildcard search to find subdomains related to a domain. Their goal was finding potential shadowed domains: subdomains of a legitimate domain that are under the control of a malicious actor without the domain owner's knowledge. The technique is also relevant for the purposes of asset discovery. Furthermore, these additional subdomains

Table 2.3: Our systematization of papers containing techniques under Edge 1.

Citation	Edge	Input asset	Discovery technique	Output asset	Remarks
Domain Input					
[65, 224]	1	Domain	A/AAAA records from passive DNS	IPv4/IPv6 addresses	
[65]	1	Domain	CNAME record from passive DNS	Domain	
[62]	1	Domain	Extract A and AAAA from DNS ANY queries	IPv4 and IPv6 addresses	
[129]	1	Domain	Use TLD zonefiles to get IDNs	IP	Uses TLD zones to get IDNs & resolve names to IPs
[101]	1	Domain	DNS Zone walking	IPv6	
[49]	1	Domain	DNS logs	IP	
[27, 65, 88, 90, 122]	1	Domain	DNS A/AAAA query	IPv4/IPv6 addresses	
[112]	1	Domain	DNS A-record	IP	Leakage after terminating a DDoS protection service
[65]	1	Domain	DNS CNAME query	Domain	
[242]	1	Domain	Resolve domains in DNS MX, TXT records	IP address	Names in MX/TXT RRs resolve to IP addr. behind a CBSP
[242]	1	Domain	Passive DNS search	IP address	Data from passive DNS may reveal an IP hidden by a CBSP
[13, 64, 153]	1	Domain	Passive DNS search	IP address	
[131]	1	Domain	Passive DNS wildcard search	Subdomains	For example, *.domain.com
[194]	1	Domain	ZDNS	IP address	
[202]	1	Domain	Satellite tool	IP addresses	The tool discovers IP addresses of used CDN infrastructure
[48]	1	Domain	DomainScouter tool	Domains	Identifies IDNs; Add. checks needed to confirm ownership
[242]	1	Domain	Search domain in IP/SSL certificate pair collection; domain from certificate that resolves differently exposes real IP.	IP address	Discovers the real IP that is hidden by a CBSP
[210]	1	Domain	WHOIS	AS	
[95]	1	Domain	DNS A queries from geographically distributed vantage points	IP addresses	Different IP addresses can be returned for the same domain if websites are hosted by CDNs
[143]	1	Domain	CARONTE tool	IP address	Comb. of techniques to identify IPs of hidden services
IPv4 Input					
[154]	1	IPv6 addresses	6gen algorithm	IPv6 addresses	Generates IPv6 Addr. candidates from IPv6 hitlists
[85]	1	IPv6 addresses	Entropy/IP	IPv6 addresses	Generates IPv6 Addr. candidates from IPv6 hitlists
[22, 42, 52, 114, 173]	1	IP address	CAIDA prefix-to-AS data mapping (pfx2as)	AS	
[64, 90]	1	IPv4/IPv6 pr.	CAIDA prefix-to-AS	AS	
[176]	1	BGP prefix	CAIDA AS-to-organization mapping dataset	ASes	Identifies ASes that belong to the same organization
[178]	1	IP address	Censys data query	AS	
[178]	1	IP address	MaxMind AS data query	AS	
[91]	1	IP address	CAIDA topology dataset search	IP address	Discovers IP address belonging to the same router
[113]	1	IP address	OpenINTEL historical active DNS dataset	Domain	Uses historic data; records could be out of date
[42, 65]	1	IP address	Reverse DNS dataset search	Domain	
[177]	1	IP address	Zmap	Domain	
[90, 91, 175, 178]	1	IP address	Reverse DNS query	Domain	
[93]	1	IP address	TreeNET	IP prefix	
[92]	1	IP address	WISE	IP prefix	
[22, 98, 123]	1	IP address	RouteViews/RIPE RIS data	BGP prefix	
[51, 66, 84, 176, 251]	1	IP address	RouteViews/RIPE RIS data	AS	
[225]	1	IP address	MaxMind WHOIS dataset search	IP addresses	The output is the IP range belonging to the organization
[174]	1	IPv6 address	UAv6 technique	IPv6 address	UAv6 is an alias resolution technique for IPv6
[65]	1,2	Domain names	Multiple probes to determine hosting	Dom. names/IP addr./srv.	Identifies multiple domains hosted on the same IP
IPv6 Input					
[81]	1	IPv6 prefixes	Algorithm in paper	IPv6 addresses	Enumerates IPv6 reverse DNS entries
[80]	1	IPv6 prefixes	Algorithm in paper (using DNS NXDOMAIN)	IPv6 addresses	Enumerates IPv6 reverse DNS entries
[35, 101]	1	IPv6 prefix	Reverse zone scanning NSEC3	IPv6 addr./networks	The output is a list of IPs in the same IPv6 zone
[24]	1	IPv6 addresses	Algorithm in paper	IPv6 addresses	Generates IPv6 Addr. candidates from IPv6 hitlists

may in turn uncover additional IP addresses.

Domain owners and organizations sometimes try to protect their websites from threats, notably denial-of-service attacks, by using a Cloud-based Security Provider (CBSP) that acts like a reverse proxy. In this scenario, the domain name is directed to an IP address under the control of the CBSP. The CBSP will then proxy the HTTP(S) traffic to the source IP address, where the web service is truly hosted, by using the domain in the HTTP Host header. Plain DNS is not helpful in these cases for discovering "true" source IP addresses. Vissers et al. [242] discuss several methods to discover source IP addresses of web servers which use a network identifier as an input, including querying a passive DNS database as discussed previously. Hiding the source IP address is problematic for protocols that do not contain any host information, such as FTP and SSH. Instead of connecting directly through an IP address, administrators can opt to create a subdomain that resolves directly to the source or "real" IP address. A method mentioned by Vissers et al. connects to a subdomain to find such a service, and subsequently discover the real IP address of a server. Another method proposed by the authors trawls an Internet-wide collection of SSL certificates. Domains using CBSPs will frequently have multiple certificates for the same domain hosted by different IP addresses, one of which ends up being the true source IP. Yet another method proposed by Vissers et al. involves DNS servers. Sometimes, DNS records such as MX or TXT records still reference the source IP address that the CBSP is hiding [235]. By requesting these MX and TXT records, the true source IP address for the web server can often be revealed even when the A record is pointed to the CBSP [242]. Finally, Scott et al. develop a methodology and tool to capture the IP addresses of Content Delivery Network (CDN) deployments used for input domains [202].

Onion URLs are network identifiers used to host Tor hidden services while obfuscating the server's true IP address. Matic et al. introduce CARONTE, an automated tool to discover location leaks that betray the source IP of Tor hidden services [143]. The automated tool uses several techniques to find potential Onion URL to IP mappings. The tool extracts URLs, email addresses and IP addresses from the page. The URLs and domains are resolved to collect more candidate IP addresses. Additionally, CARONTE extracts unique strings from the page and performs search engine queries containing that string. The domains of the pages that contain this string are also resolved and the IP addresses are added to the candidate IP collection. The tool also looks for potential identifiers in the leaf certificate of HTTPS hidden services. The final techniques utilize a certificate repository. Websites and hidden services hosted on the same server may share a certificate or public key. CARONTE searches the certificate repository to see if the hidden service's certificate or public key matches any other websites, thus creating a leak. It then validates the potential IP addresses by visiting each one directly without relaying through Tor nodes.

While IP addresses are often outputs of network identifier discovery methods, they can also be used as inputs. In a simple role-reversal, Gharaibeh et al. [91] and Cangialosi et al. [42] utilize reverse DNS to obtain domain names from IP addresses.

IP addresses may also be mapped to their BGP prefixes or used to discover which Autonomous System Number (ASN) it is routed through. Jonker et al. [114] seems to collect BGP data and use an unspecified process to map each IP address to the most specific BGP prefix containing the address. For those not interested in manually collecting BGP data, RouteViews is a project by the University of Oregon which uses probes placed in different places throughout the Internet to track BGP information and makes this information publicly available [192]. RIPE RIS is a similar tool for tracking how the Internet routes traffic [190].

Both RouteViews and RIPE RIS have a variety of use cases for finding network identifiers from another identifer, which are outlined below. Benson et al. [22], and Krenc et al. [123] utilize RouteViews as well as RIPE RIS to find the announced BGP prefixes that correspond to given queried IPv4 addresses. Chung et al. [51] and Foremski et al. [84] also use the RouteViews dataset to derive the ASN from an IP address. Yeganeh et al. [251] utilizes RouteViews in addition to RIPE RIS to accomplish the IP address to ASN mapping. CAIDA uses the RouteViews data to derive a dataset which maps BGP prefixes (for IPv4 and IPv6) to their respective ASNs, which saves on labor for some users; this dataset is commonly called pfx2as [14]. Chung et al. [52] utilizes CAIDA's pfx2as dataset to find BGP prefixes from IPv4 addresses. Then they use that same dataset to map the prefix to the ASN. Benson et al. [22] and Jonker et al. [114] also use CAIDA's pfx2as dataset with the BGP prefixes they previously obtained from IP addresses to determine to which autonomous system number (ASN) they each belong. Cangialosi et al. [42], Padmanabhan et al. [173], also use CAIDA's pfx2as dataset to map IP addresses directly to ASNs.

As a last note on Internet topology discovery, Gharaibeg et al. [91] and Czyz et al. [62] utilize CAIDA Ark data to find router interface IP addresses. CAIDA Ark is a measurement platform which collects traceroute data for random portions of the Internet among other measurements. Researchers may use this traceroute data to discover router IPv4 and IPv6 addresses. While this method of discovery is not as targeted to an organization as an active scan of an organization's known address space, one can still feasibly map a given router to an organization if the organization's address space is known.

Lastly, we discuss methods which are targeted directly at IPv6 addresses. Fiebig et al. [80, 81], Beverly et al. [24], Hu et al. [101], and Borgolte et al. [35] all look specifically at IPv6 hosts and use IPv6 addresses or (individual) IPv6 networks as inputs or outputs. The methods developed by Fiebig et al. [80, 81] and Borgolte et al. [35] take a more technical approach to the discovery process than previously discussed work. The authors develop a technique to walk an IPv6 network's reverse zone using either protocol level features of

DNS [80, 81, 101] or by leveraging peculiarities of hashing in DNSSEC [35, 101]. The work by Fiebig et al. utilizes differences in responses between DNS labels that do and do not have children to prune the reverse zone search tree (NXDOMAIN). Similarly, Borgolte et al. [35] leverage DNSSEC for exploring zones. DNSSEC allows operators to sign DNS responses, in turn allowing clients to verify if a DNS response has been tampered with. The issue here is the ability to prove the non-existence of records. DNSSEC does this by returning a non-existence record listing the (alphabetically) previous and next entry in the zone (NSEC), or hashed versions of these records (NSEC3). Borgolte et al. leverage the structured nature of the IPv6 reverse zone to break NSEC3 hashes in reasonable time, thereby allowing exploration of reverse zones. The network identifiers discovered by these methods are a list of allocated IPv6 addresses. Foremski et al. [85] use entropy analysis and statistical modeling from a known set of IPv6 addresses to create a tool that can generate a list of additional possible IPv6 addresses. Related to the earlier efforts to directly discover IPv6 addresses, Murdock et al. [154] propose a technique that utilizes a so called "hitlist," *i.e.*, a list of allocated IPv6 addresses as, *e.g.*, discovered by one of the previous techniques to generate further IPv6 addresses that *might* be active based on the allocation pattern observed in the hitlist.

2.3.2. DISCOVERING NETWORK SERVICES FROM NETWORK IDENTIFIERS – EDGE (2)

The techniques that correspond to this edge can be split up into two principal categories. The first category consists of techniques that typically leverage databases such as (passive) DNS and WHOIS by querying them, either actively or passively. It is one of the most straightforward methods of identifying network services accessible through a network identifier. Its range of network services is generally restricted to the types of records made available through those protocols, such as name servers and mail servers (DNS NS and MX records, respectively). These are well established protocols, and the data obtained from these sources is often standardized or predictable. In the specific case of passive databases, such as passive DNS, the data has been preprocessed, and researchers that use this data benefit from that. They also enable researchers to conduct longitudinal studies due to the large amounts of data that is collected over time. The second category is composed of Internet-wide scanning techniques, which are necessary to address the aforementioned shortcomings. The techniques described in this category utilize existing scanners, or create custom tools or algorithms that identify network services. These techniques are necessary because of the previous category's limited scope in terms of discoverable network services (essentially only web, mail, and name servers). As these techniques are not necessarily

Table 2.4: Our systematization of papers containing techniques under Edge 2.

Citation	Edge	Input asset	Discovery technique	Output asset	Remarks
Database Queries					
[114]	2	Domain	DNS A, AAAA, NS queries	Web server, name server	
[225]	2	Domain	Custom crawler tool	CMS	IDs cPanel, Plesk, DirectAdmin, & Virtualmin instances
[65, 71, 86, 100, 122]	2	Domain	DNS MX record query	Mail server	
[65]	2	Domain	MX record from passive DNS	Mail server	
[65]	2	Domain	NS DNS queries	Name server	
[65]	2	Domain	NS record from passive DNS	Name server	
[134]	2	IP address	Open Resolver Project dataset search	Name server	The dataset contains IP addresses of open resolvers
[132]	2	Domain	WHOIS statistical parsing model	Name servers	WHOIS parsing using a statistical model
[115]	2	IP address	DNS query for hostnames under own domain	Name server	Identifies open resolvers through successful name resolution
Internet-Wide Scanning					
[25]	2	IP address	Censys/Shodan port 80 and 8080 scans	Proxy server	Identifies MikroTik routers with enabled proxy
[247]	2	Domain	Dmap	Web server	Uses DNS A/AAAA queries to discover the network service
[247]	2	Domain	Dmap	Mail server	Uses DNS MX queries to discover the network service
[225]	2	Domain	Port 22 banner scan	SSH server (version)	
[71]	2	IP address	Port 25 scan	Mail server	The scan on this port aims to find a running SMTP service
[137]	2	Domain	Port 443 scan for DNS-over-HTTPS paths	Name server	Discovers DNS server that offer DNS-over-HTTPS
[62]	2	IPv4/v6 pairs	Server sibling detection alg. from [23, 200]	Multiple services	Identifies IPv4/IPv6 multihoming
[250]	2	IP address	Angry IP scanner	Multiple services	Identifies protocols such as SSH and Telnet on a host
[70]	2	IP address	ZGrab	Multiple services	Application scanner with banner grab functionality
[236]	2	IP address	ZMap scan on port tcp/7	Echo server	Identifies servers running ECHO using TCP SYNs
[218]	2	IP address	Zmap scan on port tcp/21	FTP server	
[137]	2	IP address	ZMap scan on port tcp/853	Name server	Discovers name servers that offer DNS-over-TLS
[163]	2	Domain	Custom Java tool	Web server	Tests HTTPS connectivity
[135]	2	IP address	ZMap scan on port tcp/8080	Bytecoin client	
[135]	2	IP address	ZMap scan on port tcp/8333	Bitcoin client	
[3, 88, 199, 219]	2	Domain	ZMap scan on port tcp/443	Web server	
[125]	2	Domain	Censys cert scan	Web server	
[178]	2	IP address	ZMap scan on port 53	Name server	Send DNS A query to verify a port is offering a DNS service
[3]	2	IP address	ZMap UDP scan on ports for IKEv1 and IKEv2	VPN	Scan ports for IPsec IKE protocols to find VPNs
[88, 89]	2	IPv6 address	ZMapv6 scan on tcp/80,443, and udp/53,443	Web server, name server	ZMapv6 is ZMap with IPv6 scanning capabilities
[128]	2	IP address	ZMap for DNP3, Modbus, BACnet, Tridium Fox, and Siemens S7 protocols	Industrial control system	
[19]	2	IP address	ZMap on ports 25, 110, 443, 465, 587, 993, 995	Web server, Mail server	Ports correspond to email related services
[181]	2	IP address	Censys port 443 scan data	Web server	
[194]	2	IP address	ZMap with custom QUIC extension	QUIC services	
[12]	2	IP address	TCP ping for ports used by various protocols	Multiple services	
[152]	2	IP address	ZMap scan; analyze sigs. in fetched banners	Honeypot	Discovers/identifies honeypots; (From IEEE/IFIP INM)
[31]	2	IP address	Search IP in botnet population	*Bot services*	Creates a botnet churn simulator
[58]	2	IP/Dom./URLs	Scans of S3 buckets	S3 buckets	Identifies misconfigured/public Amazon S3 buckets
[156]	2	IP address	Active probes	Service (reverse proxy)	Identifies reverse proxies using TCP fingerprinting
[214]	2	IP/MAC addr.	Layer 2 and 3 timing probes	SDN service	Uses timing attacks to identify SDN services
[65]	1,2	Domain names	Multiple probes to determine hosting	Dom. names/IP addr./srv.	Identifies multiple domains hosted on the same IP
[105]	2,3,4	IP/MAC addr.	nmap/arpscan + HW side-channels	IP/MAC addr.	Identifies HW colocation on public cloud platforms

restricted by the specifications of existing protocols, they widen the search space and give researchers more freedom to find the network services necessary for their research. This is important for asset discovery, as there exists a plethora of additional network services that can be running on an organization's network, with many more that will be developed in the future. Techniques corresponding to this edge can be found in Table 2.4. We start by discussing the first category of techniques.

As noted in the previous section, DNS is used extensively in Internet measurement research. It also proves useful in identifying network services. Such techniques correspond to the first category. Many authors use DNS queries to discover network services. Jonker et al. [114] and Dell'Amico et al. [65] query name servers using a certain domain name and extract A, AAAA, and NS records to identify web servers and name servers. Similarly, Foster et al. [86], Durumeric et al. [71], Dell'Amico et al. [65] and Kountouras et al. [122] query for MX records in order to identify mail servers.

Some DNS servers provide recursive name resolution services to any user on the Internet, either on purpose or by accident. Such servers are called *open resolvers*. Liu et al. [134] describe a method that queries Open Resolver Project data [170] using an IP address to discover whether an open resolver is accessible through that address.

Querying WHOIS is also widely employed, but automatic parsing of such data remains an issue due to the lack of format standardization [132]. Liu et al. develop a statistical model for parsing WHOIS data that takes a domain as input, enabling users to automatically extract the addresses corresponding to a queried domain's name server [132]. The authors report an accuracy for parsed fields of over 99% for `com` domains. For TLDs outside of `com`, the accuracy decreases, although a specific number is not given. For instance, the model mislabels 16 out of 127 fields in `emheartcu.coop`'s WHOIS record.

The second category will be discussed below, which includes Internet-wide scanners and other custom tools. Most of the papers discussed here specifically involve Internet-wide port scanning and use the same scanner to discover network services, namely ZMap [72]. Since its introduction, ZMap has been ubiquitous in Internet measurement research. ZMap is a network scanner specially designed to enable fast Internet-wide scans [72]. The authors highlight three optimizations that allow it to perform such scans at a much faster rate. Firstly, ZMap does not throttle its transmission rate to avoid saturating the scanned or scanning network. Instead, ZMap sends messages as quickly as the network interface card permits. Furthermore, ZMap does not maintain state for each connection to track its scanning progress. Since the goal is to scan random portions of the address space, the scanner avoids storing previously scanned addresses by making use of randomly permuted IP addresses to select targets. It tracks connection timeouts by embedding state information in packet fields. Finally, ZMap opts not to retransmit lost packets. While this results in a 2% loss of network coverage, the authors view this to be an insignificant amount "for typical research applications." [72]

Researchers use this tool to identify network services accessible through domains or IP addresses. Adrian et al. [3], Springall et al. [219], and Scheitle et al. [199] scan port 443 on a host to identify web servers that support HTTPS. Durumeric et al. use ZMap scans on port 25 to find mail servers running SMTP [71]. Adrian et al. also use ZMap's UDP functionality to probe ports for IKEv1 [96] and IKEv2 [116] to find IPsec VPNs. Loe and Quaglia find hosts running Bitcoin and Bytecoin clients by using ZMap to scan ports 8333 and 8080, respectively [135]. Springall et al. also use ZMap to scan port 21 in order to to discover FTP servers [218]. Lu et al. scan port 853 to discover DNS servers that offer DNS-over-TLS [137]. Morishita et al. use ZMap to scan IP addresses on a large collection of ports. From the responses to the scans, they extract the banners and create signatures that allow them to accurately detect and discover different versions of honeypots [152]. Finally, Gasser et al. use ZMapv6 in their study, a variant for scanning IPv6 addresses. They perform TCP scans on ports 80 and 443, as well as UDP scans on ports 53 and 443 to discover web servers (potentially using the QUIC protocol [111]) and DNS servers [89].

While ZMap is used for network service discovery using port scans, it is not able to discover any specifics about the network service. Thus, if such custom functionality is desired, users need to develop a custom ZMap toolchain of their own. For instance, Adrian et al. implemented the SSH protocol in the ZMap toolchain to examine the version and Diffie-Hellman cipher usage in SSH servers, and also added `DHE` and `DHE_EXPORT` ciphers to discover HTTPS servers using TLS that support these ciphers [3]. Similarly, Scheitle et al. first use ZMap to discover HTTPS servers on port 443, after which they employ a custom TLS scanner to examine the corresponding TLS certificates. While Springall et al. discover FTP servers using standard ZMap functionality, they needed to implement a custom FTP enumerator to perform a connection and extract information from the server.

Durumeric et al. developed another scanning tool called ZGrab [70], which is an application scanner that scans ports on a host to discover what services are running on said host. At the time of publishing, ZGrab supported scanning of IP addresses in order to identify services such as HTTP, HTTP Proxy, HTTPS, SMTP(S), IMAP(S), POP3(S), FTP, CWMP, SSH, and Modbus by means of protocol handshake initiations. This tool is extensible, meaning that custom protocols can be added if necessary. ZGrab has since been deprecated and replaced by its successor, ZGrab2 [231]. As an improvement over ZGrab, ZGrab2 allows users to scan targets on multiple ports using multiple protocols.

Wullink et al. develop a scanning tool of their own that discovers network services [247]. Their tool, Dmap, takes domain names as input and discovers whether any DNS, HTTP, TLS, or SMTP services are accessible through that domain (or IP corresponding to that domain), using three crawlers. The HTTP crawler attempts to connect the website hosted on the domain, thereby discovering a web server. Apart from A, AAAA, and TXT records, the DNS crawler extracts MX records to discover the mail server corresponding for the input domain. The SMTP crawler attempts to connect to the discovered mail server address through both IPv4 and IPv6 addresses, if available.

Censys [43] and Shodan [208] are search engines that employ such Internet-wide network and application scanners to collect data about devices that are publicly accessible on the Internet. In fact, ZMap itself is the scanner behind Censys [70]. Bijmans et al. use Censys and Shodan scan data on ports 80 and 8080. By querying the data with IP addresses, they discover whether that IP address belongs to a router (specifically the MikroTik brand) with its proxy functionality enabled [25].

Other researchers opt to create custom tools tailored to their own specific needs. Tajalizadehkhoob et al. discover the presence of admin panels on a server by taking a domain and crawling the ports that are usually associated with cPanel, Plesk, DirectAdmin, and Virtualmin [225]. To improve measurement performance, they instructed the crawler to navigate to the URLs that are often used by these admin panels (e.g., `/panel/`).

The following works use techniques from the two different categories to achieve a goal. Czyz et al. utilize data from Internet-wide DNS ANY queries to extract A and AAAA records corresponding to the same domain name [62]. By probing the obtained IPv4 and IPv6 addresses, they identify the network services accessible through these addresses (e.g., SSH, Telnet, HTTP, among others). The authors found many IPv6 misconfigurations, where services were accessible through the IPv6 address but not the IPv4 address, presumably by accident. The implication of this finding is that you can determine what network services are offered on an IPv4 address by observing a multi-homed IPv6 address.

However, determining whether an IPv4 and IPv6 address pair actually point to the same server is not straightforward. Several authors have developed techniques for making this link. Beverly and Berger developed a TCP-layer fingerprinting approach that probes IPv4/IPv6 address pairs and utilizes TCP Options signatures and TCP timestamp skew to determine whether the address pair points to the same server [23]. Scheitle et al. developed a similar algorithm, but taking more features into account, such as network latency, TTL values, as well as other calculated features [200]. By combining either of these approaches with the findings from Czyz et al. [62], it is possible to discover the purpose of (and, therefore, the network services offered by) an IPv4 address, even though the service is not accessible through the IPv4 address.

Table 2.5: Our systematization of papers containing techniques under Edge 3.

Citation	Edge	Input asset	Discovery technique	Output asset	Remarks
Reverse Information Flow					
[242]	3	Mult. services	Access net. service without explicit host info	IP address	Real IP leaked if a service on a subdomain behind CBSP lacks protocol information
[97]	3	Name server	Query open resolver for `v6only` zone	IPv6 address	Identifies IPv6 addresses of open resolvers
[105]	2,3,4	IP/MAC addr.	Probing (nmap/arpscan), HW side-channels	IP/MAC addr.	Identifies HW colocation on public cloud platforms
Misconfigurations					
[130, 242]	3	Name server	Dictionary attack	Domains	Only the domains in the used dictionary file are discovered
[130, 242]	3	Name server	DNS zone transfer	Domains	Only works if the DNS server has enabled zone transfers
[242]	3	Web server	Trigger outbound connection from web server	IP address	Return traffic not going through CBSP exposes the real host

2.3.3. DISCOVERING NETWORK IDENTIFIERS FROM NETWORK SERVICES – EDGE (3)

Edge 3 encompasses techniques that leverage information from network services, such as DNS records, to discover network identifiers, such as IP addresses. Notably, this is the only edge that uses a higher level attribute to infer a lower level one, as illustrated in Figure 2.1. Some of the techniques discussed here are similar to those employed in Edge (1), except that the flow of information could be reversed, or a new flow may result from a misconfigured service. For example, more than simply resolving domain names to IP addresses (Edge (1)), DNS can be used to discover new network identifiers (Edge (3))–a difference we will illustrate below. Edge 3 techniques typically seek to discover *origin* IP addresses (i.e., the real network location of a service situated behind a content-delivery network or other cloud-based service provider), and do so by leveraging potentially misconfigured services and/or their byproducts (e.g., files these services create).

In their comprehensive study of techniques to bypass cloud-based service providers (CBSP), Vissers et al. describe several Edge (3) techniques, as they seek to expose the origin IP address of a CBSP-protected domain. An attacker with knowledge of the origin IP address can in turn completely bypass any CBSP protection, as traffic sent to that address would not be routed through the CBSP security infrastructure [242]. The first technique involves leveraging DNS records associated to services that may be running in the host, such as mail servers. For instance, if a CBSP only forwards HTTP traffic, SMTP will need to establish a direct connection with the mail server, thus leaking the origin IP address. Rather than just performing identifier-to-identifier resolution (domain to IP), a network service (SMTP) here provides us with a new identifier (origin IP). A second method described by both Vissers et al. [242] and Liu et al. [130] uses the zone transfer functionality of (misconfigured) DNS servers (network service) to obtain zone records (network identifiers). This is possible when an attacker pretends to be a secondary DNS server and asks a main DNS server for zone records. Unless the main DNS server has restricted zone transfers, it will oblige and send the records to the attacker.

A third set of techniques rely on potentially specific misconfigured services that may exist in the host that leak the IP addresses through a variety of ways. For instance, Vissers et al. note how "sensitive files" (e.g., verbose error pages or log files, such as `phpinfo()` files) revealed by misconfigured services could cause a leak [242]. Similarly, non-web protocols may also be a concern when improperly handled. Some cloud-based service providers act as a reverse proxy and rely on HTTP `Host` headers to separate requests for different clients. Hence, protocols that, unlike HTTP, do not contain `Host` information (such as SSH) may not be supported. Then, administrators may elect to create subdomains for non-web protocols which directly resolve to the origin IP address – effectively bypassing voluntarily any CBSP protection. A dictionary-based attack of common subdomains could then, easily be used to retrieve the origin IP address [242]. Services that trigger outbound connections can cause similar issues, as the connections may not be routed through the CBSP and leak the origin IP address.

2.3.4. Discovering Network Services from Network Services – Edge (4)

Table 2.6: Our systematization of papers containing techniques under Edge 4.

Citation	Edge	Input asset	Discovery technique	Output asset	Remarks
DNS Based Discovery					
[155]	4	Name server	Authoritative name server discovery technique	Name server	Identifies auth NS from apex and NS records in the child
[6]	4	Name server	DNS query for hostnames under own domain	Name server	Identifies outbound addresses of (open) resolvers
[7]	4	Name server	DNS query for hostnames under own domain	Name server	Identifies open resolvers sharing a cache
[119]	4	Mail server	Send email to inexistent dst. in target domain	Name server	Uses email bounces to identify recursive resolvers of MTAs
[198]	4	Mail/DNS srv.	Send email from domain with auth. NS control	Name server	Identify recursor used by MTA due to triggered SPF check
Other					
[179]	4	Web server	Port scans using server-side requests	Multiple services	
[237]	4	Web server	`Prober` bash script tool	Multiple services	Discovers protocols on a server with ALPN/NPN
[105]	2,3,4	IP/MAC addr.	nmap/arpscan + HW side-channels	IP/MAC addr.	Identifies HW colocation on public cloud platforms

As described in previous sections, DNS is fundamental to asset discovery. Similar to Edge (3), we will describe several techniques based around DNS server functionality. In fact, most of the techniques in this section are DNS-based. While DNS techniques under Edges 1 and 2 focus on asset discovery through DNS queries (e.g., A/AAAA, MX, NS), the techniques described here involve the inner workings of the DNS protocol and the network service itself. It is often of interest to discover an organization's internal infrastructure of DNS resolvers. Due to its fundamental role, DNS security remains an important topic both in industry and research communities. The other techniques mentioned in this section cannot be grouped into a single category, as they use different methods and discover different assets. We start by describing the DNS-based techniques.

Al-Dalky and Schomp [7] devise a method to discover name servers by leveraging how the servers collaborate during resolution. The authors set up two instrumented hostnames and send two queries to the authoritative name server of their own domain. If the resolver is part of a pool of collaborating name servers, the two queries may arrive at the authoritative name server from two different IP addresses, revealing a previously unknown name server.

Al-Dalky et al. [6] scanned the IPv4 address space to discover DNS resolvers by sending crafted DNS requests for hostnames from their own domain and recording the queries arriving at their experimental authoritative server. When a DNS resolver receives a request with domain names specified, it needs to query name servers unless it is already cached from serving a prior request. By setting up researchers' own name servers, the researchers are able to get the message out of DNS resolvers, allowing them to understand how DNS resolvers work. In this case, they also configured the name servers to only respond to queries with the EDNS0-Client-Subset (ECS) option set to discover DNS resolvers using ECS.

Schomp et al. [201] developed a set of methods to discover client-side DNS infrastructure. Similar to Al-Dalky et al. [6], they also registered their own authoritative domain and deployed their own authoritative DNS to probe the IP address space by sending crafted DNS requests to potential DNS resolvers and receiving the data from DNS resolvers to their name servers. Again, the DNS requests they send out attempt to resolve various hostnames within their own domain, so the DNS recursive resolvers, which are out of their control, will query the name servers as specified in the original DNS requests, arriving to the name servers the researchers have control of. The difference is that they studied the IP addresses arriving at their authoritative name space to find recursive DNS egress resolvers as well as open DNS ingress servers to gain a deeper understanding of different DNS resolvers.

Klein et al. [119] used a similar probing strategy to Schomp et al., but furthered the device discovery process to find hidden caches used in DNS infrastructure. Apart from directly sending DNS requests to open resolvers, they also used web browsers and mail servers to generate DNS requests. The former was achieved by embedding a script in an ad network page and attaching it to a static URL. The latter involves sending emails to non-existing email addresses in the target domains. To be compliant with the SMTP standard, email servers are required to generate a Delivery Status Notification (DSN, or bounce) message to the originator, and the server has to perform DNS resolution to generate the bounce message. The responses from DNS resolvers sent to researchers' name servers contain the data for discovering DNS services.

Scheffler et al.[198] leverage the functionality of both mail and name servers. The Sender Policy Framework (SPF) was developed to combat spam by verifying the identity of senders. In SPF, valid mail transfer agent (MTA) IP addresses are listed in a TXT record in the organization's domain. The authors set up their own domain and sent out emails as part of their measurement methodology. When the emails arrive at an MTA, they trigger an SPF check. The MTA's resolver then queries your authoritative name server, thereby revealing the IPs of those DNS resolvers.

Naab et al. [155] implement a custom DNS resolver that uses QNAME minimization to discover all of the available authoritative name servers within a queried zone. QNAME minimization [36] was developed to improve privacy by not sending the full original DNS query name (QNAME) in each query. Instead, name servers are only queried for the domain level they are responsible for. Their custom DNS resolver queries all available authoritative name servers for each zone in the domain using three different methods. It extracts name servers from the name in delegation (NS) records and from the IP addresses contained within glue (A) records. Lastly, name servers are extracted from all other NS records in the zone apex.

Pellegrino et al. [179] uses the server-side request functionality of a website running on a server. One can provide to a web server, for instance, a request containing the URL and port that one wants to probe. The web server then performs the server-side request to the provided URL and port. When the web server fails to parse the probed server's response as HTTP, it returns an error message. These error messages can in turn reveal which services are running on the probed server, such as the software running on the probed port or which service is running on the probed port.

Finally, Varvello et al. [237] create a bash script Prober to scan for multiple network services. Prober uses OpenSSL to perform ALPN and NPN negotiations on web servers. It does this to discover numerous protocols that might be announced by the server.

2.3.5. Systematization Summary

Tables 2.3 to 2.6 contain the systematization of the literature performed for this survey. Note, that one paper might occur in multiple lines if it uses multiple techniques, and multiple papers may appear on the same line, if they use the same asset discovery technique. The *Edge* column corresponds to the edge number in Figure 2.1. Furthermore, the techniques presented in the table follow the systematization syntax described in Section 2.2.3.

A simple, but interesting, observation is the significant underrepresentation of *network service-to-network identifier* and *network service-to-network service* (Edges (3) and (4)), when compared to the other two types of asset discovery techniques. Much of the surveyed literature deals with Internet measurement. Given its nature, the initiation of this type of research necessitates basic network identifiers, such as IP addresses and domains. This logically leads to an overrepresentation of asset discovery techniques that take a network identifier as input.

The techniques used in Edge (1) can be roughly grouped into three groups: discovery using 1) passive data sources, 2) functionality of existing technologies and infrastructure, and 3) novel algorithms making use of existing technologies. CAIDA prefix-to-AS and RouteViews data seem to be the standard choice for discovering BGP prefixes and ASs. Both passive and active DNS are widely used in academic research.

Edge (2), discovering network services using network identifiers, is the second overrepresented edge. A significant amount of the discovery techniques described in the literature revolve around the network scanner ZMap.

Edges (3) and (4) use network services to discover network identifiers and more network services, respectively. Most of the techniques associated with these two edges involve the usage of flaws in configuration or the technology itself.

2.4. Applying the Asset Identification Framework to Case Studies

Having mapped the literature onto the various edges of the asset identification framework, we now illustrate its use by applying that framework to several case studies. In the previous section, we extracted the particular edge that researchers used in their work. Here, we construct sequences of edges into paths indicating different asset-discovery processes from beginning to end.

External estimation of enterprise cyber risk A slew of firms, such as Security Scorecard, QuadMetrics, Bitsight, and CyberCube, now offer external assessments of enterprise cyber risks. Their tools provide quantitative scores based on externally observed network characteristics. For these scores to be valid, they first need an accurate asset inventory for the target organizations, which they have to obtain without the organization's cooperation.

While the methods employed by these firms are proprietary, we can glean some insights into their approach by studying the academic papers published by the founders of QuadMetrics [134, 253]. For the bootstrapping phase (Edge (0)), the authors used data from regional Internet registries to identify all the organizations asserting ownership over IP address space. From there, they associate the advertised IP addresses with each organization. They supplement this by searching for the organization's domain name on a search engine (Edge (0)), then resolving the domain to its IP address and identifying the subnet on which the IP address resides (Edge (1)). This in turn identifies more IP addresses belonging to the organization.

Because the researchers' goal is to identify network misconfigurations [34, 67] that could lead to a security breach, they identified network services where misconfigurations could readily be identified. This included identifying open DNS resolvers (Edge (2)) that could be used in DDoS amplification attacks. They also went beyond the identification of services to identifying weaknesses, including BGP misconfigurations and problematic HTTPS certificates, and open SMTP relays.

Take-away: We now have a standardized view into what these researchers did to identify assets. Others carrying on similar efforts could improve upon their findings by utilizing more Edge (1) and (3) actions from the Tables 2.3 and 2.5. The authors' choice of Edge (2) actions reflects their underlying goal of identifying security misconfigurations that may be predictive of future breaches, so it is not as clear whether the efforts would benefit from identifying additional network services.

Understanding IPv4/IPv6 security configuration inconsistencies A more research-driven problem is an investigation of configuration inconsistencies between IPv4 and IPv6 in dual-homed hosts. With the global IPv4 address exhaustion, more and more organizations start to deploy IPv6 on their networks. However, this opens the door to simple misconfigurations, where access policies differ between IPv4 and IPv6, as—depending on the platform—firewall configurations have to be configured in other places than for IPv4 [67].

To investigate this issue, we can either start with the exploration of IPv4 or IPv6 addresses of assets for all organizations connected to the Internet (Edge (0)). As soon as we identified IPv4 or IPv6 addresses, we can start "matching," *i.e.*, identifying corresponding IPv4 or IPv6 addresses (Edge (1)). This can be done by resolving the reverse DNS entries of the identified addresses, and subsequently resolving the corresponding A/AAAA RRs of the returned hosts (Edge (1)). As soon as we identified a set of dual homed hosts, we can explore the available services on these hosts (Edge (2)). This service exploration can then be used to further refine our network resources list, *e.g.*, by checking that services running on the identified IPv4 and IPv6 addresses present the same banners (Edge (3)).

Take-away: By using the proposed framework, we can formalize the discovery process of assets. Especially for the procedure of identifying multi-homed hosts, it provides clarity on *what* we want to discover using *which* means. More fundamentally, the asset discovery procedure as outlined in this paper provides a guideline from which researchers can start to design their discovery procedure. The seemingly obvious choice of what to start (IPv4 or IPv6 addresses) becomes more evident as an *and*-choice, not an *either-or*-choice. Furthermore, the option to utilize services to further refine the selection of multi-homed hosts seems obvious when using our framework. Nonetheless, earlier work did not directly utilize this option when conducting such a study, and only focused on address discovery using DNS (Edges (0) and (1)) [62].

Discovering IoT services Since `Nmap` creator Gordon "Fyodor" Lyon scanned "the entire Internet" in 2008 [140], replicating this feat has become commoditized. Nowadays, we have services and tools that literally save days of scanning networks, either by speeding up the process or simply providing the list of open ports as a service. These services have recently specialized in the identification of Internet-enabled devices that are openly accessible, the so-called, Internet of things (IoT). Based on similar principles as `Nmap`, multiple online search engines have emerged providing identification of services running on IoT devices such as Shodan.io, Zoomeye.org and Fofa.so.

The starting discovery technique of these engines is quite simple, *i.e.*, they take IPv4 addresses and identify any service running on that address (Edge (2)). Once all these services are identified they store any metadata related to the open service(s). After the initial exhaustive discovery data storage are completed, these tools allow to search for any Internet device, filtering by date, location, ports, protocols, operating system, and much more. All these different network resources and services are indexed in an online search engine which allows the researcher to discover resources and services over all edges of our framework. Taking Shodan [208] as an example, we can map the network resources of any organization by entering its name (Edge (0)). This search will identify a set of network resources that can be used to identify more network resource (Edge (1)) and/or network services (Edge (2)) simply clicking on the identified resources. Shodan also allows searching for a particular network service (e.g., `UPnP`) which will discover additional network resources (Edge (3)) and network services (Edge (4)) that have that port open. Moreover, as Shodan also stores banner and protocol communication information for openly accessible network resources, it allows characterizing network services and thus identifying IoT devices.

Take-away: By scaling up a simple tool introduced in 2008, we can now identify services running in any openly accessible IoT device. Unfortunately the techniques behind these tools do not scale up for IPv6 connected devices. By leveraging IPv6 discovery techniques from Table 2.3 such as [81] and scan data as collected by Shodan [208], researchers could further identify more services running on IoT devices.

Illicit marketplace forensics Since the early 2010s, illicit online anonymous marketplaces [50, 217] have experienced rapid economic growth. Due to the very nature of the goods being sold (primarily narcotics [50], but with digital goods a sizeable component as well [245]), those markets are constantly under scrutiny and attempts at take-down by law enforcement.

The set of discovery tools described above might seem to be of limited use to this application case, since the markets are hosted on Tor hidden services [68], which conceal IP addresses and related metrics (e.g., BGP AS numbers); by the same token, no DNS records are available, if these servers are properly configured. However, the hidden service protocol involves the registration of a `.onion` pseudo-DNS name with Tor's directory servers. Similar to what happens with DNS, some operators elect to have vanity addresses—the now defunct Silk Road market was notoriously hosted at `silkroad6ownowfk.onion`. As such, the `.onion` address, like a DNS name, becomes a network identifier that might be enumerated. Some online aggregators such as `dark.fail` provide a list of verified links (primarily to prevent phishing scams) as a form of directory for some `.onion` services (Edge (0)). Along the same lines, even Wikipedia sometimes contains links to these services (*e.g.*, the former Silk Road address is mentioned in the relevant article). In addition, researchers have shown that it was possible to exhaustively list all onion services by passively listening to announcements [28] (Edge (1)). While this specific problem has long been fixed, it is worth noting that `.onion` names, much like domain names are discoverable network identifiers.

In addition, misconfiguring a hidden service frequently leaks information that can be used to identify traditional network identifiers and services (Edges (3) and (4)). For instance, the notorious AlphaBay marketplace briefly leaked an email address belonging to its operator—ultimately facilitating the marketplace take-down and the operator's arrest [165]. While AlphaBay is an extreme example of a data leak, it is not particularly unique—Silk Road [50] also fell victim to a similar mishap. Setting aside such egregious mistakes, tools like CARONTE [143] automate the exploitation of information leaks to attempt to deanonymize hidden services by revealing their IP addresses (Edge (1)).

More generally, much like their legitimate counterparts, modern online anonymous marketplaces rely on several servers, for load balancing traffic, denial-of-service resilience, and decoupling of basic functions (e.g., backend database vs. "hot wallet" server hosting cryptocurrency holdings). Using the type of asset discovery techniques discussed earlier, one of these servers leaking information (*e.g.*, an IP address) could point investigators to other servers of interest. This is what seems to have happened with Silk Road, though details in the criminal complaint [1] are scarce. For instance, even though no public details about the investigative techniques used in Operation Onymous [233] were shared, the mere fact that a police operation succeeded in taking down multiple marketplaces at the same time suggests that successful asset discovery took place. Because the take-down was incomplete (leading markets such as Evolution survived this police operation), we can only speculate that the markets that were identified might have shared part of their infrastructure, which turned out to be vulnerable. For instance, they might have had some portions of their services hosted on the same vulnerable platforms.

More generally, a hidden service could leak information by its mere existence. Indeed, assume that you run a virtual private server, and notice that one of your hosted machines only connects to a single IP address, which happens to be a Tor node, and that all traffic patterns resemble typical Tor patterns: this is pretty clear evidence that a Tor hidden service is running on the hosted machine. Subsequently, an adversary might use this information to discover *which* hidden service is running on the machine. Historically, this could be done by injecting a large number of nodes in the Tor network and hope the hidden service eventually picks one of the adversarial nodes as a "guard" (the first connection in the Tor circuit used by the hidden service to talk to the rest of the world), thereby revealing its IP address to the adversary [172] (Edge (4)). While this issue has mostly been resolved by picking long-term guards, more recently, novel passive attacks against hidden services [126] have been proposed. The short story is that maintaining anonymity is extremely difficult, and subject to similar techniques.

Take-away: Even in cases where assets purposely attempt to avoid detection, the techniques described above make it possible for an investigator to obtain considerable additional information from limited information leaks.

2.5. Discussion and Concluding Remarks

Asset discovery is typically not the main focus of network measurement research. As a result, it often does not get the attention it deserves, since enumerating a population of assets involves many trade-offs. It would be far better to explicitly consider the choices and leverage novel techniques proposed by others. However, in practice, an understandable focus on primary measurement tasks, and the lack of a consistent framework to "plug together" asset discovery techniques, often result in an incomplete or ad hoc asset discovery process.

To address these issues, we present a framework for asset discovery. Our framework proposes a syntax to make explicit the steps in the asset-discovery process. This in turn provides a natural way to identify gaps in study design, thereby creating opportunities to build on earlier efforts by broadening the set of assets discovered. Furthermore, our systematization of recent advances in active asset discovery can help researchers select relevant techniques. Finally, we apply our framework to various use cases, which illustrates how techniques can be combined and how to identify where gaps remain.

Looking at the past five years of newly developed asset discovery techniques, the introduction of ZMap in 2013 had a long-lasting impact on the asset discovery process by enabling exhaustive scans of the IPv4 address space. However, the continued migration toward IPv6 has prompted research on IPv6 space scanning, which cannot be done exhaustively. Despite preliminary progress, this remains an area of future research.

DNS emerges as a central tool for discovering assets, especially network services. Interestingly, this appears to be a fairly recent trend, as the academic research community has seemingly overlooked DNS as an asset discovery tool for many years. Similarly, we find DNS helpful in locating IPv6 network resources. We therefore conclude that DNS-related asset discovery techniques could become more prevalent, as more corner cases and niche features of DNS are explored.

For topology-related asset discovery, the CAIDA prefix-to-AS and the RouteViews data have become the *de facto* standard across several recent publications. Nonetheless, these datasets also highlight that asset discovery is usually not the main goal. Instead, we find that researchers regularly create datasets and techniques to model how the Internet works, and not to discover assets in the cases we outline, such as red team use. Similarly, we find a large body of work focusing on different forms of misconfigurations [67, 82] and how to discover them on the Internet, without having the operational aspect of computer security in mind. This, again, highlights the value of our classification and formalization we provide in this work, as it enables researchers to assess those techniques' value for being used in asset discovery.

Our framework can also foster new approaches to asset discovery, by re-assessing and re-positioning established techniques in a new context. For instance, the discussion of illicit marketplace exploration in Section 2.4 shows how our framework can help identify functional analogies and similarities between *a priori* different protocols. Service discovery techniques for a given protocol may then be successfully repurposed for a different protocol. In turn, our framework can help implement asset discovery for concealed network services, *e.g.*, via CBSP or even Tor.

Additionally, this paper sheds light on some causes behind the replication challenges present in cybersecurity and network measurement research. A natural explanation for why it is hard to replicate network measurement results is that we are studying a dynamic network whose configurations regularly change (*e.g.*, IP address churn through DHCP, changing BGP routes). While these are real impediments, inconsistent use of different combinations of techniques can yield different asset compositions for investigation, which hampers replication. While not eliminating these problems, our framework does make such differences explicit, which may point to a solution.

Besides its immediate significance, we believe our work will remain relevant and valuable for researchers in the future. While protocols evolve, *e.g.*, DNS over HTTPS (DoH), the underlying concepts—connections, state, and identifiers—remain the same as they were 40 years ago, when TCP/IP was first introduced. Consequently, the techniques we describe will continue to be usable in research for some time. Furthermore, many of the techniques are not explicitly advertised as asset discovery techniques. Having them described and referenced in our paper might help keep future researchers from having to reinvent the wheel. The framework itself can be continuously extended as new techniques emerge in scientific literature. When such new techniques arise, researchers can map them onto the framework using the proposed syntax and combine them into tool-chains needed for conducting their own research. For example, we have found a lack of IoT discovery techniques in the scientific literature, even though there are more IoT connections to the Internet than non-IoT connections [138]. Additionally, we noted a dearth of research focusing on discovery of network services running on non-standard ports, an important but overlooked subject.

The proposed syntax and framework is very general and abstract enough to fit changes in networking. Even networks that do not operate on TCP/IP communicate through network identifiers and network services. Nonetheless, major changes in networking and protocols are entirely possible (e.g. New IP [102], LoRaWAN [136]). If a significant change occurs in the future such that networks will behave differently than those of today, the techniques mentioned in this systematization will become less relevant. In that case, however, our framework can still incorporate the changes, as the concept of communication does not change: network services will always have to make identifiers accessible to users.

3

Evolution of NIDS rules, alerts, and incidents

Notwithstanding the predicted demise of signature-based network monitoring, it is still part of the bedrock of security operations. Rulesets are fundamental to the efficacy of Network Intrusion Detection Systems (NIDS). Yet, they have rarely been studied in production environments. We partner with a Managed Security Service Provider (MSSP) to gain more insight into the evolution of rulesets, the alerts that they trigger and the incidents that get investigated. We analyze a combined ruleset –including both commercial and proprietary rules– that consists of 130 thousand rules and was used to monitor hundreds of networks. We find that these rulesets keep growing over time but there is almost no overlap among them in terms of detection options or indicators of compromise. The combined ruleset triggered more than 62 million alerts and led to 150 thousand incident investigations by SOC analysts, though the vast majority of rules never triggered a single alert. We find that just 0.5% of all rules are responsible for more than 80% of the alerts and incidents and only 1.2% of all alerts were deemed to merit closer investigation. Of all incidents, 16% were labeled as false positives and 9% carried significant risk to the client organization. Independently of the type of rule, updating rules is a minor activity. Most rules are never modified and only a fraction is deleted, except for periodic purges in some sets. Seven in-depth interviews with rule developers corroborate the patterns found in our analysis. Finally, we identify several rule management practices that influence rule and ruleset efficacy, such as supplementing commercial rules with your own and making rules as specific as possible.

3.1. INTRODUCTION

Over the last decade, the number of network attacks has steadily increased, putting even more emphasis on the efficacy of network monitoring [83]. Recent research in network intrusion detection has focused on statistical and machine-learning methods—e.g., [148, 209, 220]. The underlying motivation is that conventional rule- and signature-based methods are deemed unable to keep up with the fast-evolving threats and therefore they will become increasingly obsolete. As early as the turn of the century, industry reports were predicting the demise of signature-based network monitoring [39, 240]. Yet, two decades later, signature-based monitoring is still part of the bedrock of organizational security. Walk into any security operations center (SOC) and what you will see is analysts triaging alerts generated by network intrusion detection systems (NIDS) that still rely heavily on rulesets. Most, if not all, Managed Security Service Providers (MSSP) purchase and develop rulesets in order to detect relevant events on their client networks. Despite important advances in host-based detection and anomaly detection, there are few signs that this is changing any time soon.

To put it a bit facetiously, rules work in practice, but not in theory. Rulesets are incredibly important for protecting organizations everywhere and yet they are barely researched in real-world production settings. Prior work has focused mostly on analyzing the impact of rulesets on the system performance of an NIDS in a test setting, measuring sensor-based metrics such as CPU load, memory usage and packet loss [10, 63, 207, 246]. These studies looked at the sensors, not at the rulesets themselves, nor how they are managed to optimize detection capabilities. Other work has focused on improving the signatures used in NIDS [124, 213, 241, 243, 252]. The closest work we could find was a recent study by Gashi and Asad [16], who compared four different open and commercial rulesets, their changes over a period of five months and the alerts they generated in a test on a part of a university network. They concluded that there was barely any overlap in alerts, suggesting that organizations should combine rulesets to optimize detection.

To the best of our knowledge, we present the first study that opens up the black box of NIDS rulesets and their management in a real-world production setting. We have partnered with an MSSP that monitors hundreds of client networks using various commercially-acquired rulesets as well as a ruleset it develops in-house. We present an analysis of four rulesets that collectively consist of 130 thousand rules, span 13 years (2008–2021), and functioned in a production environment to monitor hundreds of networks. The rulesets received more than one million modifications, triggered more than 62 million alerts and led to 150 thousand incident investigations by SOC analysts.

How do rulesets evolve over time? How fast do they get revised and updated? How many rules actually trigger an alert in practice? How many of these alerts represent real threats?

We are especially interested a critical underlying tradeoff: How are rulesets managed in order to maximize detection of intrusions (true positives) and minimize the number of alerts that need to be investigated (false positives)? You can only detect what you create a rule for and yet the larger the (combined) ruleset, the harder it will be to keep it accurate and up to date and the higher the likelihood of overwhelming the SOC analysts with irrelevant alerts.

We analyze the changes in the rulesets, the scale and type of modifications and the relation between the rules and the alerts and incidents that are triggered by them. We complement this analysis by a small, but in-depth, user study of seven rule developers within the MSSP. We conducted semi-structured interviews to understand the process of rule writing and management.

In summary:

- We present the first longitudinal analysis of NIDS rulesets spanning up to 13 years (2008–2021), the alerts they generated and incidents that they helped detect.

- We created a tool to quantitatively analyze NIDS rule changes. This tool allows tracking the evolution of a rule over its lifespan, extract metadata from the rule's syntax and classify the different types of modifications. We open source this tool to the community.

- While in use, around 23% of all rules are updated in terms of detection capability. We quantify the factors that trigger changes within rulesets and find two major causes: (i) changes in the threat landscape; and (ii) reducing the number of alerts by making rules more specific. These factors are then cross-validated through semi-structured interviews with seven experts in the area of rule developing and security incident response.

- We find that 80% of all alerts were triggered by a mere 0.5% of all rules. The overwhelming majority of rules never trigger a single alert. This helps explain why rulesets can grow into the tens of thousands over time, without overwhelming SOC analysts. Still, only 1.2% of all alerts were deemed to require closer investigation, and only 0.3% of all alerts carried significant risk to the organization.

- We identify a number of rule management practices employed by rule developers and their influence on the efficacy of the rules. Specifically: (i) multiple rulesets are used simultaneously due to the lack of exhaustive coverage of the threat landscape by any single ruleset; (ii) proprietary rules are updated more often than commercial rules leading to a higher incident detection rate; and (iii) false positive incidents have a noticeable impact on rule updates.

3.2. RELATED WORK

Research on ‘traditional’ signature-based NIDSs and NIDS rulesets is relatively rare. Modern research focuses primarily on creating statistical or machine learning-based NIDSs in lieu of studying or improving upon signature-based approaches, which remain an industry standard to this day. Examples include Srivastav and Challa[220], Shone et al. [209], and Mirsky et al. [148].

In the rare instances where research has indeed looked at signature-based NIDSs, the effort has gone into studying the performance of the system itself and what the effect is of ruleset volume changes, instead of the processes driving rule changes. Soumya Sen [205] examines the Snort IDS and its performance when varying bandwidth and, perhaps unsurprisingly, that as bandwidth increases, the size of the ruleset must decrease in order to keep Snort’s error rate below a certain threshold. This effect is magnified for a given bandwidth when the size of IP payloads decreases, as Snort will have to process more packets during the same amount of time, resulting in more packet loss.

Thongkanchorn et al. [232] perform an analysis similar to Sen [205]. The authors take the Snort, Suricata, and Bro (now Zeek) NIDSs and analyze their performance when varying attack type, ruleset sizes, and network traffic rates. For the experiment, they use the Emerging Threats (ET) Open ruleset for all NIDSs. They select a set of different types of attacks and generated the corresponding packets to test the different NIDSs. Results from the analysis include low packet loss and low CPU usage for TCP traffic, and that a higher traffic rate leads to higher CPU usage, higher packet loss, and a higher number of generated alerts. Finally, the authors also find that using more rulesets also causes a higher number of generated alerts. However, the study does not examine the underlying causes for the performance detriments, such as potential inefficiencies in the rules.

Gashi and Asad [16] perform a study more closely related to our work, as they analyze the evolution of and similarities between four different rulesets: Snort’s official Community, Registered, and Subscribed rulesets, as well as the ET Open Suricata ruleset [187]. The analysis was performed on five months of rule updates. They examine the diversity and overlap between both blacklisted IPs and rule content options for the four different rulesets. The authors find that there is minimal overlap between ET’s and Snort’s IP blacklists and rule content options, with only 1% of rules containing matching content options. As a result, the alert triggering behavior of the rulesets from both sources also has minimal overlap, potentially indicating that both may be useful to provide more defense to a network.

While the work from Gashi and Asad [16] is somewhat related to the analysis we carry out in this work, there is only marginal overlap. To the best of our knowledge, there has been no research into the nature of rule changes and the rule development process as a whole. Neither has this rule data been combined with alert and incident data from production settings – in our case: an MSSP – to investigate the evolution of the rulesets in a changing threat landscape.

3.3. BACKGROUND

At the core of this work are two closely related topics, namely intrusion detection systems and the rulesets they employ to detect potentially malicious traffic. In this section we explain these concepts and place them within the context of an MSSP.

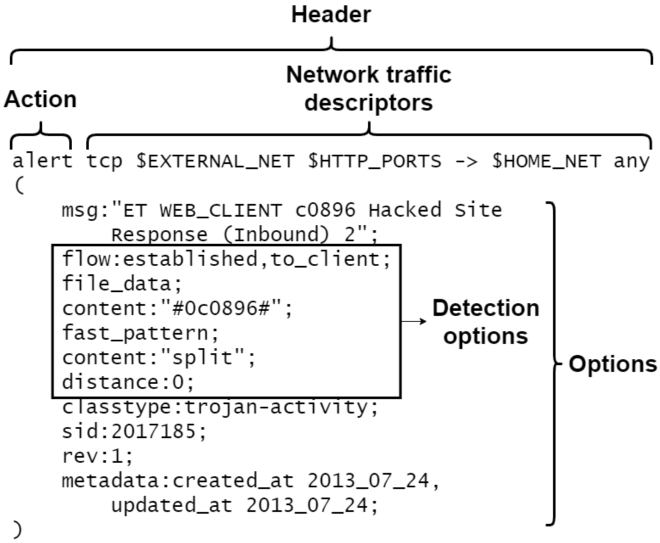
3.3.1. INTRUSION DETECTION SYSTEMS

IDSs are classified by their placement and by the techniques that are used for detection itself. As for the placement of an IDS, there are network-based, host-based, and application-based intrusion detection systems. Furthermore, depending on its method of detection, an IDS can be signature-based or anomaly-based [146]. The type of IDS that is the focus of this work is the signature-based NIDS.

A network-based IDS is placed at a strategic point within a network and analyzes all network packets it receives to detect attacks. Signature-based IDSs find malicious activity by matching it with a predefined set of patterns or events that are characteristic of known attacks. Examples of such IDSs include Snort [53], Suricata [223], and Zeek (formerly Bro) [182].

3.3.2. NIDS RULES AND RULESETS

Signature-based IDSs detect threats in a network with rules that inform the system what to look for in network traffic. Figure 3.1 illustrates the syntax of such a rule. An NIDS rule consists of two main parts: (1) the header, and (2) the options. The header can also be split up into two different parts: (1a) the action, and (1b) the network traffic descriptors. The action is the portion of the header that tells the NIDS what to do in case the rule is triggered (e.g., raise an alert, drop the packet, log the packet, among others). The second portion of the header that we identify are the network traffic descriptors, which specify the origin and destination IP addresses and ports, as well as the protocol of the packet that the rule looks for. The secondary section of the rule is the options, which can be split up into two parts: (2a) the detection options, and (2b) the non-detection options. They determine how a packet is analyzed, and contain fields for matching packet headers, or sections of the payload content, among others. Non-detection options provide additional information regarding the type of traffic the rule wants to detect, such as the category of threat, references to documentation about the threats, as well as rule ID and version number.



```
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any
(
    msg:"ET WEB_CLIENT c0896 Hacked Site
        Response (Inbound) 2";
    flow:established,to_client;
    file_data;
    content:"#0c0896#";
    fast_pattern;
    content:"split";
    distance:0;
    classtype:trojan-activity;
    sid:2017185;
    rev:1;
    metadata:created_at 2013_07_24,
        updated_at 2013_07_24;
)
```

Figure 3.1: Rule syntax used by Snort and Suricata NIDSs.

Rule developers create rules that are as specific to the threat as possible, but also generalizable to different versions of the same threat present on the Internet. Regardless of how this trade-off is made, it will inevitably imply some fraction of alerts being false positives, due to the sheer amount of network traffic flowing through the Internet. A false positive is defined as normal or legitimate traffic that is mistaken as malicious. Conversely, malicious activity can be missed by an IDS if there is no rule present that explicitly detects it. In order to keep this risk at a minimum, organizations need to continuously keep up with the latest threat intelligence and ensure their NIDS is up to date. Organizations can do this by either purchasing a subscription to a commercial ruleset created by a third party, or develop the rules in-house.

Figure 3.2 illustrates the typical life cycle of an NIDS rule. Rules developed by the MSSP are either directly added to the production environment, or are first added to a test environment to evaluate accuracy and false positive rates. Commercial rulesets are added to production directly, as it is assumed that these rules have already undergone an acceptable level of quality control. Once in production, the MSSP only updates its own proprietary rules; the commercial rules are left alone and are only filtered out through other methods. Rules in the production environment trigger alerts if the logic within the rule matches with an incoming packet. Finally, only if an alert is deemed severe enough by the analysts in the SOC, the alert is grouped together with a number of related alerts into an *incident*. This incident is then thoroughly investigated by the security analysts.

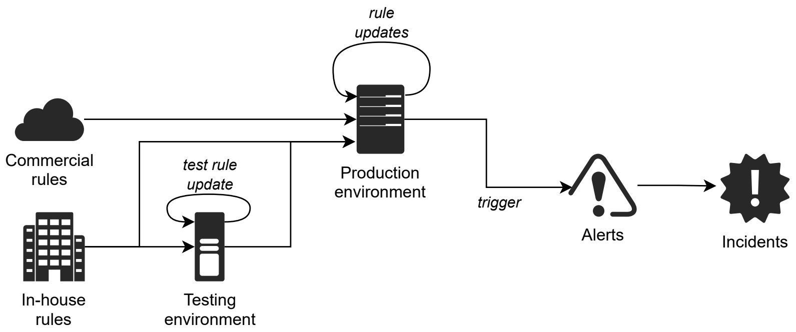


Figure 3.2: Life cycle of an NIDS rule. Commercial rules are added to production immediately, while many rules developed in-house are tested within a testing environment before they are added to production. Once in production, rules are updated or removed from the ruleset. Rules in the production environment produce alerts if they detect packets that match its detection logic. If deemed severe enough, related alerts are grouped together into an incident and thoroughly investigated by analysts.

3.4. DATA AND METHODOLOGY

Our collaboration with an MSSP has enabled us access to a number of datasets surrounding their NIDS and SOC operations. In this section we discuss these datasets and how we use them to perform this work.

3.4.1. DATA COLLECTION

NIDS rulesets The MSSP uses rulesets from different origins on the network sensors that they install on their customers' networks. In addition to commercial rulesets purchased from Emerging Threats [186] and VRT [54], they also employ a proprietary ruleset that is created and maintained by an in-house team of developers. All rulesets are hosted on internal Git repositories. The MSSP has been using its own ruleset since 2008, and the repository has been tracking the changes to that ruleset ever since. Commercial rulesets have also been used since that time, but have only been mirrored on a local repository since 2018 or 2019, depending on the ruleset, meaning that the commercial ruleset data are limited to those years. Both VRT (now Talos) and Emerging Threats are market leaders in NIDS rulesets, with the former being the maker of the official Snort ruleset, and the latter being one of the three paid services specifically mentioned by OISF, Suricata's maintaining organization, in their 2018 SuriCon conference [168]. We feel this is a representative sample of the NIDS market, and we therefore limit our study to the two commercial rulesets.

Alert data The rules installed on the network sensors produce alerts when they detect threats, and all the alerts are logged when they arrive at the SOC. We have access to alert logs that date from mid-2009 to mid-2018. The logs are in CSV format and contain the following information: (i) alert timestamp; (ii) rule ID, revision, and priority; (iii) rule category; (iv) protocol of packet (TCP, UDP, ICMP, IP, numerous application layer protocols); and (v) potential corresponding incident. Between 2017 and 2018, the MSSP was gradually migrating to a different logging platform, causing this logging data to be incomplete throughout that last year.

Incident data One or more alerts deemed critical enough by SOC analysts to merit closer investigation are grouped together into an *incident*. The analysts then investigate all related alerts to determine the cause and severity of this incident and ultimately label it accordingly. These labels are *Undetermined*, *False positive*, *Not interesting*, *Interesting*, *Low risk*, *High risk*, and *Successful hack attacks*. We consider all but Undetermined and False positive incidents as true positives. The difference between a *High risk* incident and a *Successful hack attack* is that the former involves activities that will directly lead to network compromise, while the latter is an incident where such network compromise has already occurred without the SOC being able to halt the attempt. *Undetermined incidents* (0.9%) are excluded, as we cannot establish a reliable label for them.

The incident logs contain the following: (i) incident open, response, and close timestamps; (ii) corresponding customer; (iii) category label; (iv) whether the incident was escalated to the customer. This data also dates from mid-2009 to mid-2018 and is also missing entries from 2018 due to the platform migration process.

Interviews We carried out semi-structured interviews with seven analysts and rule developers employed by the MSSP to better understand their heuristics and work processes. We recorded, transcribed, and coded the interviews to extract the relevant information.

3.4.2. QUANTIFYING RULESETS' EVOLUTION

Although changes to the rules are tracked by the Git repository, the evolution of the ruleset as a whole is not. For the MSSP's proprietary ruleset, there are no set guidelines when it comes to managing rules and rulesets. New rules are created on the basis of new threat intelligence, and from analysis of potentially malicious software. Furthermore, we learn from the interviews that there are, in general, two instances where ruleset updating occurs: (i) when analysts or rule writers have no other work activities planned and go through the ruleset out of their own initiative; (ii) when a rule triggers alerts more often than is deemed acceptable by the analysts. While the creators of the commercial rulesets might also have such guidelines, the reasons behind rule updates are not logged (publicly) either.

We make a distinction in the methodology that we apply to the two different datasets that we have access to. The first is the repositories of both the commercial and proprietary

NIDS rules. The second is the datasets containing the triggered alerts and incidents, which will be discussed in the next subsection. We analyze the repositories of the commercial and proprietary rulesets separately, since the two types of rulesets differ significantly not only in size, but also in purpose.

First, we implement a tool that is able to track the evolution of the ruleset over time. This tool checks out every commit made to the repository and keeps track of the additions and deletions made to the ruleset. It does this by traversing the repository in a reverse chronological order and parsing the complete ruleset for every commit in pairs of two. A *new rule* is a rule that is present in the latter ruleset and not present in the former. A *deleted rule* is a rule that is present in the former ruleset and not present in the latter; or a rule that is not commented out in the former ruleset and is so in the latter ruleset; or a rule that has been moved to a specific file designated for deleted rules. An *updated rule* is a rule that is present in both former and latter rulesets and differ in text. However, not all updates are made equal. Only updates to the header and detection options will influence the performance of a rule. Therefore, the tool also differentiates between the types of updates made to a rule. The result of this analysis is a list of all the rules that were ever added to the repository, paired with every modification performed on each rule over the lifetime of the repository.

To the best of our knowledge, a free and open-source NIDS ruleset analysis tool has not been developed and released before. We realize that such analyses might be valuable for other researchers and professionals. As a contribution to not only the scientific community, but the numerous users of this type of NIDSs in industry, we have published the source code on Github [1].

The tool's output allows us to calculate a number of statistics, both in total and longitudinally, that we used for further analysis of the rulesets. Such statistics include rule changes occurred in total and their type, the lifespan of individual rules, and size of the ruleset over time. All statistics are discussed in Section 3.5.1.

Manual inspection of rule changes was also performed. Specifically, we randomly sampled 50 rule updates and determined the reasons behind them (i.e., was the rule made more specific/efficient/etc.).

All rules, commercial and proprietary, have their own unique ID, and the MSSP ensures that there is no overlap between the IDs of proprietary rules and commercial rules. Every alert record includes a reference to the unique ID of the rule that triggered it. This enables us to link an alert to a specific version of its corresponding rule. By examining the changes to the Git repository, and combining this information with alert data and statistics, we can learn not only how rules change, but also ascertain the reasons behind those changes and identify

[1] https://github.com/mathewvermeer/ruling-the-rules

trends in the ruleset's evolution (e.g., moving away from very specific to more generic rules, more rule updates than rule deletions, etc.). This data also allows us to examine the relationships between rules and alerts.

The proprietary ruleset itself consists of two parts: (1) the rules in production, and (2) the rules still in testing. In this study, we focus primarily on alerts that influence the workflow of the SOC and its analysts. For that reason, we focus only on the ruleset in production and discuss the evolution of rules in the testing environment separately.

3.4.3. INTERVIEWS

To complement our data on the rulesets, alerts and incidents, we also conducted seven interviews with security professionals within the MSSP who write or manage NIDS rules, or have done so previously in their career. The participants were recruited though snowballing—i.e., asking each participant for a list of names of other people who would be relevant to interview on the company's NIDS rule development and management. This process was halted when the only names we received were professionals who we had already interviewed before and two participants who did not respond to our invitations. The semi-structured interview protocol consists of 37 questions (Appendix A.2). Each interview was conducted by two interviewers, recorded and transcribed.

These interviews are part of a larger, separate study with similar interviews in other MSSPs (Chapter 4). Here, we only include the data from interviews inside the partner MSSP for two narrow purposes: (i) to reconstruct the rule development process and understand the datasets at the MSSP (Sections 3.3.2 and 3.4.1); and (ii) to help interpret and validate our findings from the analysis of the rules, alerts and incidents. In Section 3.8 (Discussion), we compare our findings to what we heard in the interviews.

3.4.4. ETHICS

To conduct the user study, we received formal approval from the Human Research Ethics Committee at our institution. All interviewees explicitly consented to the recording and transcription of the interview and to the usage of quotes. We minimize the risks of data leaks by anonymizing all data gathered during the interviews. The recordings were stored for the duration of this research on an encrypted hard drive. All answers given were confidential and were only available to the other researchers involved in this project.

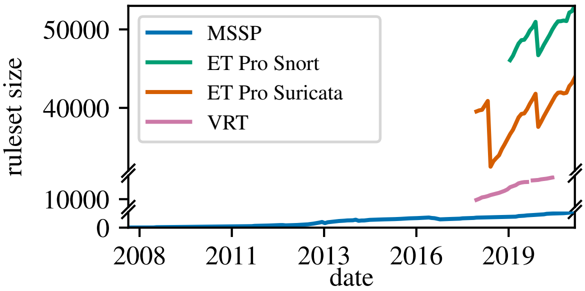


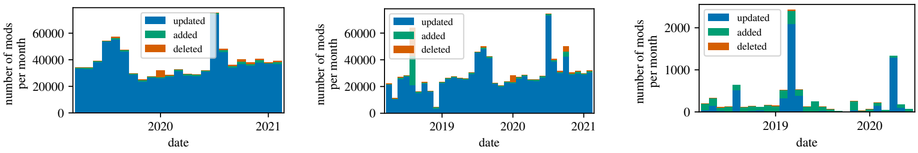
Figure 3.3: Size of the proprietary and commercial rulesets employed by the MSSP over time. The following are the ruleset sizes at the start and end of their respective curves: MSSP: 20-2,065; ET Pro Snort: 46,074-53,000; ET Pro Suricata: 39,546-43,863; VRT: 9,908-12,760.

3.5. RULE EVOLUTION IN PRACTICE

The rulesets employed by MSSPs are very large and, at the surface level, we find that changes are made to these rulesets constantly. Upon further inspection, though, we discover a number of different phenomena within the rulesets. Firstly, many of the created rules are never meaningfully updated while remaining in use over a long period of time, though this differs per ruleset: 84% and 68% for the ET Pro Snort and Suricata rulesets, 97% for the VRT and 65% for the MSSP rulesets. Then, there is the subset of rules that are deleted without ever being updated. This subset makes up 12% of the MSSP ruleset, and at most 0.6% of the commercial rulesets. Finally, there are rules that are updated throughout their lifetime. The updates themselves can be divided into four different groups: updates made to a rule's (1) action, (2) network traffic descriptors, (3) detection options, and (4) non-detection options. The second and third types of updates are what we will mainly focus on, since those are the types that affect detection. The ET Pro rulesets overwhelmingly perform network traffic descriptor updates over all the other types, while the VRT ruleset receives much more non-detection option updates, each signalling different priorities as to their ruleset management strategies. The subsections elaborate on the deeper examination we performed to shed light on the unique behavior of each of the rulesets.

3.5.1. RULESET EVOLUTION

Ruleset size. Figure 3.3 illustrates the sizes of all the different rulesets employed by the MSSP. It seems to show a general increasing trend for all rulesets. However, the data we have of the commercial rulesets are limited to the last two to three years, trends across longer time spans are not visible. The evolution of the ET Pro rulesets are punctuated by significant purges of rules, with a linear increase between the purges. The third commercial ruleset used

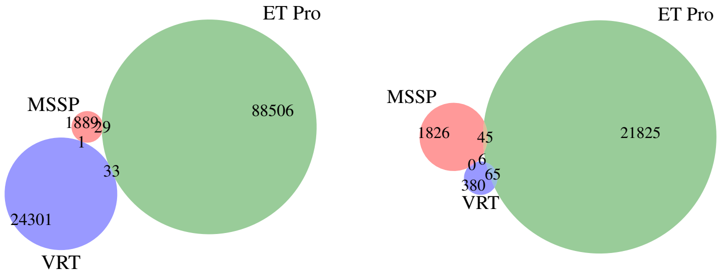


(a) ET Pro Snort ruleset

(b) ET Pro Suricata ruleset

(c) VRT ruleset

Figure 3.4: Number of additions, modifications, and deletions performed on the different commercial rulesets.



(a) Detection options overlap

(b) Unique IP addresses and domain names overlap

Figure 3.5: Detection options, and unique IP and domain overlap, respectively between the different rulesets.

is the one created by VRT, now Talos, which is the organization that maintains the official Snort rulesets [54]. Though the size of this ruleset is significantly smaller than the ET Pro rulesets, it exhibits similar behavior to the ET Pro rulesets in the apparent steady addition of new rules without concurrent deletion of old and potentially outdated ones. Figure 3.4 illustrates this. This could mean that they either prefer to have the most number of threats covered by their rules regardless of quality, or deem most newly created rules of a high enough quality to remain (unchanged) in the ruleset indefinitely. The MSSP's ruleset is significantly smaller than the commercial rulesets, which can be explained by the fact that ruleset creation is not the core of the business. Similar to the ET Pro rulesets, the MSSP also has occasional purges of rules. As evidenced by the rising curve in Figure 3.3, though, this deletion of rules does not happen regularly, nor does it happen enough to maintain the ruleset at a (near) constant size, similar to the VRT ruleset. This indicates that efficiency through ruleset size limits is not a priority for rule management, though it certainly was in the past [205].

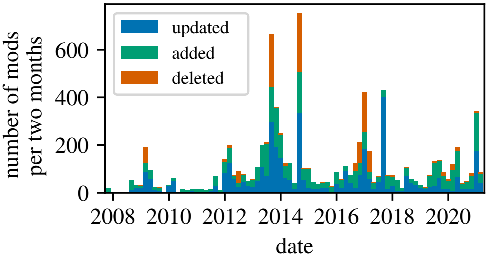
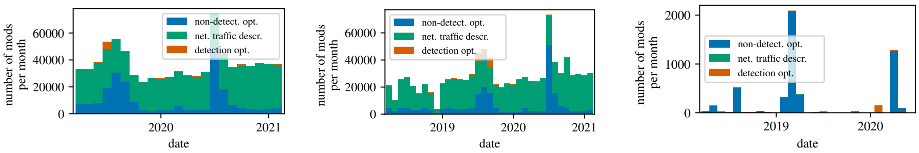


Figure 3.6: Number of updates in the proprietary ruleset.



(a) ET Pro Snort ruleset **(b)** ET Pro Suricata ruleset **(c)** VRT ruleset

Figure 3.7: Number of updates to existing rules per month and the type of modification for the different commercial rulesets. Updates to non-detection options are shown in blue, network traffic descriptors in teal, and detection options in red.

Next, we examine the overlap between the rulesets using two different measures, illustrated in Figures 3.5b and 3.5a, respectively. The first method used here is comparing the detection options extracted from all rules in a manner similar to that performed by Gashi and Asad [16], which aims to measure the functionality of the rules themselves. We supplement this overlap measure with the overlap in unique IP addresses and domains present in a rule. This measure is an indicator for a subset of threat intelligence-based rules that act as a blacklist (i.e., raise an alert solely due to the presence of a specific IP or domain). While the former measure has limitations, since different implementation of the same detection logic would result in no match, taking both measures together yields a general overview of the threat coverage of the different rulesets. Irrespective of these limitations, both figures show that despite the large sizes of the commercial datasets and the large coverage they provide, the overlap is nearly negligible.

Rule updates. Of all rules currently active in the different rulesets, many have never been updated after their creation. This percentage stands on roughly 14% for both ET Pro rulesets, 42% for the MSSP ruleset, and 69% for VRT. There is also a set of rules that are never touched until their deletion. This phenomenon occurs more often in the MSSP ruleset

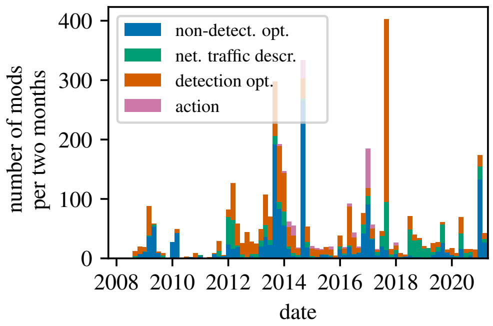


Figure 3.8: Number of modifications to existing rules in the proprietary ruleset per two months and the type of modification made. Updates to non-detection options are shown in blue, network traffic descriptors in teal, detection options in red, and to the action in pink.

than the commercial rulesets: of all rules ever added to this ruleset, this number stands at 12%. For all commercial rulesets, this percentage is at most 0.6%.

Then, there is the set of rules that are updated throughout their lifetime. We disregard trivial updates to the non-detection options, as these have no effect on the working of a rule. To investigate the nature of these rule updates, we sampled 50 rule updates that alter either the rule header or the rule's detection options from the different rulesets and grouped them into a number of broad categories:

- *Efficiency improvements*: Changes to a rule that make the rule faster or less resource intensive, for instance using the `fast_pattern` keyword or optimizing byte comparisons or regular expressions, while keeping the rest of the rule unchanged.
- *More specific*: Narrowing the rule's search space by, for instance, increasing the length of the content string to match or explicitly specify the subset of ports on which to look for the threat.

Table 3.1: Rule update categories for the 50 randomly sampled updates from every ruleset.

Type of update	MSSP	ET Pro Snort	ET Pro Suricata	VRT
Efficiency improvement	10%	-	-	6%
More specific	36%	-	-	30%
More general	10%	-	-	24%
Bug fix	4%	-	-	30%
Threat intel update	22%	100%	100%	-
Other	18%	-	-	10%

- *More general*: Widening the search space in a manner contrary to the previous.

- *Bug fixes*: Updates made to a rule due to the rule not working as intended or improper usage of certain rule options. Examples include detecting buffer overflows using `isdataat` instead of the `dsize` keyword, as usage of the latter can lead to false negatives.

- *Threat intel updates*: Implement into a rule discoveries made through threat intel such as changes to malicious domains or IPs or changes in how certain malware operates.

- *Other*: Any other type of modification such as change in rule action, editing of flowbits, or fixing typos.

This categorization is illustrated in Table 3.1. Interestingly, for the ET Pro rulesets, all of the sampled rule updates fall under the "threat intel update" category. The sampled updates from both the proprietary and VRT rulesets are distributed in a more balanced manner across the different categories. It seems that there is a general trend towards making rules more specific, perhaps to reduce the chances of false positive alerts.

We can also split the non-trivial updates into two broad categories, the first of which is network traffic descriptor updates. Emerging Threats is, essentially, the only organization that focuses largely on network traffic descriptor updates (see Figures 3.7 and 3.8). The MSSP also performs such updates, although hardly to the extent as Emerging Threats. VRT, on the other hand, performs a negligible number of such updates. Both this, and the findings in Table 3.1, are consistent with Figure 3.5b.

The second category of non-trivial updates are those made to a rule's detection options. We first examine these changes made by the updates by means of the Levenshtein ratio between two consecutive versions of a rule's detection options, shown in Figure 3.9, which is a measure of their similarity. The ratio is computed with the following formula: $Lev_{ratio} = \frac{\text{sum of string lengths}-2\times Lev_{distance}}{\text{sum of string lengths}}$. Clearly, this metric is not suitable for the analysis of rule update semantics, since single-character changes could be the difference between detecting a threat or not, and large changes could be an attempt at efficiency improvement without affecting detection at all. Therefore, we simply use it to identify update trends and mass update events. We see that the ET Pro rulesets exhibit notably different behavior: the Suricata ruleset has a spike at around the 0.5-mark, and both spike at around the 0.75-mark. The latter spike corresponds to nearly $4,000$ nearly identical rule updates in each ruleset, each with a Levenshstein distance of 46. Most of the updated rules detect DNS lookups for potential trojans, ransomware, malicious URLs, etc., and the updates themselves change the manner in which the first 12 bytes of the UDP packet are matched. Interestingly, the spike at the 0.5-mark in the ET Pro Suricata ruleset corresponds to updates made to the same rules as the latter spike, and again altering the way these DNS lookups are detected. In this case, the detection options are updated with the Suricata-specific `dns_query` keyword, which explains why this spike is absent in the ET Pro Snort curve. Other minor spikes in both the ET Pro Snort and Suricata rulesets are similar in that they involve the introduction of case-specific keywords due to efficiency improvements [142]. Interestingly, though, these updates were not the direct effect of keyword release or deprecation, since these updates were made many years after the fact [167], and many rules using the deprecated keywords still remain in the ruleset. In fact, Emerging Threats announced support for these new features back in late 2019 [227], but noted that the implementation of the features were still a "work in progress and under active development," explaining why such updates are still made periodically.

The fact that these spikes are so evident when looking at the ratio indicates that many of the updates in the ET Pro rulesets are made in a single wave to all of the rules of that type. And although the curve of the VRT ruleset appears much smoother than the ET Pro curves, only 355 updates are made to the detection options during the two years we have records of for that ruleset — a minuscule amount compared to the nearly 13,000 rules in this set. Finally, we see that the MSSP ruleset also produces a very smooth graph, indicating that there are no mass update events, and that the updates that do occur are done on a more case-by-case basis. Thus, most detection option changes are made to the syntax of a rule without changing detection itself in a meaningful way, such as the aforementioned DNS lookup rules.

Altogether, it seems a minority of updates are actually made due to rule performance issues. Instead, most updates involve either changes in metadata, syntax, or swapping out threat indicators. Most rules are, evidently, of high enough quality once they enter production. Furthermore, rule development is primarily input driven, with the focus lying not so much on the eventual efficacy or precision of rules as observed by an organization or SOC, but on creating these rules for new threats in the first place.

Rule deletions. Aside from ruleset management through the updating of existing rules, sometimes rules are also deleted from the ruleset. Looking at the deletion behavior in all of the analyzed rulesets, there seems to be little reason to throw out rules. As is the case with update behavior in the ET Pro rulesets, deletions also revolve around changes in threat intelligence. For instance, a large portion of the deleted rules in the major ET Pro rule purges (see Figure 3.3) involve outdated C&C servers and SSL certificates [183–185]. Thus, the absence of threat intel-based rules from the VRT ruleset could explain the lack of deletions for this particular ruleset. The MSSP ruleset somewhat shares the deletion behavior of the ET Pro ruleset, since it contains a number of threat intel-based rules, although not to the

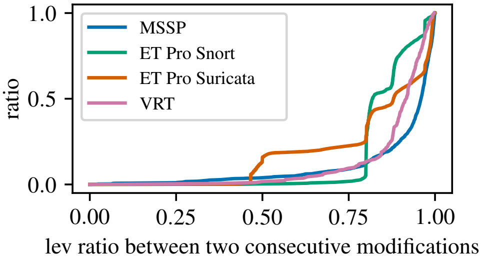


Figure 3.9: CDF of the Levenshtein ratio between the detection options of a rule and its previous version. A higher ratio indicates a high similarity between both versions.

degree of ET Pro. Many of the deleted MSSP rules, however, fall outside of this category. This unique behavior can be explained by the direct feedback loop that the rule developers have from the MSSP's SOC.

3.5.2. RULE TESTING

We must not overlook the fact that rules are tested before being enabled in a production environment. Interestingly, we see that the overwhelming majority of rules are added fairly quickly. It seems that the MSSP's priority is to have a rule pushed to production as quickly as possible, and only optimize the rule after the fact if necessary. That said, 26% of rules from the testing environment never make the cut and are deleted from testing before they make it to the production environment. While many of these rules are deleted relatively early on (roughly a third within the first week), the remaining rules are deleted in a random fashion.

3.6. ALERTS TRIGGERED OVER TIME

The MSSP with which we collaborate has a particular manner in which alerts arriving at the SOC are processed. Under normal circumstances, all alerts that arrive are processed by the analysts working in the SOC. Deviation from these normal circumstances occurs in cases of false positive floods, for instance. In such cases, the alerts from the responsible rules may be manually suppressed to prevent overburdening the analysts and taking out the SOC back-end systems. In case that one or multiple alerts are suspicious and merit further investigation, these suspicious alerts are grouped together into an *incident*. Incidents are composed of one or multiple alerts that are potentially part of the same threat. Every incident is individually investigated. From this investigation, the analysts determine whether the incident is a false

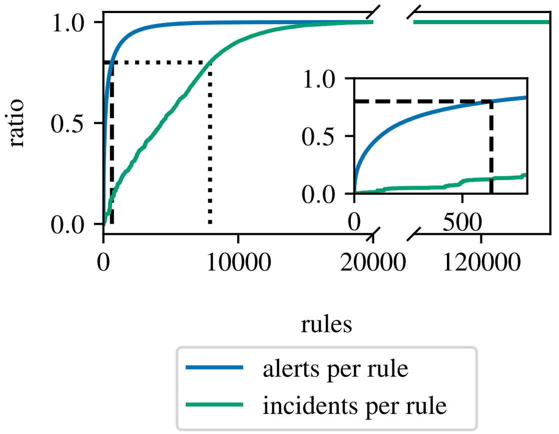


Figure 3.10: CDF of the number of alerts and incidents triggered over the rules employed by the MSSP. Includes all distinct rules that have ever been added to both the proprietary and commercial rulesets. The dashed and dotted lines indicate the 80%-mark for alerts and incidents, respectively.

positive, and if so, it is labeled as such. In case of a true positive, the analysts assess the severity of the incident and label the incident. If an incident is deemed severe enough, an escalation takes place whereby the customer itself is informed of the incident in order to carry out the necessary defensive and mitigation measures.

We have obtained nine years of alert and incident data for our analysis, from mid-2009 to mid-2018. We first analyze the raw alert data. It simply contains every alert triggered by any active rule present on the probes. Naturally, not every rule will trigger as often as the rest, as some malicious activities are more common than others. We expect this distribution to follow a power-law-esque distribution. Indeed, this is what we find if we plot the data. Figure 3.10 illustrates this, and we see that 672 rules are responsible for 80% of all alerts. Additionally, out of all the proprietary and commercial rules that we have records of, over 110,000 rules—85%—did not trigger a single alert. However, we do not have access to the entire lifespan of the commercial rulesets. As a result, we have no records of commercial ruleset modifications before May 2018. Some rules may have been created and subsequently deleted before the MSSP began mirroring the ruleset on its local repository. The actual number of rules that never generated an alert could, therefore, be much larger.

To further investigate the nature of these highly productive rules, we randomly sample and manually examine 50 of the 672 rules. This examination allows us to identify characteristics of the rules that makes them trigger the number of alerts that they do, as oftentimes rules or rule descriptions are unclear, ambiguous, or mislabeled. Of the sampled rules, 52% consists of reconnaissance activity detection and detection of known vulnerability exploitation

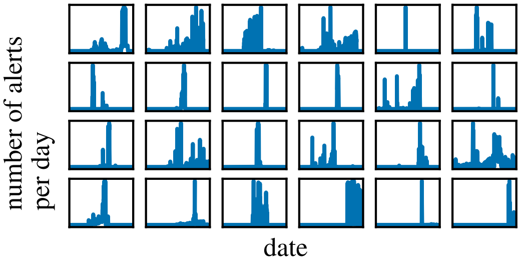


Figure 3.11: Alerts over time of 24 of the uncommonly triggered rules in our 50-rule sample. Notice the sudden spikes above the quiet baseline for many of them.

attempts. This explains the large number of alerts, since these activities are carried out by many a malicious actor on a daily basis across the entire IPv4 space. The remaining 48% not in the two aforementioned categories consists of activities that are not as common an occurrence. Examples include internal network policy violations, DNS requests for malicious domains, or usage of vulnerable software. Therefore, a high level of alerts maintained over a longer period of time is not a realistic explanation for the productivity of these types of rules. Indeed, what we find is that 28% of the total sample population are, in essence, quiet rules until a single event causes the rule to trigger up to thousands of times in a single day (see Figure 3.11).

As the size of the proprietary ruleset increased since 2008, so have the number of rules that trigger alerts: from 3,758 after the first year to 6,787 just before the MSSP started migrating to a new logging platform (mid-2017). Not only that, but the proportion of the total alerts that are triggered by proprietary rules has also increased from 1% to 6%. Nevertheless, the number of rules that compose 80% of all the alerts per year remains fairly stable every year: roughly between 200 and 250 rules. Between 18% and 50% of these rules overlap year on year.

Of the aforementioned 672 rules responsible for 80% of alerts, 164 (26%) are proprietary rules, even though proprietary rules make up just 3% of all rules employed by the MSSP that we have a record of. Additionally, of the roughly 20,000 distinct rules responsible for all of the alerts, around 1,000 rules are from the MSSP's proprietary ruleset. Taking into account the smaller size of the proprietary ruleset compared to the commercial rulesets (see Figure 3.3), it is apparent that the proprietary ruleset performs better in terms of rule utility.

Since different rules detect different threats, and, therefore, exhibit different alerting behavior, we examine if such differences in behavior are also reflected in a rule's updates. Table 3.3 shows that rules in both the commercial and proprietary rulesets that trigger alerts are over eight times as likely to receive an update than rules that do not trigger any alert, and almost thrice as likely when looking solely at the MSSP rules. We stated in Section 3.5.1 that most of the commercial ruleset updates are threat intelligence-based, where, for instance, out of date indicators of compromise are swapped out for new ones. These are updates that occur regularly, and it makes sense that they are not affected by the the number of alerts that these rules trigger. Looking solely at detection options, the number of updates drops drastically in both alert and non-alert subsets, and the update frequencies per rule drop to 0.18 and 0.16, respectively, meaning that rules that trigger alerts are 12.5% more likely to receive updates than rules that do not trigger.

3.7. SECURITY INCIDENTS

Most of the alerts produced by the rulesets are not deemed relevant or severe enough by the SOC. Of the millions of alerts that the SOC has processed over the years, only a relative handful are investigated more thoroughly: 735 thousand (or 1.2%), which are produced by 6,720 different rules. This subset of alerts add up to 150 thousand incidents. The precise numbers are specified in Table 3.2.

Of the 6,720 rules that are present in the incidents, 4,806 (71%) of them are present in 80% of the rules; a very uniform distribution that is in stark contrast to the 672 that produce 80% of all alerts. Interesting here is that of these 672 rules, only 89 are investigated by the SOC, indicating that high alert-producing rules do not necessarily provide usable security information.

The incident labels described in Section 3.4.1 allow us to group the incidents into two larger groups: risky and non-risky incidents. The risky group contains incidents labeled *Low risk*, *High risk*, and *Successful hack attack*, while the non-risky group is made up of *False positive*, *Not interesting*, and *Interesting* incidents. Due to the granularity of the labeling, this alternate categorization perhaps better represents the relative severity of the different incidents.

Figure 3.12 illustrates that the number of alerts and (risky) incidents have not only increased over time, but are also highly correlated. All three curves in the figure are also very much correlated with the increase in the MSSP's customer base. However, we must exclude the exact customer data from the paper due to reasons of confidentiality. The correlations suggests that organizations remain exposed to external threats to a constant degree as time goes on. We encounter the same phenomenon when observing the rate of triggered incidents by rules of a certain age, illustrated in Figure 3.13. The figure shows that the newest rules produce the most incidents per week, but this activity levels out as the rule gets older. Within several weeks, the rule ceases to produce as much incidents as in the moments closer to its inception. This can indicate an evolution of threats and vulnerability to novel threats, as well

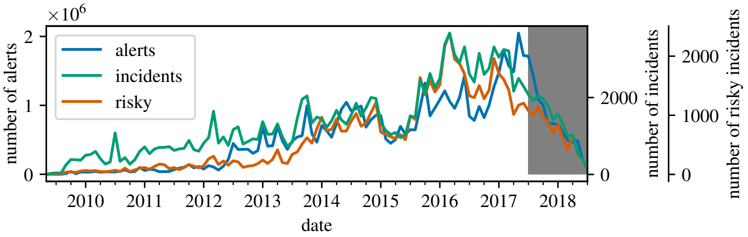


Figure 3.12: Number of alerts, incidents, and risky incidents handled by the SOC. The grayed out portion on the right-hand side of the graph indicates the MSSP's period of migration to a new logging platform. Note the secondary and tertiary axes.

as an adaptation to older threats by organizations as to no longer remain vulnerable to them.

Table 3.2: Number of alerts that compose the different types of incidents, and the corresponding number of distinct rules that triggered the alerts.

	#	Associated alerts	Distinct associated rules
Alerts	-	62,321,663	19,744
Incidents	150,437	735,262	6,720
True positive incidents	69,471	674,177	4,731
Risky incidents	13,589	157,388	2,618
Successful hack attacks	106	734	80

With the commercial rulesets being significantly larger (over 20 times as large), one could likely expect the alerts and incidents to be caused by the different rulesets in similar proportions. This is not the case, however, as we actually find that MSSP rules are overrepresented in many of the true positive incidents, highlighting type of "quality over quantity" philosophy. We calculate the precision of all rules in the different rulesets by dividing the number of true positive incidents of that rule by the total number of incidents in which that rule is present. The ET Pro and VRT rulesets have an average rule precision of 0.68 and 0.65, respectively, and the proprietary ruleset bests both with an average rule precision of 0.74, which clearly shows the higher utility of the MSSP's proprietary ruleset. Indeed, despite making up a fraction of all the rules employed by the MSSP, the proprietary rules are present in 27% of all true positive incidents. Thus, a smaller but contextualized ruleset adds significant detection capability, which supports the economic reasons that were mentioned in the interviews to develop the ruleset.

Most rules that cause false positive and low risk incidents are updated more quickly than the rest of rules. Specifically, with a median of 13 and 15 days, respectively, as opposed to the 30+ days for other incident categories. After the occurrence of false positive incidents,

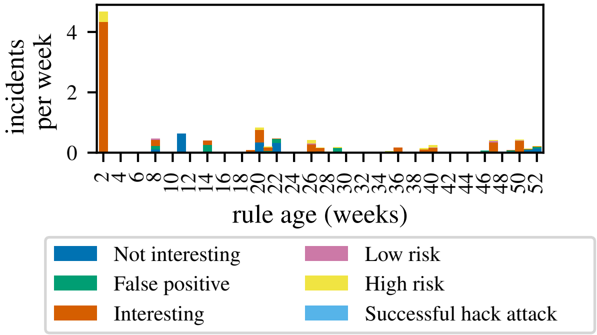


Figure 3.13: Average number of incidents per week triggered by MSSP rules of a certain age within the first year.

Table 3.3: Average updates per rule (to header or detection options) for different rule subsets.

Rules subset	# Rules	# Updates	Updates per rule	# Updates (detection options)	Updates per rule (detection options)
No triggered alerts	115,482	276,339	2.39	18,738	0.16
Triggered only alerts	9,988	370,417	37.1	1,768	0.18
Triggered only incidents	6,513	31,635	4.88	1,910	0.29
No triggered alerts (MSSP rules)	2,506	1,097	0.44	626	0.25
Triggered only alerts (MSSP rules)	447	576	1.29	432	0.97
Only TP incidents (MSSP rules)	276	244	0.88	124	0.45
At least one FP incidents (MSSP rules)	293	512	1.74	405	1.38

updates mainly fall into three different categories, namely bug fixes, threat intel updates, and making rules more specific (see Table 3.1 for the full list of categories), as is expected in case of false positive incidents. As for the low risk incidents, most of the rules updated within the first two weeks are updates made to the non-detection options, indicating that low risk incidents are no cause for concern and its corresponding rules are working as intended.

Since the MSSP also tests many of its own rules before adding them to the production environment, we examine its effects on the incidents triggered by these rules. The benefit of testing rules over a longer period of time is not immediately clear from the data. For all rules that trigger incidents and are tested before being added to the production environment, they remain in testing for a median of 11 days, with a notable outlier being the true positive-only rules that are tested for a median of 16 days. The fact that this subset of rules is tested for a longer period seems to indicate that this action has a positive effect on the quality of rules. However, only 21% of this true positive-only subset was tested at all before being added to the production environment.

In addition to examining the incidents themselves, this section will also investigate the effect of incidents on ruleset update behavior. Section 3.6 points out that the update behavior of alert-producing rules differs significantly from rules that never trigger throughout their lifetime. This section will expand on those findings and analyze the effect of incidents and their type on rule update behavior. In a similar vein as alert-producing rules, Table 3.3 shows that rules that trigger incidents are more likely to receive updates than rules that do not trigger at all. This holds for both types of updates shown in the table.

Updates made to the commercial rulesets are independent from the MSSP's alert and incident statistics; the MSSP does not influence the update behavior of the commercial rules. Furthermore, the datasets provided to us give us no way of knowing which commercial rules have been suppressed manually, due to, for instance, alert floods or false positives. When looking at the effect of incidents on update behavior, we therefore also focus on the MSSP's proprietary ruleset.

We find in Table 3.3 that false positive incidents are a much higher driver of updates. Rules that appear in at least one false positive incident are over twice as likely to receive updates than rules that only appear in true positive incidents. You don't change a winning team, and it seems the same goes for NIDS rules. This also makes sense from the perspective of a SOC. If a rule is causing valuable man-hours to be wasted on a fruitless investigation, it may be well worth it to make sure that rule performs as intended.

3.8. DISCUSSION AND LIMITATIONS

Our analysis has generated several takeaway findings. First, rulesets have grown over time and now a single sensor is running a combined set of over 65k rules. The focus in earlier research on the performance impact of large rulesets on an NIDS, seems to have disappeared as a concern. Only sporadically are rules purged from the set.

Second, notwithstanding the size of each ruleset, there is almost no overlap among them in terms of detection options or what indicators of compromise they contain. This suggests each ruleset covers a different and very limited fraction of the threat landscape, even those from leading vendors like Emerging Threats and Talos (formerly VRT). Thus, to maximize detection, all rules are combined and enabled by default, regardless of official recommendations against this practice [55]. This finding also underlines the economic reasons for the MSSP to invest also in in-house rule development. It is tailored to high-end threats relevant to their client base which the MSSP felt were not sufficiently captured by the commercial rulesets. The results bear out their intuition. While these proprietary rules make up just a tiny fraction of the entire collection of rules, they contribute disproportionately to the detection of incidents that are evaluated as posing a real threat.

Third, maintenance of the rulesets to improve detection is a minor activity. Most rules are produced once and then either remain untouched or they are modified in bulk for technical reasons, like changes in syntax or replacing the indicators from threat intelligence. In the small fraction of updates where the detection options are changed, the primary reason for the update is to make the rule more specific to the threat it is trying to detect, possibly in response to alerts flooding the SOC. Rules that trigger alerts more often are also more likely to receive updates. In rule management, as elsewhere, it seems that the squeaky wheel gets the grease.

Fourth, most rules do not contribute directly to detection. In fact, 85% of all rules never trigger a single alert in their lifetime. Only a minuscule fraction of these tens of thousands of rules are responsible for tens of millions of alerts: 0.5% of all rules generate 80% of all alerts. The overwhelming majority of these are noise. Only 1.2% of all alerts – generated by 5% of all rules – are deemed worthy of investigation by the SOC as incidents. And only a fraction of these investigated alerts – 0.3% of all alerts – turn out to carry legitimate risk to an organization.

3.8.1. INTERVIEWS

Our findings are supported and informed by the interviews with the security analysts who wrote or managed the rulesets. In terms of growing rulesets, the interviewees expressed that they did not really see a downside to keeping all the rules enabled by default. Even though older rules are much less active than newer ones, they are rarely deleted: "*Most of the rules don't really get out of production anymore, because it was proven that the rule works. [...] Maybe the [malicious] tool has new version and the old rule doesn't trigger on this new version, but you cannot say for certain that an actor wouldn't use the old version, so then it's better to have it active.*"

Most participants did not really see experience concerns for the impact of growing rulesets and continually adding rules to the NIDS: *"Unless it starts to flood [...] or becoming slow, maybe we don't even notice it. Then there's no process to evict this rule. [...] I probably think this rule will still be in the ruleset."* That said, some interviewees did express some concern, perhaps carried over from an earlier time, when performance impact might have been a factor. When asked about a computationally-expensive rule for the detection of a banking trojan, one interviewee stated: *"yeah, that's the trade-off, [...] if they already say the performance impact is high, I wouldn't deploy it."*

The lack of overlap – and thus threat coverage – of each ruleset was indeed a concern that led to the development of in-house rules: *"I do think you also need to have rules for [...] more advanced malware that maybe Emerging Threats doesn't really look into."* One analyst remarked that their proprietary rules make up *"thirty to forty percent of the cases we escalate, so these rules are so much better. [...] They are a really small part of our rule set but they have enormous impact."*

In terms of rule maintenance and modifications, one security analyst noted that: *"we tend to only delete rules if they're actually, like, flooding."* This fits with our finding that substantive changes to rules are often to make them more specific to the threat. Another analyst pointed to the fact that rules would be tested, before they are taken into production, thus reducing the need for later modifications: *"usually you give it like one or two days, or preferably a week, to run this rule [...] in testing."*

Finally, no interviewee had remarked on the fact that most rules never contribute directly to detection – i.e., never generating a single alert. This is understandable to some extent. No one would expect all rules to trigger alerts, so they would not evaluate rulesets that way. Lack of coverage would be much more critical than redundant rules. Thus, there is an intrinsic drive towards more rules, even if they never get triggered. All interviewees did remark on false positives. One analyst said that an ideal rule is one that is *"extremely specific for a specific type of malware [...] with as low amount of false positives as possible, while maximizing true positives."* This is an important trade-off, since most interviewees agree that a false positive alert flood is the most severe situation that can occur in a SOC: *"rule-wise, I think that is the worst thing that could happen: flooding of the SOC. Not only because you cannot handle the alerts anymore, but it could take the back-end down because the database* *is not responding."* There are no rules on how to make trade-offs like these. Instead, they depend on the experience of a developer. And the resulting outcome is much more skewed than the interviews led us to believe: nearly 99% of all alerts are never even investigated and only 0.3% are ever evaluated as indicating actual risk. This corresponds to 16% of all rules.

3.8.2. LIMITATIONS

The ruleset repository tool we developed contains a limitation that is inherent to its design. While it is able to accurately track additions, updates, and deletions from the ruleset, it is possible to introduce errors in its analysis. Since the tracking of rules across commits is based on a rule's ID, altering the ID of an already existing rule is counted as an addition of a new rule with the new ID and a removal of the rule with the old ID. While this does affect the computation of total additions and deletions over time, it does not affect the ruleset's total rule count. Through manual examination, we are able to estimate that of the approximately $8,800$ modifications made to the proprietary ruleset, only around 100 have been ID alterations. Rule IDs are meant to be unique identifiers, and thus this limitation is an unfortunate symptom of ruleset mismanagement.

The manner in which the alert data is collected affects the result of the analysis. Specifically, the size of the customer base affects the distribution of alerts over rules. As the customer base of the MSSP grows, the same rules will trigger more often, and the more skewed the distribution will be (see Figure 3.10). This does not give a realistic view into the alert behavior within a single organization. However, we have approached this analysis from the perspective of the MSSP, not a single organization. Such an issue is inherent to an MSSP and undoubtedly plays a role in the processes that go into the management of the different rulesets.

3.9. CONCLUSION

Traditional NIDSs and their rulesets are an often overlooked portion of network security. This work presents the first study that aims to shed light on this cornerstone of network security.

After analyzing four different NIDS rulesets containing around 130 thousand rules, we find that the vast majority of rules fail to produce a single alert, i.e., 80% percent of all alerts were triggered by a mere 0.5% of all rules. However, this does not pose a problem for the SOC analysts as rule developers keep adding new rules and barely modifying the existing ones. In fact, only around 23% of all rules are updated in terms of detection capability, with primarily two objectives: (i) adapt to changes in the threat landscape; and (ii) reducing the number of alerts and false positives by making rules more specific. Hence the possibility of using large rulesets without overwhelming SOC analysts. Just 1.2% of all alerts were deemed important enough to be investigated by the SOC, and only 0.3% of all alerts carried significant risk to the organization.

We also identified a set of common rule management practices that include: (i) using multiple rulesets simultaneously due to the lack of exhaustive coverage of the threat landscape by any single ruleset; (ii) creating proprietary rules to cover client-specific threats and updating these with higher frequency than commercial rulesets; and, (iii) reducing false positive incidents by updating the rules that triggered the corresponding alerts.

Finally, our analyses allow us to conclude that the rumours of the death of signature-based monitoring were greatly exaggerated. Contrary to what appeared to be popular opinion, the findings in this chapter seem to indicate that signature-based NIDSs and their rulesets remain vital in network security. Future work is needed to compare this with different approaches, such as host-based detection in the form of end-point monitoring. With such analysis, we move closer to determining the fate of traditional signature-based systems: is it an archaic and obsolete technology or does it remain an indispensable part of a secure network?

4

NIDS RULE AND INCIDENT MANAGEMENT PROCESSES

Signature-based network intrusion detection systems (NIDSs) and network intrusion prevention systems (NIPSs) remain at the heart of network defense, along with the rules that enable them to detect threats. These rules allow Security Operation Centers (SOCs) to properly defend a network, yet we know almost nothing about how rules are created, evaluated and managed from an organizational standpoint. In this work, we analyze the processes surrounding the creation, management, and acquisition of rules for network intrusion detection. To understand these processes, we conducted interviews with 17 professionals who work at Managed Security Service Providers (MSSPs) or other organizations that provide network monitoring as a service or conduct their own network monitoring internally. We discovered numerous critical factors, such as rule specificity and total number of alerts and false positives, that guide SOCs in their rule management processes. These lower-level aspects of network monitoring processes have generally been regarded as immutable by prior work, which has mainly focused on designing systems that handle the resulting alert flows by dynamically reducing the number of noisy alerts SOC analysts need to sift through. Instead, we present several recommendations that address these lower-level aspects to help improve alert quality and allow SOCs to better optimize workflows and use of available resources. These recommendations include increasing the specificity of rules, explicitly defining feedback loops from detection to rule development, and setting up organizational processes to improve the transfer of tacit knowledge.

4.1. INTRODUCTION

Security Operations Centers (SOC) are a key part of defending enterprise networks against attacks. Located inside these networks are a variety of security tools and systems – such as Network Intrusion Detection Systems (NIDS) – that generate alerts. Analysts in the SOC are burdened with triaging the deluge of alerts that consist mostly of false positives [8]. For years, academic research and industry reports alike have raised the problem of alert fatigue and analyst burnout [180, 221]

A lot of academic research on improving SOC operations has pursued better automation – most notably using machine learning (ML) to analyze the alerts and generate more informative alarms for the analysts to investigate. Indeed, an "insufficient automation level of SOC components" was considered the top issue by SOC managers, according to a recent study [120]. Prior work includes human-subjects studies with stakeholders, aiming to understand the problems in the SOC workflow to handle alerts [8, 120, 221, 222]. To address workload issues using ML, studies have focused on alert prioritization [94, 171, 244] and false positive detection [215, 244]. While the promise of ML for SOC operations is clear, adoption has been low, and solutions have under-delivered on the promise [60, 110].

The attempts to use ML for improving SOC operations have in common that they are "end of pipe" solutions: they take the flow of alerts as a given and build a system that ingests this flow in order to generate meaningful information that supports analysts in triaging the alerts. Alahmadi et al. [8] aim to improve the development of such ML solutions by identifying properties that the output of the ML, "alarms," should have to be useful to analysts. Ex-post, these systems are designed to take low quality alarm noise as input in order to produce informative and actionable output that SOC analysts can use. Well known within the ML community is that inadequate training data will have a detrimental effect on the quality of the resulting model, as is exemplified by the familiar saying "garbage in, garbage out" [195]. And even though their work focuses on alleviating the limitations of alarms through the use of ML-based tools, all of the mentioned limitations are also closely related to the features that determine the quality of a rule. By improving the quality of the rules, the quality of triggered alerts will also improve, thereby also bettering the quality of the input for whatever ML system uses this data.

Thus, we propose a different and complementary approach: rather than end-of-pipe, we focus on the source of the problem and want to understand how to improve the quality of the alerts that go into the pipe. Where do the alerts come from? Typically, they are generated by systems that compare the network traffic against a set of rules or signatures. As early as 2010, experts had forecast the demise of rule-based detection [39, 240] and focused their research on statistical and machine-learning approaches – e.g., [148, 209, 220]. So far, however, the predicted death of rule-based approaches has not materialized. More than a decade later, the industry still predominantly relies on rules for generating the alerts that are the basis for SOC detection of attacks.

The craft of developing effective rules is, thus, critical to producing meaningful alerts and enabling detection, also for those solutions that rely on “end-of-pipe” ML. Yet, we lack understanding as to how practitioners in SOCs design rules for processing network activity and how an alert is established and evaluated as being effective in practice. Rule development and management take place with limited knowledge, such that alerts require investigation to determine if a threat exists on the network. Most rules never trigger any alert [239]. Of those that do, the overwhelming majority of the alerts are false positives. True positives are scarce and false negatives might remain hidden. Under these constraints, how do rule developers figure out what is a good rule versus a bad rule? How do they evaluate the quality of the overall ruleset? What practices do they follow in producing rulesets that are effective in detecting attacks?

In this paper, we present the first interview study focused on professionals who develop or revise rules used in network threat detection – as the input for SOC incident reporting systems, rather than how the analysts manage the outputs of these systems. We aim to answer the following questions: (1) What does the organizational ruleset management process look like?; (2) What are the main factors and success criteria in managing and evaluating NIDS rules and rulesets?; (3) How can security professionals improve rule management and network incident monitoring workflows to optimize SOC processes?

Between June 2020 – March 2022, we interviewed 17 professionals who have developed NIDS rules, and who either work at Managed Security Service Providers (MSSPs) providing network monitoring as a service, or at government agencies and firms that conduct their own network monitoring. Some of the rules they work with were manually developed in-house, some were commercially acquired, and some were shared within their community. While our interviewees have different roles, inside and outside SOCs, they all are mandated to develop or change the detection rules running in their production environment.

While rules are designed to various degrees of quality, we found that the true test of quality only emerges after the rules are deployed in production. Such evaluation is mostly based on total number of generated alerts and false positives. Most quality criteria are optimizing for the SOC analysts’ workload, rather than for detection – in other words, for reducing false positives rather than false negatives. The aspects of NIDS rules that determine their quality are balanced against each other to match the organizational processes and resources. In fact, SOC teams appear to work in a delicate equilibrium, operating on a knife’s edge, where their resources are in balance with the workload and the services they offer to clients. They provide ‘enough’ security to keep clients content, and, given their specific organizational processes, more resources would not necessarily result in a ‘better’ product. In terms of necessary improvements, we were unable to identify common issues, as there was no consensus between respondents regarding any critical challenges; most conversations revolved around specific aspects of the employer’s organizational processes. This emphasized the importance of understanding analysts’ workflows. Contrary to recent work on SOCs, no interviewee mentioned automation to replace rules – that is, using AI or machine-learning approaches like anomaly detection – as an important direction that they believed would improve their work.

In sum, our main contributions are:

- We present the first interview study focused on professionals who make active use of – i.e., develop or revise – rules for network detection, and how they experience and perform their specific rule creation and management processes within their organizations;

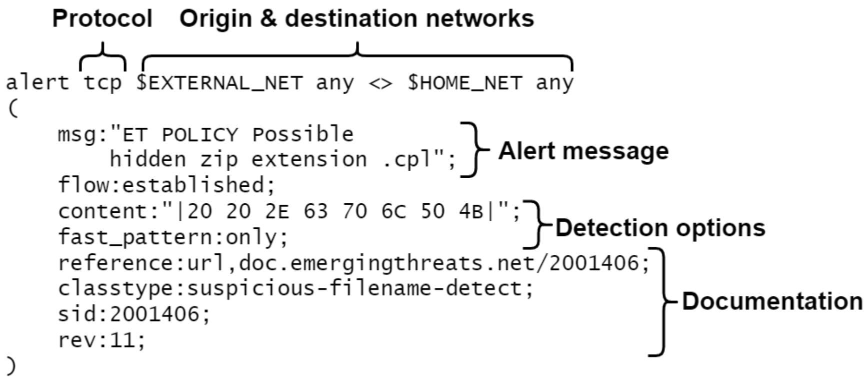
- We find that there is no one trivial way in which to manage a SOC; many critical factors in the function of a SOC (such as rule specificity and false positive thresholds) must be balanced against other equally critical factors. This speaks to the value in understanding how these decisions are made in context, and providing evidence to inform decisions such as rule management and SOC resource allocation;

- SOCs balance resourcing and capabilities against customer expectations, wherein we evidence alternatives to a traditional approach to security of ‘more is better’. Examples include security tailored to specific threats and environments instead of blanket coverage of the global threat landscape;

- External (e.g., commercial) rulesets create a negative externality for its users: while they are designed to provide broad threat coverage, users incur the costs of deactivating or fine-tuning individual rules, or manage the noise they often generate by them if they opt not to do so;

- There is barely any feedback loop in place for handling false negatives. Since SOCs cannot know when and how an incident will be missed, and proactive ‘horizon scanning’ of threats is an effort to compensate for this shortcoming;

- We present a number of recommendations for internal SOCs processes that can help with improving the overall effectiveness of SOC teams and the services that they provide, with a focus on making improvements at an earlier stage of the network monitoring and incident response process.



```
alert tcp $EXTERNAL_NET any <> $HOME_NET any
(
    msg:"ET POLICY Possible
        hidden zip extension .cpl";
    flow:established;
    content:"|20 20 2E 63 70 6C 50 4B|";
    fast_pattern:only;
    reference:url,doc.emergingthreats.net/2001406;
    classtype:suspicious-filename-detect;
    sid:2001406;
    rev:11;
)
```

Figure 4.1: Example of a rule used by Snort and Suricata NIDSs. Rule developers document the purpose of the rule and background information related to the threat in the alert message and “documentation” blocks, respectively. The “detection options” block is used by the system to detect the actual threat.

4.2. BACKGROUND

4.2.1. NIDS, NIPS, RULES, AND RULESETS

Intrusion detection systems (IDSs) and **intrusion prevention systems** (IPSs) are categorized based on where they are placed and their methods of detection. There are network-based, host-based, and application-based systems. Furthermore, they can be either signature-based or anomaly-based depending on how they detect threats [146].

A network-based IDS (NIDS) is positioned at a key point in the network and scans all incoming network packets for signs of attacks. By comparing the content of packets with a predefined collection of patterns or events that are typical of known attacks, signature-based IDSs identify malicious activity. **Network-based IPSs** (NIPSs) are installed inline with a network, which gives it the ability to actively block traffic in case of malicious activity detection. Examples include Snort [53], Suricata [223], and Zeek [182].

These systems detect threats in network traffic using **rules** that tell the system precisely what to look for in network traffic that might be indicative of malicious activity. Using these rules, analysts can specify characteristics of network traffic such as origin and destination IP addresses, protocol used, packet payload content, among others. Additional documentation and metadata may be added to the rule to improve manageability. See Figure 4.1 for an example of such a rule. A collection of rules are grouped together into a **ruleset** and deployed on an IDS/IPS.

When traffic meets the conditions specified in a rule, it triggers an **alert**. This alert is then typically sent to an analyst to be validated as an actual attack or threat and of what kind. These analysts can be in different locations, but they typically are in SOCs. Some organizations operate their own SOC, some have it outsourced to an MSSP. A SOC is confronted with an alert flow that can quickly go into the thousands of alerts per day.

Rule developers are professionals with the mandate to write, change or remove rules. They might be analysts in the SOC who, based on the alerts they see, change the rule to provide more useful signals – i.e., make the rule more precise to avoid legitimate traffic triggering the alert. Other rule developers work outside the SOC. At MSSPs, for example, they might be located close to the threat intelligence department, to craft new rules in light of attacks observed elsewhere. And some rule developers work at companies that sell a whole set of detection rules as a service, e.g., Proofpoint's ET ruleset [186] and Cisco Talos' ruleset [54]. Often, these commercial rulesets consist of tens of thousands of rules [239]. Organizations, including MSSPs, subscribe to these rulesets in order to complement the rules they develop themselves. They can then mandate people in their own organization to assess, deploy, change or remove rules from these commercial sets, depending on what alerts they trigger.

4.2.2. Related Work

Shutock and Dietrich consider people, process, and technology in SOC management [211]. They enumerate current challenges in operating a SOC, including staffing issues, and outsourcing of SOC capabilities from within organizations to external MSSPs. The authors discuss platform consolidation against the converse view of SOCs maintaining their own internally-developed tools – we find that there are factors relating to business offerings and client needs which inform these choices, beyond the amount of effort involved.

Kokulu et al. [120] interviewed 18 SOC analysts and managers, with a view to both technical and non-technical challenges. An argument is that current SOC arrangements are insufficient to counter current threat levels and that they are failing to operate sufficiently well; further, the authors presume that there are practical problems within SOCs that can be identified and addressed to improve their capabilities. Among their findings are that false positives in malicious activity detection do not majorly impact SOC operations. The authors included budget-related questions for managers, Identified SOC issues were framed according to whether they were perceived by one of the analysts or managers, or both (mismatched or matched). Among matched issues were 'poor quality reports and logs', and 'high false positive rate'. Of note is that managers in the Kokulu et al. study stated that they had 'sufficient' budget to operate their SOC, where the authors identify 'insufficient budget' for SOCs in terms of managers being very aware of having limited budget which can impact their capacity to enact change programs, training or travel.

Sundaramurthy et al. [222] apply Activity Theory in a long-term (3.5 year) observation of how SOC professionals work together to satisfy goals and objectives. This uncovered tensions and contradictions, specifically issues with tools and operating rules.

Alahmadi et al. [8] focus on false positive alert generation in SOCs, examining this persistent challenge through interviews with 20 practitioners. The authors arrive at five indicators of quality for alerts: reliable, explainable, analytical, contextual, and transferable. An approach to improving alert quality is taken as opposed to a general focus elsewhere to find tools to reduce the volume of alarms.

We agree with Alahmadi et al. [8] that improving alert quality is urgent and critical to better detection and SOC performance. However, their call also underlines that so far, the prior work has focused on how to handle the flow: helping analysts to triage and validate the voluminous influx of alerts into false positives, true positives, and more fine-grained distinctions. It has not investigated how that flow is generated, namely via detection rules. To improve the quality of alerts means improving the quality of the rules that generate these alerts. How professionals aim to do that is the focus of our study.

Basyurt et al. [21] conduct interviews with nine SOC practitioners to uncover challenges that SOCs face when collecting and analyzing data, as well as communication cyber situations. The challenges that the authors identify are of a technical nature, such as the collection and compilation of data sources and trustworthiness assessments of information, and suggest the development of a tool that is able to automate those tasks. As opposed to Basyurt et al., we lay the focus more on organizational challenges.

4.3. METHODOLOGY

Here we detail the structure of our interview study, along with our recruitment activities and approach to analysis.

4.3.1. STUDY DESIGN

Our overarching research question is, “How do security professionals manage network incident monitoring processes to achieve security?” We addressed this through interviews with professionals. Interviews have been used in prior studies to understand challenges around management of provisioned security (e.g., [120]).

Our interviews were structured to capture the following information, as also detailed in the question set itself (see Appendix A.2):

1. Participant details, including their role, organization, and qualities of the services they provide;

2. Analyst's organization services and workflows, focusing on how detection rules are constructed and managed on a regular, day-to-day basis;

3. The specific processes analysts use for ruleset evaluation;

4. The management of rules, including collaboration with others in the organization, and related responsibilities for assessment and corroboration of evidence around rule-related decisions;

5. How analysts meet objectives in practice, including views on improvement.

At the beginning of the study, we performed pilot interviews in order to test our interview protocol. Minor adaptations were made to the phrasing of some questions. An additional sample rule for evaluation was also added.

4.3.2. Recruitment and participants

We recruited professionals to participate in interviews from a range of different organizations providing managed security services (similar to [8]). This allowed for the comparison of working practices and decision-making around rulesets and their management, etc.

The initial seven participants from Org1 (see Table 4.1) were recruited through snowballing—the process of asking from each participant a short list of names of other people within the company's NIDS rule development and management teams whom they believe to be relevant to this study. This process was halted when the only names we received were professionals whom we had already interviewed before and two participants who did not respond to our invitations. After the initial batch of participants, we encountered similar recruitment challenges as Alahmadi et al. [8], with it being challenging to recruit from a profession wherein the occupation requires individuals to almost always be active or available for work-related activities.

Due to the hour to 1.5-hour duration of this interview, and the time constraints and daily task requirements that analysts work with, we were unable to find many more participants at the analyst level. Instead, the majority of the participants in this study, after the initial seven, are security professionals at a more senior level.

We leveraged our research team's institutional relationships to contact the remaining 10 participants, either through direct emails or through open invitations on LinkedIn. The prerequisite for participants was that they must be involved in the processes surrounding the creation or management of NIDS/NIPS rules, be they analysts or managers.

The participants in this study come from nine different organizations across five different sectors. Not only do they differ in sector, but also in size. See Table 4.1 for an overview of the participants. Although we cannot disclose specific characteristics of these organizations for reasons of confidentiality, their sector may provide an indication of the scale of the networks that they manage.

4.3.3. ETHICS

Before starting our research, we followed the ethics approval and research data management procedure outlined by our institution. Data Management Plan has been approved for this study to ensure accountability, transparency, and compliance. We also followed the principles of the Menlo Report of ethics for ICT Research [117] during the study. This included respondents explicitly consenting to the recording, transcription of the interview, and to the usage of quotes, as well as being informed about their options as to their participation in the study. We minimized the risks of data leaks by pseudonymizing all data gathered during the interviews. The quotes have been assessed by the team members regarding the risk of reverse identification and de-pseudonimization of research participants. The recordings were stored for the duration of this research on an encrypted hard drive and destroyed when it was no longer necessary to keep them. All answers were confidential and only available to the researchers involved in this project.

4.3.4. DATA ANALYSIS

Interviews were transcribed, after which thematic analysis [37, 38] was conducted with the transcripts. This involved a process of qualitative coding – three coders of diverse backgrounds were involved in the codebook development, and regular codebook review meetings to arrive at a final codebook of cross-cutting themes that emerged from the interviews (and interviewees) themselves. For the quotations used, reverse identification checks were conducted by the researchers to safeguard the anonymity of research participants.

The thematic analysis resulted in the following themes emerging: (1) services offered by the organizations and associated workflows; (2) rule evaluation; (3) ruleset evaluation; (4) internal and external collaboration processes; and (5) desired points of improvement. The subsections in the next section (Results) are arranged according to these themes, elaborating on the views that emerged from the interviews according to these cross-cutting areas. We present the prominent codes under each code theme, representing the strongest points of discussion across our participant group.

Table 4.1: Overview of all participants, their corresponding organization and business sector, and their role and experience within that organization.

ID	Org. ID	Sector	Role	Experience
P1	Org1	Cybersecurity	Analyst	4 years
P2	Org1	Cybersecurity	Forensics	3 years
P3	Org1	Cybersecurity	Analyst	3 years
P4	Org1	Cybersecurity	Senior security expert	14 years
P5	Org1	Cybersecurity	Incident response	6 years
P6	Org1	Cybersecurity	Incident response	6 years
P7	Org1	Cybersecurity	Analyst	6 months
P8	Org2	Cybersecurity	Analyst	17 years
P9	Org3	Government	Analyst	5 years
P10	Org4	Cybersecurity	Senior security researcher	16 years
P11	Org5	Shipping	SOC manager	7 years
P12	Org6	Telecom	Senior analyst	8 years
P13	Org7	Consultancy	SOC manager	4 years
P14	Org8	Government	SOC manager	15 years
P15	Org8	Government	Project architect	19 years
P16	Org9	Consultancy	Senior analyst	15 years
P17	Org9	Consultancy	Analyst	1.5 years

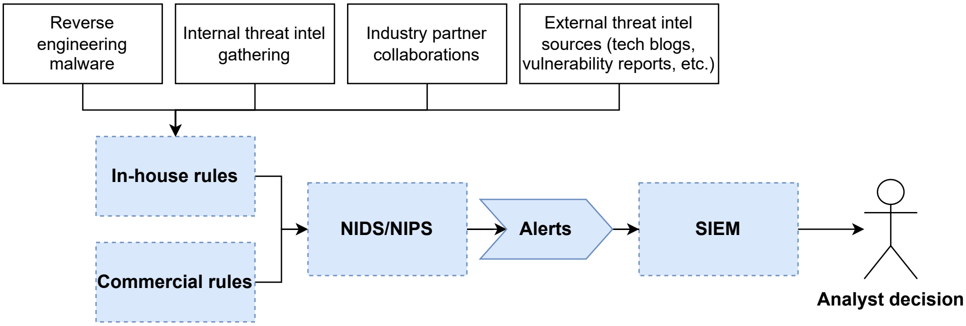

Reverse engineering malware
Internal threat intel gathering
Industry partner collaborations
External threat intel sources (tech blogs, vulnerability reports, etc.)
In-house rules
Commercial rules
NIDS/NIPS
Alerts
SIEM
Analyst decision

Figure 4.2: Simplified version of the NIDS/NIPS pipeline employed by organizations. In-house and/or commercial rules are installed on an NIDS or NIPS, which produces alerts that are used as input by SIEM (tools). Analysts then make decisions regarding detected threats based on the output of the SIEM tools. Illustrated are also the different data sources that organizations use when creating their in-house rulesets.

4.4. RESULTS

The interviews revolved around four main topics: (1) the different workflows within the organization that make up its incident monitoring and response services, (2) the different processes and objectives that surround the management of NIDS rules and rulesets, (3) how the participants collaborate with peers and clients to carry out said processes and objectives, and (4) points of (dis)satisfaction regarding work processes expressed by the participants.

In the sections below, we specifically use the term “alert” to refer to the notifications generated by NIDS/NIPS rules. This is in contrast to the term “alarm”, which is used by related literature to refer to the notifications generated by SIEM (security information and event management) tools “as a result of the correlation of multiple alerts” [8]. This is illustrated in Figure 4.2: the NIDS/NIPS generates alerts, which are used by the SIEM as input.

4.4.1. ORGANIZATION SERVICES AND WORKFLOW

To understand the culture and workflows that govern the SOC, we first set out to identify the different aspects that make up the security services each SOC offers. This section will discuss the results that describe initial client interactions and on-boarding, ruleset management and development practices, and type of network monitoring offered by the SOC (i.e. NIDS or NIPS).

CLIENT ON-BOARDING

All organizations but one mention client on-boarding procedures, whereby the organization gathers information about a client's network before active network monitoring begins. This allows SOCs to calibrate the network sensor output to the resources available to the SOC and familiarize themselves with the client network. It can consist of in-person meetings and preliminary alert processing, which includes setting up the network sensor infrastructure, and analyzing alerts without performing direct incident response. Gathered information can include (sub)network descriptions, most valuable assets, and non-standard software running within the network. The exception to this is the shipping organization Org5, which does not monitor its clients' networks; instead, it monitors its own network, which it makes available to all of its clients. However, Org5 does ensure that the NIDS/NIPS rules that monitor their network do not interfere with the functionality of client software interacting with the network.

Participants mentioned the lack of a formal time limit set for this on-boarding procedure, although they all indicate a usual duration of several weeks. Seeing as the time limit for this phase is not formally defined, it likely varies from client to client, and it is up to the analysts to determine when the network sensors and rules are properly configured to the client network.

Notable on-boarding procedures exist at Org3 and Org9. Instead of having a single ruleset that is used for all client environments, these two organizations create a custom ruleset in conjunction with their clients that are completely tailored to the networks of said clients. Similarly, Org7 creates custom rulesets for each client, although with much less client involvement.

RULESET MANAGEMENT AND DEVELOPMENT

Ruleset types can be split into the following: free external (i.e., community), paid external (i.e., commercial), and in-house rulesets. Both types of external rulesets can be obtained from threat intelligence vendors, and can contain tens of thousands of rules [239]. In-house rulesets vary from a few tens to a few thousands, depending on the size and resources of the organization, and its business model.

All organizations except Org3, Org4, and Org9, use some sort of external ruleset. Although there is considerable diversity regarding the usage and management of external rules, there certainly is a consensus around their quality. All organizations that use such rulesets criticize the poor quality of rules, which leads to a large amount of noise and many false positives (Org1, Org2, Org5, Org6, Org8). However, participants also admit the added value that these external rulesets provide: *"I do think you need to cover all types of malware, also like the new ones that are coming out. So that is what I think we have covered by [the external ruleset]" (P14).* Though the value of external rulesets is recognized, many participants also noted the noisiness that these rulesets tend to produce: *"We tend to find that they're [...] extremely noisy in our environment. Just little things can set them off. (P11)"*

P10 explains that it is due to this noisiness and tendency to produce many false positives that Org4 refrains from using such external rulesets. Since Org4 operates an inline NIPS, packets are dropped when they trigger a rule. Having a rule erroneously trigger on legitimate traffic then becomes a more significant problem. Floods of false positives will cause much legitimate traffic to be dropped, negatively affecting their clients' business continuity. Org5 operates a hybrid system, where rules can be set to either just alert when triggered or drop the potentially malicious packet. While this system is also set up inline to be able to block packets, this allows them to switch blocking rules to alerting rules in case of sub-optimal detection performance (e.g., too many false positives) so as to not affect business continuity in the same manner as Org4.

P8 mentions that free NIDS rulesets are not used at all or disabled immediately due to the large number of false positives it produces. P8 also claims that these rulesets *"generally look at non-conformity towards RFCs (Request for Comments),"* and since many applications on the Internet *"bend the RFCs"* to properly work, using such rulesets will easily flood a SOC with alerts. P8 is also disillusioned by the detection performance of commercial rulesets. Many rules contain IP addresses and domains of known C&C servers, for instance, that are out of date and are no longer used for malicious activities, and can therefore trigger many false positives. *"Those are generally, for 99% out of date, and they will trigger way too much false positives"* (P8). P8 continues: *"I will come across indicators of compromise easily three or four years old still in a rule set so, and from a paid service you'd expect more."* P11 shares a similar opinion: *"we preach about it a lot in our environment that the [IP and domain] indicators are worthless."* Thus, an organization that decides to employ external rulesets must design its operations such that dealing with that amount of noise and false positives is feasible for the technologies and the manpower available to it.

Due to this lack of quality, many organizations build their own in-house ruleset. All organizations except Org8 indicate the use of a custom ruleset crafted in-house. Such in-house rulesets can be built and tuned to the specific requirements and workflow practices of each organization.

In-house rulesets are used by Org1, Org5, and Org6, in a fashion complementary to external rulesets. As P4 states, the external rulesets—specifically the commercial rulesets—are made by a bigger team and thus generate more general coverage, allowing Org1 to focus their own rule development efforts on more severe and potentially targeted threats that are not covered by the external rulesets (which may focus more on, e.g., botnet traffic, untargeted commodity attacks). Org3, Org4, Org7, and Org9 forgo external rulesets completely to avoid the noise and false positives it can potentially produce. Instead, they rely solely on their own in-house ruleset, opting to delegate the security provided by the external rulesets to another layer in their defenses. Even though an in-house ruleset is currently not employed at Org8, both participants from Org8 (P14 and P15) acknowledge the importance of such a custom ruleset and share the plans of their organization to incorporate this aspect into their workflow in the near future: *"we found out that it's really important to separate things, separate the rulesets from the engines to be able to always [...] use your own rules"* (P15).

Org3 and Org9 create a custom-made ruleset for every client. The rulesets created by these two organizations are particularly small (under 100), especially compared to the rulesets employed by the rest of the organizations, and are developed to detect threats specific to the network and threat landscape of a client. The rest of the organizations employ custom rulesets with a volume that can run up into the thousands of rules (tens of thousands if external rulesets are counted). Different data sources are employed by organizations to create their in-house rulesets, as illustrated in Figure 4.2.

NETWORK INCIDENT MONITORING

All but one of the organizations uses an NIDS for network monitoring. Org4 is the only organization that employs an NIPS. P14 mentioned that Org8 had recently made the switch from running NIPSs to NIDSs due to issues with latency (as the system is set up inline with the network) and false positives dropping legitimate traffic. As a result, their rule management procedures and evaluation criteria needed to be adapted to this new form of network monitoring. This is described in Section 4.4.2.

4.4.2. RULE EVALUATION

Rules that are developed and put into operation require a level of quality that makes them effective at securing the network. As different organizations have different processes for securing networks, there cannot be a universal definition of what a quality rule is. Still, analysts from distinct organizations look for these attributes in much the same ways in order to assess the quality of rules. This quality control is done in two stages: pre-deployment,

Table 4.2: Criteria used by participants to determine the quality of rules.

Rule quality criteria	**Participants**	**Percentage**
Rule design		
Rule specificity	17	100%
Syntax and structure	6	35%
Dislike of domain and IP blocklists within a rule	4	24%
Usage of the fast_pattern keyword in rules whenever possible	4	24%
Specifying origin and destination networks	3	18%
Rule performance		
Number of total alerts triggered by a rule	17	100%
Number of false positives triggered by a rule	16	94%
Danger of resource intensive rule	4	24%
False negatives	3	18%
Ruleset evaluation		
Volume and resource intensity	17	100%
Legacy rules	17	100%
Coverage, of which	15	88%
- threat coverage for a specific environment	7	41%
- overall threat coverage	2	12%

during rule design, and post-deployment, when the detection performance of a rule is evaluated after adding it to the production environment. This section will elaborate on the different factors that the participating organizations take into account when determining the quality of a rule. An overview of the different criteria can be found in Table 4.2. Our findings identify contributing factors ahead of the impacts that analysts have raised as challenges in the work of Agyepong et al. [4], specifically volume of alerts, false positives, manual and repetitive processes, and workloads.

Rule design

Much of what a SOC "sees" originates from the rules that are installed on NIDS systems. How these rules are designed largely determines the information that a SOC receives about potential threats. Organizations have no say in how external rules are designed; that is entirely within the purview of the rule vendors. As a result, assessment of external rules is limited to post-deployment evaluation of detection performance – options are limited to switching off rules rather than modifying external rules (as ruleset updates undo those modifications). For pre-deployment assessment, organizations can only focus on how they design their own in-house rules. Analysts and rule writers often go through research, experimentation, and multiple rounds of testing for every rule that they design. After conducting the interviews, we identified characteristics of a rule that influence its potential quality.

Specificity. The single most important aspect of rule design is to make the rule specific to the threat for which it is being designed; all participants make mention of the importance of this quality criterion during their respective interview (see Table 4.2). A rule that is specific to a particular threat will, for instance, try to detect the threat using some very unique characteristic that make it easy to distinguish from other types of traffic. By contrast, a rule that is more generic will use indicators that are less tied to a particular instance of a threat, in order to capture more variations of said threat. As such, a generic rule will likely trigger more often than a more specific one.

As there can be many exploits that target the same vulnerability, and malware families using these exploits can become quite large, creating a rule for every single instance quickly becomes infeasible: *"I'm going to write as few rules as possible to detect as many web shells as possible. Rather than creating [...] static checks for each and every one of them, [...], you try to find a pattern between all these web shells and see if you can make one rule to cover maybe 80% of all the web shells out there, and then see if you can make it work"* (P6). Therefore, these two aspects must be balanced when developing rules. P10 explains: *"you want to be as specific as you can to prevent false positives, but you don't want to be so specific that you're going to miss different variants of a piece of malware or something other than the POC (proof of concept)."* Create a rule that is too specific, and it might not trigger on different versions of the same threat; create a rule that is too generic, and you might overload your analysts and SOC: *"in practice [it can be] really difficult [...] to actually get really good detection rules that apply to the threats but do not generate false positives, and it's this balance that we need to keep up every day. Sometimes we get too strict and you don't see things, sometimes we write and update the rules, then we could get called by the 24/7 team that is getting a lot of noise in the monitoring system because the rules apply to a lot of traffic"* (P12).

Indeed, the aspect of specificity does not exist in a vacuum. It not only has an effect on the potential false positive alerts, but also on the number of alerts in general. This will be discussed in Section 4.4.2.

Rule syntax and structure. Six participants mention a rule's syntax and structure as an aspect that influences quality. These aspects determine the readability of a rule because it makes maintenance of the rules easier (P4, P10, and P11). Three participants (P2, P4, and P6) mention the importance of using variables to specify the origin and destination networks of the traffic being analyzed by the rule. Configuring a rule to be executed for any origin and destination IP will increase the resource intensity of the rule, since the rule will be applied to every packet entering and leaving the network. Specifying origin and destination networks or addresses for a rule ensures that the rule is applied only to relevant packets, thereby reducing its resource consumption. Additionally, four participants (P1, P4, P6, and P10) mention the usage of the `fast_pattern` keyword in rules whenever possible. With this keyword, an analyst can tell the system how to perform more efficient pattern matching for the potentially malicious content of the packet, which can also greatly increase the speed at which the rule is executed.

Blocking IPs and domains. Finally, four of the participants also displayed some animosity towards the use of domain and IP blocklists within the rule, not only due to the resource intensity of these types of blocklist rules, but also because they are difficult to maintain and quickly become out of date, making them easily prone to false positives.

Although looking at the rule itself can provide an indication of its quality, it by itself does not provide enough information for a definitive verdict: *"it's very hard to classify a rule as good or bad based on only the rule itself, so you really need to [...] have the rule tested on real network traffic"* (P1).

Rule performance

All participants share a similar opinion regarding characteristics that determine rule quality. Interestingly, the only element on which all participants unanimously agree is the significance of the number of alerts that a rule triggers. The number of false positives a rule triggers is a close second, with all but P15 making a mention of this second aspect. The degree to which the remaining characteristics are deemed important differs depending on the workflow practices of the organizations.

A rule that triggers an abundance of alerts, and thus potentially floods the SOC, was stated by all participants to be undesirable. Analysts need to process every alert that arrives at the SOC, and an alert flood interferes with the analysts' ability to perform their work effectively. Furthermore, a rule that starts producing a very large amount of alerts might also overload the SOC systems, causing downtime, leading to the SOC not being able to monitor their clients' networks at all. Indeed, most participants deem flooding of the SOC the most severe failure that can occur within their operation.

An example of when this can go wrong comes from a few participants from Org1: *"So [our rule vendor] created a rule. They found out that SMB1 traffic was now deprecated, and should be considered malicious because it can be exploited. So, basically, they said 'if SMB1 traffic occurs, then it should create an alert, because that is a really bad sign.' And in some way it is. However, there are a lot of companies who still use that legitimately. So basically what happened when they introduced that rule and we synced it into our production environment, [is that] it broke everything"* (P5). P4 elaborates: *"The flooding of the SOC is not only bad because you cannot handle the logs anymore [...], but also it could take the* *backend down because of the database not responding [...]"* Ways to safeguard against such failures include mandatory testing of rules on real network traffic before being placed in production, as well as syntax checks to ensure that rules only trigger a set amount of times within a certain period.

P3 describes the Org1's heuristic process of creating "good" rules that takes the amount of triggered alerts into account. Apart from triggering on true positives, with that being the ideal scenario, another gauge of rule quality is the *lack* of triggers while being tested on customer network traffic. It is difficult to test a rule's true positive rate, since there is no guarantee that it will detect specific malicious traffic during its testing period. And depending on the rarity of a threat, many of the alerts will turn out to be false positives. So instead of going through the difficulties of measuring a rule's true positive rate, rule developers can look at the *absence* of false positives to estimate the quality of a rule. This heuristic is clearly not a one-size-fits-all and includes implicit assumptions about the severity of the threat in question, since many (potentially less severe) threats are much more accessible and present in the wild, and will inevitably trigger these less severe rules when exposed to real network traffic (e.g., port scans). *"Another good indicator [of rule quality is the following.] We [push] the [rules in our] testing repository [to several of] our biggest customers [for] a reason: that is because if it doesn't trigger there, then the rule is probably well written. [...] So then if it doesn't trigger [there,] then we [...] make the assumption that, [though] it's not triggering, it's also not flooding. So the traffic is unique enough — or at least the indicators are — to put in production. And that's when we put it in production"* (P3).

It is not to say, however, that "good" rules will remain that way indefinitely. Due to the constant evolution of the Internet and the types of traffic that you can find in it, there remains the chance that even the most specific of rules will start detecting a new type of traffic as potentially malicious. It is then up to the SOC to handle such situations quickly and effectively to prevent further incidents.

The number of false positives triggered by a rule is closely tied to the previous aspect. A rule that triggers many false positives will inevitably tie up SOC resources unnecessarily, as these cases will still need to be evaluated individually by an analyst. There is a trade-off here, though, that was mentioned by several participants from Org1, namely that the acceptable proportion of false positives a rule triggers depends on the severity of the threat that the rule is trying to detect: *"we as writing experts can say, well, this is truly malicious behavior and we are willing to risk, let's say, a 50% false positive rate so we can catch these as soon as possible, because the weight of it will balance the false positives"* (P5). P13 also mentions such a trade-off that is sometimes made in cases of new discoveries of severe threats. In such instances, the severity of a threat and the necessity of being able to detect it can sometimes trump potential standards for noise and false positives: *“Generally, these rules are a little more cowboy fire from the hip, because we’re trying to be fast, and we accept some additional noise”* (P13).

Org4 operates an NIPS that drops traffic when it triggers a rule, as opposed to an NIDS that raises an alert, and Org8 also did up until their recent switch to an NIDS. The case of Org8 gives us the opportunity to examine the changes in rule evaluation criteria caused by the shift in network monitoring approach. When operating an NIPS, P14, along with P10, who also operates an NIPS, describe preventing false positives as the most crucial aspect of rule development and management, since dropping legitimate traffic impacts business continuity directly. After switching to an NIDS, P14 states that they are still concerned about false positives, although for a different reason, namely the burden placed on the SOC and analyst fatigue.

Participants from Org1, Org4, Org6, and Org9 mention the danger of resource-intensive rules, which can potentially lead to sensor downtime. Thus, evaluating the resource intensity of a rule also plays a role in determining the quality of the rule, although not as significant a role as the previous aspects.

Only three participants (P5, P7, P14) mention false negatives as a main concern in regards to NIDSs. Given that the participating organizations are tasked with securing client networks, it is curious why failing to detect an intrusion is not considered more critical, especially since such SOC failures could be directly caused by an inadequate rule for which analysts and rule developers are responsible. P5 regards false negatives as most severe, while viewing false positives as “not that bad”, in contrast to most of the other participants. This could be due to P5’s role within the organization’s incident response team: false negatives can quickly lead to security incidents, meaning that P5 would be directly influenced by such occurrences. This is in contrast to the majority of the other participants, who are responsible for SOC operations instead of incident response. P14 views both false negatives and false positives as equally severe, although depending on the point of view of the assessor: from a risk point of view, false negatives are deemed most severe, while from the point of view of security analysis workload, false positives can be seen as most severe. Finally, P7 is an analyst with six months of experience in the field. Ahmad et al. state that junior analysts have “less developed mental models,” [5], which offers an explanation as to why their opinion does not match with the statements of their more senior peers. The issue of false negatives is also identified by Agyepong et al. as one of the challenges that analysts face, although to a significantly lesser degree [4].

4.4.3. RULESET EVALUATION

Through the interviews, we identified three primary drivers of ruleset quality, namely threat landscape coverage, ruleset volume and resource intensity, and management of legacy rules (rules designed to detect older threats, software, or systems).

Coverage. The topic of ruleset evaluation that featured the most is the aspect of coverage. Out of all participants, 15 agree that coverage is a primary factor of ruleset quality. The type of coverage different organizations strive for, however, does differ, and can be generally split up into two broad categories: "overall threat coverage" and "threat coverage for a specific environment". The majority of the organizations adhere to the latter category, while only Org1 and Org5 explicitly strive for larger overall threat coverage. An interesting case is Org3, where P9 states that, ideally, a client's entire environment threat landscape be covered, but clients of Org3 themselves find such coverage unnecessary and are satisfied with more moderate monitoring.

Legacy rules. There also seems to be some overlap between this philosophy on threat coverage and the way organizations deal with "legacy" rules. These are rules that, for instance, detect older threats, old versions of software, or rules that simply do not trigger any alerts anymore. This is not to say, however, that such rules are considered "bad". These are simply rules that are potentially no longer relevant for a specific detection environment.

Five of the seven organizations that aim for threat coverage of a specific environment remove these legacy rules from operation if the client network no longer expressly requires it (e.g., all instances of a certain software are updated to a newer version). Management of legacy rules brings with it different costs that the organization can incur, one of which is the traffic processing costs in case legacy rules remain in the ruleset indefinitely, which will be touched upon later in this section. On the other hand, removing legacy rules may incur higher costs to the organization. These rules would have to be individually and manually assessed before deciding on whether to keep or remove them. This means that the larger a ruleset grows, the larger the cost will be that an organization will incur when performing this manual work. This could explain why the aforementioned five organizations are able to easily remove superfluous legacy rules: the environment-specific rulesets that they create are smaller and, therefore, more manageable.

The rest of the organizations choose to keep these legacy rules in place if the threat is deemed severe enough, lest they miss a security incident in the future due to the removal of that rule. Org4 presents an interesting case here, where they balance between providing environment-specific threat coverage and coverage of more severe threats that might not be as relevant to a network anymore. In general, it also seems that the size of the ruleset employed by organizations is not a determining factor when it comes to ruleset quality.

Each organization operates with rulesets of different sizes, and their network monitoring services have been calibrated to work efficiently with the rulesets of that size; none have indicated desire for a different type of ruleset than they already use. The SOC teams in the organizations have calibrated their processes to the types of services that they offer. For instance, P8 says of Org3 that the analysts are capable of processing around 300 alerts per day, which is almost exactly what the SOC receives from the client networks using only a commercial ruleset for threat coverage. Adding more rules to increase coverage will likely not add any benefit for clients, and may in fact be detrimental to overall security, since the analysts will then become overburdened.

Volume and resource intensity. On a ruleset-wide level, resource intensity is primarily influenced by the number of rules contained within the ruleset. As was the case for the resource intensity of individual rules, a ruleset's resource intensity does not seem to be a major issue for most of the organizations. *"There's a correlation, obviously, between a hundred thousand rules vs. one thousand rules in terms of performance. Now, I can have a thousand poorly performing rules that perform worse than a hundred thousand good rules, so you can't say there's a causation there [...]. But if you start talking about engine specifics and how it's doing everything from the multi-pattern matching to the actual rule inspection, [...] all that makes a difference, so volume doesn't really factor into it as much,"* explains P10. However, Org6 and Org8 do explicitly acknowledge the point, the reason being that both organizations monitor networks where the data rate can reach 100 Gbps. At these high rates, the network sensors used by organizations can easily be overwhelmed if every packet has to be tested against a ruleset that is much too large. Other organizations do not monitor networks or network segments that generate that much traffic, or simply operate with a single ruleset that is maintained small enough that throughput degradation due to volume will never be an issue.

4.4.4. COLLABORATION

When asked about the way they worked together with immediate peers to achieve their workflow objectives, participants spoke mostly about two main elements: clients and peers. Client input/feedback was a theme that featured most often and consistently across all the interviews. Some respondents came up with informal examples of interactions, such as *"it's a good change that you've done"* (P16) and *"yeah, why didn't you put these rules into production?"* (P17), while the others described a detailed setup communication mechanism on case-by-case basis: *"For instance, we have an entire email inbox set up for end users [...] to email us directly and say, hey, there's something weird either going on my computer or something I received in an email. Do you mind checking it?"* (P11).

In terms of processes, client input is seen as necessary early on, during on-boarding (as reported in [169]) to familiarize with client systems, and also to fine-tune specific rules. Responding to needs extends to when clients are reacting to specific developments reported by news sources, e.g., *"you'll see the news reports, whatever hits the news, all clients come in [and say] 'oh, do you have coverage for SolarWinds?"'* (P10), and changes in the business environment, e.g., *"the government decides what we have to do"* (P14). Although there may be these inputs to the on-boarding and management of rules, generally engaging with clients constructively, it can nonetheless swing between imploring the client to *"trust us and let us do our thing. We'll show you the summaries of what we've done"* (P13), and having client representatives equally versed in the technology who might be *"a really up-to-date security officer, or maybe a network engineer who is curious if we cover [a particular threat] too"* (P12). Although Onwubiko et al. [169] also mention the importance of the client perspective, their work makes no specific mention of client input regarding the desired threat coverage.

The second element, collaboration with peers during rule design, development, and evaluation, is essential to safeguard the flexible, tailored approach that we have mentioned in the previous sections. *"We usually don't have an issue working together. To be honest, it's the opposite - it's actually working together has got us through situations [where] we had no idea what to do"* (P16). Respondents from different organizations describe having agile protocols in place to execute the "four eyes" principle[57] and to resolve issues within the team, as well as possibilities to solicit external expertise. Trust and freedom to both ask for clarification and to implement the solutions deemed necessary are commonplace. *"There's a mutual trust between us as coworkers that you know what you're doing; and if you didn't, then you should have asked"* (P3). Specialist expertise can be solicited through various means: workshops, chat groups, collaboration with colleagues from the other teams, industry-specific circles of trust, as well as through conducting own research. Time and capacity restrictions influence collaboration in different ways: prioritizing threats (*"Yes, so our first line does the initial filtering [...] So that goes from 300 alerts to five that go to my line and my colleague"* (P8)), and prioritizing analyst work-life balance (*"I want to say we will turn off the rule if it goes haywire, doesn't let us sleep at night"* (P17)). A notable improvement point is calling for even broader collaboration within the industry and security community: *"so if you have the second, third, thousandth opinion from a number of researchers and security professionals around the world working on different rules, that is very very useful. So I'm a big fan of these standardized languages for rules. And I think more people should use them in the industry"* (P16).

4.4.5. Improvement points

In general, we could not identify a common issue for the respondents. When specifically asked about the possible points for improvement, participants related mainly to different organizational aspects, in turn reflecting the diversity of their work processes. The improvement mentioned most often was better documentation of the different aspects of the rules (6 out of 17 interviews), such as actions that have been taken in the past or history of changes. Platforms or tools to systematize and access this information were mentioned as another desired improvement point: *“You don’t have, like, one dashboard that says: this is what’s happening right now, and these people should be cleaning up their rules”* (P6).

Other suggestions varied greatly and included better-organized client feedback loop, introducing an asynchronous aspect into peer review, better flow between deployment and testing, better quality of the rulesets, better and more testing, and more integrity checks. Automation was mentioned two times as an improvement component: automation of some “boring” processes, as well as extra quality assurance in an automated fashion to be able to employ more part-timers (i.e., a solution to a possible problem rather than a current one). This is mirrored in existing work [222]. As for the areas that could become better, one participant mentioned broader research to maintain detection quality, and another stressed the importance of keeping up with the current trends: *“[...] you start basing it on techniques and tactics and procedures that attackers use, [...], so it goes more from an artifact-based approach to behavior-based approach. And from that, I guess, you could move into even more, say, non-standard territory: statistical models and baselining; and you start going into data science and artificial intelligence stuff, right. So that’s probably what, I would say, is an improvement that needs to happen across the board”* (P16).

Perhaps indicative of the optimized nature of SOCs is the diversity in the desired improvement points within SOCs. Participants seem generally satisfied with internal processes; there is no single shortcoming that is severe enough for all participants to point out and that needs immediate fixing. As systems become more optimized to specific work processes, they also introduce fragility into the system, as any change will upset the delicate balance necessary for all workflow optimizations to function properly. This is exemplified by an occurrence at Org1, where the addition of a SIEM tool to their existing pipeline that was meant to lighten SOC workload actually overloaded analysts by tasking them with using a tool they were unfamiliar with.

4.5. Discussion

Many factors are considered in decisions around the use of rules in NIDS and NIPS systems. These are weighed against each other on a continuous basis in order to optimize the management of potential intrusion events. This phenomenon has been explored elsewhere [206] but only through construction of a conceptual SOC model representing analyst expertise and alert triage times, rather than direct empirical data gathered from practice (though the requirement to ‘trade off’ aspects against each other is highlighted). Many proposed solutions aim to optimize network monitoring and response workflows at a higher level (e.g., after the SIEM layer [8]), taking the inefficiencies of lower-level components (e.g., NIDS rulesets) as a given and immutable. Our results demonstrate that such inefficiencies are not immutable at all, as it is the proper balancing of the aforementioned factors that determines the quality of threat detection and SOC effectiveness. Through the balancing of these factors and trade-offs at a lower level, SOCs can significantly reduce the impact of such low-quality information on the higher levels.

4.5.1. Creating quality rules and rulesets

A recurring theme during our interviews with practitioners was the importance of using good rules, since that is what provides the SOC with security information and allows them to respond accordingly to threats. Creation, acquisition, and management of rules allow a SOC to both detect incidents and have the capacity to meaningfully respond to the incidents, supporting the *detecting-responding alignment* [212].

Kokulu et al. [120] investigate challenges surrounding the scalability of SOC operations, referring to *“complexity and the difficulty of integrating new technology”* into a SOC as the singular issue regarding scalability. Furthermore, while prior work has identified trade-offs made by SOCs during rule management, network monitoring, and incident response, they are limited to balancing false positives and false negatives [120], and trade-offs revolving around monetary costs of SOC analysts [206]. In this work we identify numerous additional aspects within SOC operations that are weighed and balanced against each other to effectively operate a SOC. Additionally, we also find that the structure and content of rules are determined not only by the rule design itself, but also by the externalities created by the characteristics and workflows of SOC teams. By examining these characteristics at the lower level of NIDS rules, we are adding to the understanding of how management practices and security practices influence each other.

In order to write good rules and create effective rulesets, a combination of experience and understanding is needed in order to find an appropriate balance between the multiple critical factors that influence rule and ruleset quality. The critical factors that were mentioned by participants are (1) specificity, (2) number of alerts and false positives, and (3) coverage. These separate factors are often interlinked, and so by altering one factor, another factor will be affected by that change, perhaps detrimentally. Choices about how to balance these factors are reflected in the security processes and approach of each organization. This indicates that balancing these factors is a non-trivial task that varies per environment. Such tasks thus require a significant degree of familiarity with the respective environment and its (human and computerized) components to make an appropriate decision. A particular configuration of these factors might work for one SOC, but possibly overwhelm another. This can also influence how new technologies are implemented into existing workflows, and their potential effectiveness within its respective environment. Slight differences in resource and workload capacity between SOCs can result in wildly distinct outcomes.

Specificity. Specificity in a rule interacts with all other rule management factors; increasing a rule's specificity to a threat makes it less prone to false positives, but also inherently decreases the threat coverage provided by the rule. In turn, this requires a larger volume of distinct rules to provide broad coverage. This is balanced against having more general rules, which, although they may increase the threat coverage to detect, e.g., variations of the same exploit or malware, it will also potentially cause the rule to produce more false positives.

The general consensus among our participants is that it is more desirable to have specific rules than generic rules, in order not to overwhelm the SOC – and the analyst team – with false positives. A study analyzing SOC and rule update logs found the same pattern: a general trend towards making rules more specific [239]. This may also explain why commercial rules are regarded as less than ideal by our interviewees, and why multiple organizations complement commercial rulesets with their own in-house ruleset, or choose not to use such rulesets at all.

Alerts and false positives. All participants remarked that minimizing alert floods and false positives is an objective to strive for. This ties into the previous point on specificity, which can reduce the risk of both. There are instances, however, when a larger amount of false positives is tolerated, namely when the severity of a threat is deemed high enough, or after the emergence of a new threat for which no coverage or understanding yet exists. Due to their time-sensitive nature, such pressing circumstances thus force detection to rely on underspecified rules that also trigger on benign events.

In the case of Org2, for instance, work processes are calibrated to instances where the used commercial ruleset works as intended (i.e., no floods or excessive false positives). During instances where it does not, the SOC disables the infringing rules, thereby reducing false positives and workload on the SOC, sacrificing threat coverage in the process.

Coverage. In Section 4.4 we explained how organizations deal with threat coverage in distinct ways, which can depend on the type of service that they provide to clients. One of the decisions an organization needs to make is whether to optimize for overall threat coverage, or to optimize threat coverage for a specific environment.

Among our participants, two particular themes emerged: firstly, retaining rules that rarely triggered but were seen as important to keep just in case, because the impacts associated with a true positive or incident absolutely had to be avoided. This relates to a concern of potential ‘Black Swan’-style risks [226] (low probability but high impact). This also relates to non-trivial decisions about the retention of legacy rules.

Second was a phenomenon similar to the ‘benign triggers’ observed by Alahmadi et al. [8], where in essence, a rule was triggered which analysts had already determined a response for, i.e., action had already been taken prior to the alert and there was seen to be no need to be told about it again. A simple example is network policy rules that are tolerated due to client-specific use cases. Persistent triggering of such a rule was then because the rule had not been refined, seemingly because it was not seen as worthwhile to do so, and the fact that the rule still works as intended and, therefore, still provides the threat coverage it was designed for. While tolerated, such benign triggers can potentially provide additional information about a potential security incident in the future.

INTERNAL AND EXTERNAL FEEDBACK LOOPS

The interviews tell us that there is no strictly-defined feedback loop for quality in terms of threat detection on an organizational level, and that rule management relies in part on the intuition of the analyst. Analysts rely on their intuition to make appropriate decisions. Since experienced analysts perform better than more novice peers [206], this indicates that a feedback loop exists for the fine-tuning of this intuition. Decisions regarding network incident monitoring and response are very context-dependent, and analysts state that the process for developing this intuition consists of investigating and analyzing alerts and security incidents within the contexts of different networks and organizations. This iterative fine-tuning of analyst intuition is critical for the proper functioning of a SOC, since determining the value and validity of a rule is not a straightforward task.

In many cases, the only indications of rule quality are the number of alerts a rule generates (i.e., whether it floods the SOC with alerts), and the number of false positives it produces. This must be considered alongside the analysts’ understanding of the threat(s) related to the events, and to what extent it is believed that those threats relate to the organization; this is reflected in analysts’ proactive searching for emerging threats, which are then considered against the knowledge of the organization/client network and vulnerable systems.

All organizations that create their own rules test these rules on real network traffic before pushing them to the production environment in their clients’ networks. Testing yields an indication of rule quality. Only when true positive detections arrive at the SOC and analysts compare that to the number of false positives, can they truly determine whether the rule is of high quality. The proactive 'horizon scanning' for new threats is an effort to compensate for this, i.e., to avoid a false negative – a successful attack – being the moment when a threat is discovered. Among our participants, there was a perception of being successful at catching all possible threats, which indicates that they are working effectively and protecting their organizations/clients. This does mean, however, that there rarely is a feedback loop of false negatives (breaches) back into the detection process, and in turn, a lack of experience with false negatives. Detection capabilities then evolve based on a mixture of incidents 'elsewhere' that are communicated out in the professional community or via news, as well as the aforementioned 'horizon scanning' for new threats, and problem-solving as to how those threats may relate to managed environments. The feedback loop consists of targeted rule edits and additions to provide coverage for the threat(s) that contributed to the breach. Since there is no way for the SOC to know when an incident will be missed, this is the only action SOCs can take for rule management.

Prior work [234] has studied this 'horizon scanning' and the use of publicly available data in the creation of rules. While this falls out of scope for this work, future work could further examine how reliance on this data affects SOC workflows and eventual network security, especially since analysts have stated that such open-source threat research is integral to the creation of in-house rules.

EXTERNAL RULESETS

Our participants regarded commercial rulesets as akin to 'starter packs'. Whether these rulesets are useful for the organization or not depends on the manner in which they provide their security services. Akin to the concerns about rare, high-impact events, external rulesets were regarded as useful for standard threat landscape coverage, containing the low-hanging fruit of threats that smaller teams of rule developers and analysts cannot create themselves due to time and resource constraints. Additionally, since they were unanimously regarded as "noisy," organizations need to have the technology and resources to be able to manage all the noise that they create. There is then a negative externality created by these external rulesets – their intention is to provide broad and comprehensive coverage, as a signal to customers of the quality of their rulesets as a product. However, this does not consider the amount of 'noise' they generate for those using the rulesets, who must then choose whether to selectively switch off or abandon the rules, as the cost of fine-tuning each rule to be less noisy was regarded as simply too great by our participants.

Intrusion detection vs prevention

Through the interviews, we also discovered that rule quality assessment differs depending on the type of system that is being used. Three of the nine organizations considered in this study operate an NIPS, or have done so in the recent past, and all state false positives as the greatest concern in this context, since traffic is immediately dropped. Consequently, the balance between making a rule specific or generic tilts more towards making rules specific, in order to reduce the risk of false positives. The fact that false positives remain a major concern is a strong indicator of why most organizations opt for an NIDS. Client business continuity is a top priority for all organizations, which is why even Org5 uses a hybrid NIDS/NIPS where most rules drop traffic, while some rules that are prone to false positives are modified to only produce alerts when triggered, instead of dropping the infringing packet.

The case of NIPSs dropping packets directly after triggering a rule highlights the importance of start-of-pipe improvements (see Figure 4.2). Taking such drastic action beforehand, and attempting to correct the issue in the SOC afterwards is effective for neither the SOC nor the client, since the consequences of such misconfigurations have already carried out their effect on the client network, potentially, business continuity.

4.5.2. Wider implications

From the interviews, we can ascertain that all organizations seem to be satisfied with how their network monitoring processes are set up. No participant suggests that additional resources are necessary to improve internal processes.

Where Kokulu et al. [120] mention that SOCs have insufficient budget to accommodate, e.g., one-off costs such as travel, here we find our participants in general agreement that they have sufficient budget to operate their SOC and, crucially, to meet the needs of their clients. This, however, may be a consequence of action to limit ‘floods’ of false positives or ‘noise’ created by general (external) rules, so that alerts are manageable within the workload of the analysts. It is possible that budget issues are not reported because SOCs are matching their workload to their work capacity.

Current literature claims that signature-based NIDSs are becoming more antiquated by the day [39, 240], and should therefore be replaced by ML-based systems. However, ML-based are often positioned as being a solution much later in the process of protecting a network; here we see that the rules used to first detect events must be mastered, and that ML-based systems cannot necessarily build a picture of the system in retrospect so late in the process. ML-based systems are not unaffected by the problem of false positives [8].

It is also often claimed that signature-based NIDSs cannot keep up with the fast evolution of the threat landscape, are unsuitable for the detection of new threats, and can therefore create security risks for the organization that employs such systems. Not a single participant mentions this as a potential drawback of their signature-based systems. Even Org7, which is actively and explicitly investing and focusing on machine learning-based approach, nevertheless finds most clients subscribing to the signature-based service, and tends to use their ML approach in environments that are more standardized, such as point of sale systems.

However, ML was seen as having a potential role in the protection of systems, and it holds the promise of reducing the alert volume to be investigated. Moreover, explainability of alerts is mentioned by Alahmadi et al. as crucial for the efficient functioning of a SOC, and current implementation of AI and ML models largely remain opaque [8], meaning that issues regarding explainability will only get amplified in case of the implementation of these systems.

4.5.3. Recommendations

From this study we have identified a preliminary set of recommendations:

Rule specificity and reduction of false positives. Consistent with findings from Agyepong et al. [4], yet contrary to the work by Kokulu et al. [120], we find that false positives are a major concern for SOC teams, primarily due to workload issues. SOCs often take false positives as an indication that a rule needs to be made more specific to the threat it is trying to detect. Therefore, a straightforward recommendation is to make rules more targeted and specific. Ideally, though, this adjustment should be assessed on an individual basis. From our interviews, we learned that false positives are often tolerated to some extent if the threat being detected is novel or severe enough. This indicates that driving down false positives, while an important aspect of SOC management, is not a cure-all. As stated in our Results, making a rule more specific will also reduce threat coverage, and thereby also losing information about any potential malicious traffic inside a network. And as explained by P6, generic rules are more useful in cases where there is a preponderance of similar malware that can be easily covered by a single rule.

Still, reducing the amount of false positives a rule generates will make the rule more reliable, and subsequent alerts generated by that rule more trustworthy and actionable, thereby also reducing the potential workload for SOC analysts. Proposed solutions to current workload issues include ML systems that are designed to provide analysts with actionable tasks by filtering out false positive noise from already inherently noisy alerts [8]. Even though addressing the issues of false positives at the end-of-pipe stage may improve workload difficulties for a SOC, this is treating the symptom and not the cause. Furthermore, attempting to improve workflows by adding additional technologies to the network monitoring pipeline may very well upset the delicate balance within a SOC, since our participants indicated that they were generally satisfied with their workflows, and comfortable with using the tools that they are currently familiar with.

Feedback loops. Just as it is labor-intensive to create a large collection of specific rules to catch all variations in a malware family, so, too, is it to build every single exception into rules to filter out every potential false positive. This is in contrast to our finding regarding making rules more specific and is an important counter to an unquestioning implementation of such a policy: specificity is a goal to strive for, but it is too effortful to write specific exceptions for every corner case. This all depends on the rule, the threat, and the situation itself, meaning that there is no magic bullet that will work for every case. What is clear is that if a SOC wants to increase the quality of the rules, one way to accomplish that goal is to have much more feedback from the detection process in the rule development process. In practice, this feedback is very limited, often only occurring in cases of (false positive) floods. Including more feedback from additional steps in the rule evaluation processes, such as a rule's testing phase, or when rules stop producing alerts, can yield analysts with valuable information regarding rule quality. Additionally, proactively implementing discoveries within the wider security community is a process that needs to be exploited further.

Creation of quality rules mainly relies on analyst intuition, which, in turn, can be developed by means of these feedback loops. Through the inclusion of novice analysts to a greater extent within these feedback loops, organizations can help cultivate such necessary level of experience.

Document intuition and tacit knowledge. Finally, another factor closely related to aspects of internal collaboration is the aspect of tacit knowledge, something also identified in work by Agyepong et al. [4], where they recommend the creation of playbooks and distribution of documentation about SOC processes that less experienced analysts can draw knowledge from. In addition to this, we recommend the setting up of well-defined systems of collaboration within teams using rulesets (whether they are within a SOC or elsewhere), such as developing rules in a pair programming fashion, or under the four eyes principle [57], and peer review sessions of developed rules. This will allow for a more fluent exchange of tacit knowledge from more experienced analysts to the more junior ones, leading to more effective use of rulesets.

4.6. CONCLUSION

In this paper we aimed to shed light upon the processes that surround the development and acquisition of rules for network detection. We found that there are a number of critical factors, such as rule specificity and total number of alerts and false positives, that dictate the manner in which rules are managed, and that there was significant consensus regarding the importance of these factors when they are used to determine the quality of rules and rulesets. These factors are weighed against each other in different ways by different organizations and carefully calibrated to the organization's network monitoring practices. Previous work has aimed at improving SOC effectiveness through different means, many of which fail to take the aforementioned factors into account, instead opting for solutions that make SOC data easier to deal with by automatically filtering out noise. We argue that such solutions are sub-optimal, and we presented a number of concrete recommendations that address these factors at the earlier stages of the network monitoring and incident response pipeline. With this, we propose a path forward that aims not to treat the symptoms, but to address a root cause of potential SOC ineffectiveness while leveraging a SOC's current resources and technologies.

5

Security incident prediction

In the fast-paced world of cybersecurity, organizations are continuously striving to enhance their security posture and reduce the risk of successful cyberattacks. These often rely on compliance requirements and normative frameworks to prioritize security measures. However, the effectiveness of these efforts remains uncertain, as measuring security improvements is complex and challenging. Recently, novel approaches have emerged that directly measure an organization's security using external network signals. Commercial companies offer risk rating services based on externally observable data. These approaches treat security as a latent variable inferred from noisy external signals. This paper advances empirical risk estimation by comparing external signals to the internal state of an organization's network. Partnering with a Managed Security Service Provider (MSSP), we acquired data on security events from their clients' networks, labeled by Security Operations Center (SOC) analysts from October 2016 to January 2021. For the same period, we collected 338 external signals related to these networks. Employing a gradient booster classifier, we successfully predicted the number of high-risk security incidents with high accuracy ($R^2 = 0.81$). Our results confirm the predictive power of external signals for an organization's internal security state. We found that certain external features significantly influence incident predictions. However, sparse features, such as organization size, can lead to misleading outcomes. Nonetheless, by better understanding these interactions, organizations can take proactive steps in prioritizing which security controls to deploy.

5.1. INTRODUCTION

An organization can take many steps to reduce the risk and harm of a successful attack. Chief Information Security Officers (CISOs) often prioritize measures based on compliance requirements and normative frameworks [149] – like NIST [157], COBIT [108], CIS Controls (formerly SANS Critical Security Controls) [44]. While the objective is clearly to improve the security posture of the organization, it is unclear to what extent these efforts ultimately have the intended effect [149]. This is in part because measuring security and, by extension, security improvements is not straightforward.

What is typically measured by organizations is not security but effort: what controls they have implemented and how these compare against "best practices" or some normative framework like a maturity model. Those models assume that more advanced controls result in higher security. This assumption is not unreasonable, but ideally, we would like to measure the result, not the effort.

In the past decade, new approaches have emerged that try to measure the security of organizations more directly. Services like SecurityScorecard, FICO Cyber Risk Score, and BitSight offer risk rating to organizations directly and to third parties who value accurate information on security postures, e.g., for third-party vendor management and cyber breach insurance underwriting. These risk-rating services are based on methodologies that have been validated in peer-reviewed publications. A leading example is the work of Liu et al. [133]. The authors do not rely on any self-reported data from the organizations they rate. Instead, they collect externally visible data on misconfigurations – such as misconfigured DNS, SMTP, and untrusted TLS certificates – and on malicious activity in the organization's network – such as the presence of hosts in different abuse blacklists. A classifier trained on these features is then able to predict with some accuracy which organization will suffer a breach, as observed in the VERIS Community Database [238] and other sources. Similar studies were presented by Edwards et al. [73] and Kotzias et al. [121].

These approaches, both in their academic and industry implementations, currently provide us with the best available methods for empirical security risk estimation. In measurement terms, they treat security as a latent variable that cannot be directly observed, but that can be inferred from a set of noisy externally visible signals–somewhat similar to how IQ is not directly observable but is inferred from the answers to a range of questions and puzzles.

While there the predictive power of these methods provides some evidence for their validity, they do treat the organization as a black box. Do these signals really tell us the internal state of the network? Also: the models can classify organizations that suffered a large, public breach versus those that did not—a coarse distinction. Do they also provide a valid assessment of the security posture of organizations that did not suffer such a breach?

We want to advance the state of empirical risk estimation by comparing the external signals to the internal state of the network. To this end, partnered with an MSSP with data on security events in the networks of their 207 clients, as labeled by analysts in their Security Operations Center from 2016 to 2021. This provided us with reliable signals about the internal state. For a subset of 101 clients, we collect 338 external signals, similar to those used by [73, 133]. We then develop a model to predict the security events from the 338 external signals.

We find that the external features as described by Liu et al. [133] and Edwards et al. [73] do have predictive power for the internal state of an enterprise. However, the classification technique from [133] does not work, due to the fewer number of organizations in our dataset (their 3 million organizations vs. our 101) and the larger number of finer-grained security incidents of differing severity that we have access to (their 1,150 incidents vs. our 12,218 incidents). Instead, we use XGBoost [47], a gradient booster classifier. Our model is able to accurately low-risk and high-risk incidents ($R^2 = 0.65$ and $R^2 = 0.81$, respectively), although we find that prediction performance is highly dependent upon sparsity of the external signal data.

With this work, we advance the state of the art of empirical risk estimation. We confirm the assumption behind prior work that there is correspondence between external security signals and internal security events. These results open the door to numerous different use cases. Collecting these signals can help organizations to take a more proactive approach to their security by identifying worrisome security signals beforehand. MSSPs can use the signals and model to provide predictive risk metrics to clients.

In sum, we make the following contributions:

- We present the first study that explores the links between external network security signals and internal network security signals as SOCs or network operators see them.
- We replicate the work by Liu et al. [133] and, to an extent, Edwards et al. [73] in the real-world environment of MSSP client's networks.
- We find that the methods from Liu et al. [133], when applied to the finer-grained datasets of SOC security incidents, do not produce results equal in effectiveness as reported in the original work. When adjusting the original methods to our unique datasets, however, we are able to create and train a regression model that is able to predict the number of future incidents with high accuracy.
- We demonstrate that feature sparsity plays a significant part in model performance, thus limiting the applicability of incident prediction systems to organizations possessing

a sufficient level of Internet presence and exposure, especially for the prediction of more severe incidents. Thorough evaluation of NIDS alerts and incidents by SOCs is, therefore, crucial to ensuring the design of a well-functioning model by means of adequate ground-truth data creation.

5.2. Related work

The related work in this domain encompasses a range of machine learning-based approaches applied to different aspects of cybersecurity risk estimation. Soska et al. employ various machine learning techniques to estimate the probability of a website becoming malicious in the future [216]. They gather historical data from websites known to be benign as well as those known to be malicious. This data is used to ascertain the point in time when a website turned malicious. Features such as Alexa Site Rank and HTML DOM information are extracted and utilized to construct a classifier capable of predicting whether a website will be compromised within a year, achieving a true positive rate of 66% and a false positive rate of 17%.

The work by Liu et al. [133] is the closest to what we intend to study in this work. Liu et al. develop a model that predicts an organization's breaches using only externally observable symptoms of mismanagement and open incident data that indicates malicious activity originating from an organization's network. They found this data to be predictive of breaches, achieving an average true positive rate of 90% and an average false positive rate of 10%. Expanding on this work, Sarabi et al. attempt to forecast a broader range of incidents, encompassing both physical and cyber-attacks [197]. One of their objectives is to predict the type of incident an organization will confront in the future. They predict future incidents with a 90% true positive rate and 11% false positive rate and identify a correlation between organization features and incident signatures, but are unable to predict the specific type of incident with certainty.

Edwards et al. carry out a statistical analysis to identify correlations between what they term "organizational risk vectors" and botnet infections [73]. The authors gather data through external measurements of an organization's IP space, assessments of services that may be vulnerable, and externally observable security incidents. These potentially vulnerable services are referred to as risk vectors. Examples of such risk vectors are publicly accessible unsecured protocols like FTP, Telnet, and SMTP, improperly configured TLS services, and the existence of P2P file sharing on the network. Consequently, the externally observable incidents are botnet infections. Although they successfully observe a correlation between the chosen risk vectors and malicious activity, their research is limited to botnet infections only.

The work by Kotzias et al. also investigates enterprise security posture but differs from the previous papers in that the majority of security posture analysis is performed by means of internal data in its analysis: presence of malware, potentially unwanted programs [121]. They also utilize IPv4 scans of organizations to measure vulnerability patching behavior. Their findings revealed that many organizations struggled with patch management and timely vulnerability remediation. Additionally, they find that industries are affected to a differing degree, with the electrical equipment industry being affected much more than, e.g., banks.

Xiao et al. employ data from several spam blacklists, vulnerability exploit information, and end-host patching behavior to assess the extent to which vulnerabilities are exploited in the wild [249]. In particular, they identify network entities exhibiting similar behavior concerning malicious activities and compare it to the risk behavior of various end-hosts by analyzing patching data. Through this analysis, they can detect the existence of exploits in these networks with a true positive rate of 90% and a false positive rate of 10%.

These diverse studies collectively highlight the potential of machine learning techniques in predicting cybersecurity incidents and understanding security posture. Our work seeks to contribute to this field by thoroughly examining the correspondence between external network security signals and internal network security events, thereby advancing the understanding of empirical risk estimation. Importantly, our study stands out as the first of its kind to leverage internal data captured by NIDS and integrate it with external signals to gain deeper insights into an organization's security landscape. By bridging the gap between external and internal security signals, we aim to provide a more comprehensive and accurate estimation of cybersecurity risk, paving the way for more effective and proactive security measures.

5.3. Data sources

5.3.1. Datasets

Internal alert and incident data. Through our partnership with an MSSP, we access alerts from NIDSs installed across client networks. The MSSP's SOC constantly reviews these alerts, escalating significant ones to incident status for further investigation. Incidents are categorized by experts at the MSSP into several labels based on accuracy and severity: *false positive*, *low risk*, *high risk*, and *successful hack attack*. Labeling is performed by experts at the MSSP, who rely on their extensive personal experience. Thus, while labeling remains potentially ambiguous at incident-level, we assume the labels to be more reliable when taken as a whole (as demonstrated by the MSSP's continued existence).

We have selected a subset of this data from October 2016 to January 2021, covering 101 different organizations. This subset includes only the true positive incidents—comprising low risk, high risk, and successful hack attack categories—totaling $12,218$ incidents. These incidents are distributed as follows: $6,825$ low-risk, $5,039$ high-risk, and 354 successful hack attacks, representing 101 of the original 207 organizations in the dataset.

IP-to-organization mapping. To attribute IP blocks to their corresponding organizations, we use the MaxMind GeoIP database [144]. Through this database, MaxMind periodically publishes mappings of IP addresses to organization names. We use MaxMind GeoIP data from October 2016 to January 2021.

Historical IP registration data. The RIPE NCC offers a historical WHOIS API [191] for retrieving IP block allocation data. We use this API alongside the specified organization-to-API mapping to confirm the accuracy of IP block assignments to their claimed organizations at specific times.

IPv4 scans. To detect misconfigurations, vulnerabilities, and other management issues within organizations, we utilize scan data from Shodan [208]. Shodan conducts regular Internet-wide scans of IPv4 addresses across various ports, identifying active services, retrieving certificates, capturing banners, and detecting potential vulnerabilities. However, there is a limitation: Shodan's daily port scans are not consistent; it randomly selects different ports for each scan. This method expands the diversity of detected ports and services but results in less reliable feature measurements, as the availability of a service on a specific host during certain time periods cannot be precisely determined.

We also use data published by ShadowServer [229], an organization that performs daily IPv4 scans and publishes data concerning accessible services and vulnerabilities. See [228] for the full list.

Malicious activity. As signals of malicious activity occurring in the networks of organizations, we identify compromised hosts within the IP blocks allocated to the MSSP's client organizations using a number of commercial IP reputation blocklists. These include APEWS [15], CBL [2], Spamhaus XBL [230], PhishTank [56], PSBL [188], and ShadowServer [229]. These blocklists capture malicious behavior originating from hosts such as spam, phishing, and botnet infections. While the availability of the datasets differs in time span, they all overlap from 2016 to 2021. Thus, to ensure completeness of data, we extract all data only from this span of roughly five years.

BGP routing data. The RouteViews project collects BGP routing information from numerous routers worldwide [193]. This data contains detailed records of how routes change over time in the form of BGP update data. We collect this data from all available RouteViews collectors for the time span of October 2016 to January 2021.

5.3.2. ETHICAL CONSIDERATIONS

Internal incident data is sensitive information. Several actions were taken to ensure customer privacy. Firstly, all data was stored and analyzed on MSSP premises. The data analysis took place within the terms of contract of the MSSP and its clients. Finally, all data was checked by MSSP staff for personally identifiable information before being made available for analysis.

5.4. DATA PROCESSING METHODOLOGY

Measuring external network signals for our set of organizations first requires the identification of their IP space over time. A simplified flowchart is illustrated in Figure 1 (Appendix B.1). Only after identification can we process the different datasets to extract the necessary features. This section will elaborate on these processes.

5.4.1. ORGANIZATION AND IP BLOCK MAPPING

Although we obtained a list of organizations from our partner MSSP, these organization names are mostly shortened versions of the official organization names. Sometimes organization names are used by other entities, so to properly identify the organizations and determine the official names of the organizations, we first perform a manual web search for each organization in the customer list and verify with the MSSP which entity is the one receiving security monitoring and response services.

To gather and analyze the degree to which the MSSP's clients are exposed to threats, we need to identify the IP address blocks that belong to each organization. While the MaxMind GeoIP data provides an IP-to-organization mapping, we require an organization-to-IP mapping. Thus, at every available date, we query this database for all IPv4 addresses, enabling us to reverse the mapping and create a longitudinal mapping of organization names to IP blocks. Depending on the organization, this lookup will also yield IP blocks officially allocated to, e.g., cloud or hosting providers. Since this is still infrastructure under partial management of the corresponding company, we include it in the selection.

These GeoIP records are not standardized, however, and there can be considerable differences in fields between different records — even those belonging to the same entity. For instance, different records belonging to the same organization can contain name fields that differ in punctuation and capitalization. We then query this IP registration data for all organization names to cover all variations and obtain a list of all organization names that contain the queried company name with the respective IP blocks registered to that organization at every date. To ensure that the collected names are correct and contain only the names belonging to the MSSP customers, we perform manual verification of all the names in the list. Of the initial 207 organizations, we only find 157 organizations in the data. We proceed with the organizations for which we are able to find IP registration data and discard the rest.

We then query this same data, but extract all the different IP blocks that are assigned to each organization (and all valid variations of organization names). We collect these IP blocks daily — as long as the data availability allows it — for the years 2009 to 2021. Gaps in the daily sampling are filled by propagating the last valid observation forward. Afterwards, we sample the daily IP block data at two-week intervals. IP block allocations do not change very often, so limiting the queries to these larger intervals not only saves execution time and manual verification, but also mitigates instability due to potentially inconsistent allocation data.

Next, we split the list of companies into two groups: one containing companies that have 256 or fewer IP addresses assigned to it, and another containing companies with more than 256 IP addresses. The former group is used for further manual processing, which is now feasible due to the smaller number of IP addresses per organization that potentially need such manual verification. We perform this manual verification to examine the stability of the allocated IP blocks and the potential effects of IP block sub-allocation on the validity of our data. These steps are explained below.

We compare the IP blocks of each organization to the official RIPE allocation list. If the list contains the IP block allocated to the same client organization, we classify it as correct. Otherwise, we proceed to the next step.

If the RIPE record matches the organization name found in the MaxMind data used for the query, then the name is classified as correct. If the RIPE record mentions any other organization that is not a network operator or hosting provider, then it is classified as incorrect and discarded. In case that the name is a network operator or hosting provider, we assume the IP block has been delegated or sub-allocated and classify the name as correct.

If the RIPE historical WHOIS database has no record of the specific IP block at a particular date, additional steps are taken for verification. Specifically, the RIPE historical WHOIS database is queried for the parent IP block of the potential IP block. If this parent IP block is registered with a network operator or hosting provider, we assume delegation or sub-allocation and classify the IP block as correct. Alternatively, if the queried IP block and date predates the official RIPE historical WHOIS records for that IP block, the first record is used for manual verification, following the steps laid out above. Any IP block that does not pass these verification steps is discarded. This process yields a list of all organization names and the IP blocks that have been assigned to it over time.

Through this process, we find that only 8 IP block errors (totaling 72 IP addresses over time) out of a total of 8,914 IP block-date pairs were verified. We consider this error negligible, and therefore abstain from performing this manual verification on the other subset of the organizations. After this verification, we merge both subsets.

We then smooth the two-week interval data by forward filling the dates over all the days in each two-week interval. Finally, we extract the subset of data starting in October 2016 to January 2021, corresponding to the availability of malicious datasets to ensure that the appropriate features can be extracted. Through this process, we are left with a dataset containing *101 organizations* and their allocated IP blocks over time.

5.4.2. Feature extraction

After collecting each organization's historical IP block registration data, we use the IP block and date pairs to query Shodan [208] for all its scan data for every IP block at every corresponding date. This yields a dataset containing all available port scan data for each IP address, belonging to the organizations in our population. It is mainly from this dataset that we extract the network features that, traditionally, are collected through port scans.

One of the main methodologies we adhere to for feature extraction is the one developed by Liu et al. [133]. The authors aggregate features over a window of 60 days to capture the dynamic behavior of an organization over time. To closely follow and verify their methodology and results, we select the same 60-day window as a base for our own approach. We also experiment with a shorter 30-day time window, resulting in decreased model performance. We will discuss this in section 5.6. Though the number seems arbitrary, these experiments illustrate the value of an organization's behavioral patterns over longer periods.

Additionally, they select the feature windows to be 60 days before the occurrence of a breach. This is unsuitable for our data, since our set of organizations is much smaller, but the incident population is much larger. Hence, we use a regular sliding window of 60 days, where the target values to be predicted are the number of incidents that will occur in the 60 days after the feature window.

Despite evidence linking an organization's industrial sector to cyber incident risk [26, 121], we exclude this as a feature due to our small sample size, where many sectors have only one or a few organizations. This limitation prevents reliable generalization within sectors and could skew the model by capturing excessive variance regardless of other features.

The full list of features can be found in Appendix B.3.

Malicious activity

We group the blocklists we have access to into three groups: spam (APEWS, PSBL), phishing (PhishTank), and miscellaneous malicious activity (XBL, CBL). For each organization, we extract the number of unique IPs that are present in each group, for every day in the time window.

PERSISTENCE

In addition to measuring the number of IP addresses that are present in malicious activity datasets, we also measure how long IP addresses remain (i.e., *persist)* in the malicious activity datasets throughout the given time window. Thus, from the previous malicious activity time series, we extract *persistence* features that describe the general behavior of the hosts in the three different malicious activity dataset groups.

Plotting the malicious activity time series yields a graph such as the one illustrated in Figure 5.1. As illustrated, the plot can be divided by two boundaries, splitting the area into three different regions. The two boundaries are calculated by taking ±20% of the mean of the time series. The middle region represents the "normal" behavior for malicious activity during the given time series. The behavior changes as the malicious activity moves away from the mean. Accordingly, the top and bottom regions represent the "bad" and "good" regions. For each region, we calculate the following statistics: 1) normalized average magnitude, 2) un-normalized average magnitude, 3) average duration of persistence in that region, and 4) the frequency at which the time series data enters that particular region. Finally, in addition to collecting this data for the entire span of the window, we compute the same statistics for the last 14 days of the window. The specifics are described fully in [133].

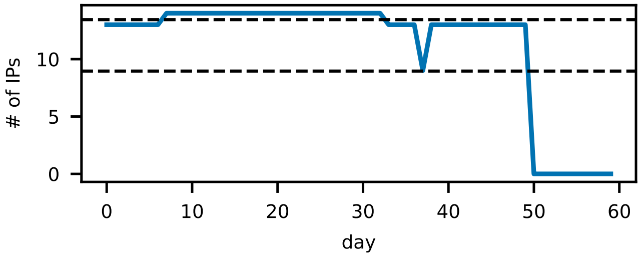


Figure 5.1: The number of an organization's IP addresses present in the spam datasets over a period of 60 days. The graph is divided into three regions by the dashed lines: the "bad" upper region, the "normal" middle region, and the "good" lower region.

MISMANAGEMENT SYMPTOMS

Liu et al. [133] specify five different types of mismanagement symptoms: 1) open recursive resolvers, 2) non-use of DNS source port randomization, 3) BGP misconfigurations, 4) untrusted HTTPS certificates, and 5) open SMTP mail relays. Unfortunately, the datasets available to us did not allow for the collection of DNS source port randomization and open SMTP mail relay features. A notable difference between our methodologies is that, while Liu et al. use a single data snapshot from which a single feature is extracted for each of the five mismanagement symptoms, we perform longitudinal measurements of each symptom. This will provide better information about an organization's security posture, as we are able to capture their evolution over time.

To collect the BGP misconfiguration data, we implemented the methodology employed by the collectors of the data employed by Liu et al. [133]. This methodology developed by Mahajan et al. [141] involves the detection of short-lived BGP announcements in the global routing table. It bases the detection on the heuristic that any BGP announcement that is withdrawn within 24 hours of its announcement is a mistake, and therefore counts as a misconfiguration. We collect historical data from all available RouteViews listeners [193], and processed the data to find such BGP misconfigurations. Unfortunately, after processing a month's worth of RouteViews data, we were not able to detect any misconfiguration. Even in 2014, it was noted by Zhang et al. that the methodology by Mahajan et al. [141] is dated, since current routing practices are much more "fine-grained" [254]. It is likely due to this limitation that this methodology was unable to detect a single misconfiguration for our limited organization population.

As for the detection of open recursive resolvers, the Shodan scan data allows us to extract the presence of such resolvers for every IP address. Thus, for every organization, we collect the total count, mean, and maximum number of open recursive resolvers that are present in their allocated IP block(s) over the duration of the sliding window, as well as for the last 14 days of that sliding window (in a similar manner as the persistence features).

In contrast to Liu et al. [133], we focus on TLS certificates in general, and not solely on certificates used for HTTPS. Furthermore, we look for errors and misconfigurations beyond verifying whether certificates are browser-trusted. As described by Edwards et al. [73], we test TLS services for the following configuration errors:

- TLS version less than or equal to SSLv3 [1],
- presence of Heartbleed or FREAK vulnerabilities,
- Diffie-Helman keys with less than 2048 bits or with commonly-used prime numbers [160];

and the following TLS certificate errors:

[1] Being published in 2019, the work by Edwards et al. [73] considers SSLv3 and older insecure. As of 2021, however, anything below TLS 1.2 is deprecated and considered obsolete [151]. Since most of our historical data originates from before 2021, we opted to leave this feature as is.

- certificates that are self-signed, expired, issued in the future, have non-standard roots, or have a broken chain of trust,
- certificates with RSA or DSA keys with 1024 bits or less, or ECC using less than 224 bits.

For every port scanned by Shodan over time, if that port uses SSL/TLS, we collect features according to the aforementioned categories. For every organization, we calculate the total count, mean, and maximum value of the number of IP addresses that host TLS services with configuration errors and TLS services with certificate errors over the duration of the sliding window, and the last 14 days of the sliding window.

LIVE HOSTS

For each organization, we aggregate daily data within the specified window to calculate the total number of unique IP addresses with open ports, the total number of open ports, and the total number of TLS-enabled services.

FREQUENTLY-USED SERVICES

Security posture information may also be inferred from the type of services that are made publicly accessible by organizations. We look for the presence of 21 different frequently used services and count the number of ports, categorized as “risky”, “neutral”, and “reasonable” services, as described in the work by Edwards et al. [73]. See Table 5.1 for the list of services.

Table 5.1: The list of 21 services and corresponding category, as defined by Edwards et al. [73]

Service	Category	Service	Category	Service	Category
FTP	Risky	SNMP	Risky	HTTP	Neutral
Telnet	Risky	SMB	Risky	NTP	Neutral
SMTP	Risky	MySQL	Risky	IMAPS	Reasonable
POP3	Risky	MSSQL	Risky	HTTPS	Reasonable
SunRPC	Risky	POSTGRES	Risky	POP3S	Reasonable
NetBIOS	Risky	RDP	Risky	SMTPS	Reasonable
IMAP	Risky	DNS	Neutral	SSH	Reasonable

For each organization, we count the number of ports per day that host any of the services in each of the three groups. Then, for each group, we sum together the counts per day. This yields three features: the total number of “risky”, “neutral”, and “reasonable” services discovered throughout the duration of the window.

SHADOWSERVER FEATURES

A limitation of the ShadowServer data is that it only contains scans of IP ranges that ShadowServer has geolocated to the Netherlands. Although many of the organizations are based in the Netherlands (and therefore primarily use RIPE-allocated IP blocks), the organization-to-IP mapping process assigned many non-RIPE IP blocks to these European organizations. The data will then be biased towards the organizations with a higher ratio of RIPE-allocated IP blocks. To regularize, we perform a (crude) regularization of the ShadowServer features: for each organization, divide all ShadowServer features by the proportion of total RIPE-allocated IP addresses to the total number of addresses.

SIZE

The *size* feature represents the total size of all IP blocks allocated to an organization at a specific time, which can change over time. For each window, we calculate the mean total number of IP addresses allocated to an organization. Liu et al. noted that this feature might capture "to some extent" the likelihood of targeted attacks [133]. However, in our experiments, the model tended to overemphasize the size of an organization, especially when dealing with sparse feature vectors (see Section 5.5.2), detrimentally affecting prediction accuracy. In the absence of workable features, the model disproportionately focused on the size feature, the most prominent non-zero feature, resulting in misleading predictions that did not accurately represent an organization's security state. Consequently, we **do not** include the size feature in the feature set.

5.5. DATA EXPLORATION AND MODEL TRAINING

After collecting the necessary data and constructing all the features, we explore the feature data and internal security incident data. This exploration enables us to identify the appropriate training and testing procedures, as well as select the most suitable model to use.

5.5.1. ORGANIZATION POPULATION

Organization network sizes vary widely, from four to nearly 8 million IP addresses, with an average size of about 3,600. Despite the expectation that larger networks would have more incidents, the data show only a weak correlation, especially for high-risk incidents and successful hacks (0.1).

The study includes 101 organizations across 20 industrial sectors, with technology, finance, and business services being the most represented. Despite the wide range of industrial sectors, we lack sufficient data for sector-specific analysis, hence our focus on the organization level. Thus, to improve the generalizability of our findings, we exclude the

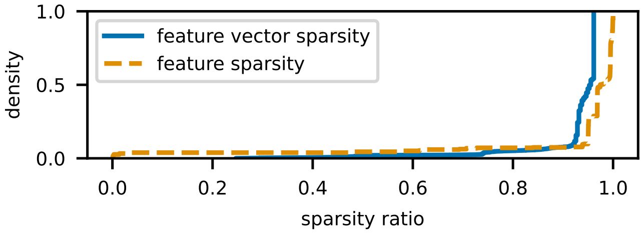


Figure 5.2: CDF of feature sparsity (how often a feature equals 0 in the dataset) and feature vector sparsity (what proportion of a feature vector is zero-valued).

sector from our dataset (see Section 5.4.2).

A potential drawback is that most organizations are located in the Netherlands. Nonetheless, considering that the Netherlands consistently scores well in cybersecurity maturity [159], we view this as advantageous. The high level of security maturity may obscure the connections between external and internal signals. This is beneficial for our analysis; if we identify a pattern here, it is likely that these findings will apply more readily to regions with less mature cybersecurity practices.

5.5.2. ON FEATURE SPARSITY

Many of the organizations in our population are relatively small in terms of network size. They are, thus, less exposed to external threats and have a lower chance of appearing in abuse datasets. We mitigate these drawbacks by collecting more features than in previous work: a total of 338 features, as opposed the 186 in [133]. Still, we find that the feature space remains highly sparse (see Figure 5.2). We define two different types of sparsity: i) *feature vector sparsity* refers to the number of features within an observation that contain no data; ii) *feature sparsity* refers to the frequency with which a feature contains no data across all observations.

Only 14 features have a sparsity ratio under 0.5, including the number of unique responsive IP addresses, the number of open ports, and 12 features depicting “normal” behavior in various abuse datasets as discussed in Section 5.4.2. Among these, the unique responsive IP addresses and open ports are significant, each with a sparsity ratio of 0.48. These features are predictive [73, 133], suggesting their importance to the target variables when non-zero values are present.

Furthermore, a preponderance of feature vectors are sparse as well: 95% of all feature vectors are at least 80% zero-valued. This further indicates that the features, however sparse they are, can provide critical discriminative information that aids in making accurate predictions.

Many of the organizations in this subset are smaller regional companies with relatively few allocated IP blocks: the median network size in the sparse vectors is 24, while that of the non-sparse vectors is 804. These smaller companies (in terms of organization size and IP block allocations) are much less likely to have enough Internet exposure to possess the necessary external signals for proper feature collection.

5.5.3. Security incident types as target variables

We focus our analysis on the security incidents that are labeled as true positives by the MSSP. These are labeled according to their estimated severity—low risk, high risk, and successful hack attacks. Successful hack attacks are most straightforward; they include, e.g., detected indicators of compromise, ransomware traffic, and successful exploit data in network traffic. High-risk incidents include early signs of intrusion or signals that precede a data breach, such as attempts at compromise, internal network scans, and botnet traffic. Finally, low-risk incidents are composed of seemingly less precarious events such as network policy violations, adware, and generally suspicious network traffic.

Using the sliding window method outlined in Section 5.4.2, all incidents within each window are aggregated. Consequently, incident counts for each window and category become separate target variables. Simply put, the model forecasts the number of each incident type occurring within specific time window.

The occurrence of the different types of incidents varies as expected, as illustrated in Figure 5.3. The figure shows, generally speaking, that the number of incidents decreases as the severity of its label increases. This is also reflected in Table 5.2, which contains the mean, median, and standard deviation of the three categories of incidents: low-risk incidents occur more often and more erratically than high-risk incidents, which, in turn, occur more often and more erratically than successful hack attacks.

Figure 5.3 also displays the erratic behavior of security incidents, possibly due to new developments in the global threat landscape. It also demonstrates the need to keep a prediction system up to date with external network measurements and internal incidents, so the model can recognize similar behavior and outliers in the future.

Table 5.2: The mean, median, and standard deviation of the low risk incident, high risk incident, and successful hack attack counts per organization in each sliding window.

	Low-risk incidents	High-risk incidents	Successful hack attacks
Mean	3.41	2.78	0.20
Median	1.00	1.00	0.00
Standard deviation	8.26	6.88	1.57

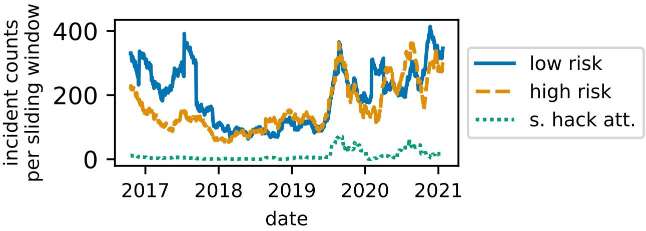


Figure 5.3: The number of low-risk incidents, high-risk incidents, and successful hack attacks per sliding window over time. All incidents within the sliding window have been added together, leading to a higher number being plotted on this graph.

5.5.4. TRAINING AND TESTING PROCEDURES

Although the alert and incident data is split per organization, we cannot train an individual model for every organization separately. This is due to the lack of data per organization, where the available data varies from a single sliding window to just 1,500. For data with 338 features, it is unlikely that a model will be able to extract from it any type of meaningful information. Thus, we aggregate all data into a single dataset, creating a “pseudo time series” of all incidents that are investigated by the SOC. The process is as follows:

1. For each organization, collect and compute the 338 features within every 60-day time window. For each of these data points, count the number of low risk incidents, high risk incidents, and successful hacks that occur in the 60-day window following each data point, and add to the record.

2. Concatenate all separate datasets into a single dataset.

3. Sort the resulting dataset chronologically.

Each data point then contains a single organization’s 338 observed features in a specific time window and the total number of low-risk incidents, high-risk incidents, and successful hack attacks the organization will encounter in the next time window. While not a true time series in the sense that every feature vector is a direct continuation of its predecessor, the data is still sorted chronologically.

The feature vectors, in the form of sliding windows, are not completely independent of one another. In fact, it is due to the smoothing of the data through the creation of sliding windows for each organization that many feature vectors are highly correlated with previous feature vectors from the same organization.

We cannot randomly select training and testing data for sequential data as we would with non-sequential data. Instead, we split the data chronologically, training the model on earlier data and testing it on later data. This approach allows the model to use past observations to predict future events, mimicking real-world scenarios.

Receiving constant, unchanging security signals offers no insights into shifts in an organization's security posture. Only by noting changes in external features and associated internal security events can system administrators effectively prepare for future incidents.

As Internet traffic evolves, these models, being trained on historical network traffic data, may experience accuracy degradation as new data with different patterns emerges—phenomena known as *concept drift* and *data drift* [150]. This indicates that such models should be continuously trained to recognize similar occurrences in the future. Based on this information, we construct the following train and test procedure:

1. train the model using the initial 50% of the feature data;
2. test the model on the subsequent weeks' worth of data;
3. add the test data to the training data;
4. train the model using the new data;
5. repeat 2) through 4) until all test data is used.

We purposely choose to train and test the model in weekly steps. Performing this process after every observation is realistic neither from a computational nor operational perspective.

5.5.5. Model selection

External features do not linearly relate to internal security incidents; variations in one do not directly alter incident counts and vary by organization. Features interrelate and affect incident manifestations, rendering simple regressions inappropriate. No standard distribution adequately fits feature, alert, or incident data.

Autocorrelation analysis reveals highly autocorrelated features at the organization level, indicating that current values depend significantly on past values. However, this autocorrelation disappears once all organization data is aggregated into a single dataset. Standard time series models are, therefore, not appropriate, and we must select a model that is designed to process more complex data.

It is also critical for the model and its output to be explainable. Explainable models are essential as they enable analysts to understand, justify, and improve decisions, and effectively investigate incidents. Additionally, models must be robust to outliers to handle erratic security incidents.

Decision tree-based models meet all requirements. Indeed, Liu et al. demonstrate the effectiveness of a random forest classifier approach [133]. They used a binary classifier for their coarse-grained incident data, mapping the predicted probability of future incidents to a binary outcome or using it as a risk metric by applying a threshold.

Table 5.3: XGBoost hyperparameters after tuning.

colsample_bytree	subsample	max_depth	n_estimators	learning_rate
0.97	0.69	9	248	0.12

However, our data contains multiple different security incident measurements from the internal state of networks, and we aim to predict how many of such internal incidents will occur in the near future. Since our data is more fine-grained, every organization will suffer a security incident at least once within a set time window, making a binary classifier trivial to construct. Though we do not stray from a decision tree-based approach, we ultimately opt for XGBoost [47] to construct a regression model, given its demonstrated effectiveness in Internet measurement tasks [103, 196].

5.6. RESULTS

We used the XGBoost regression model to predict the number of low-risk incidents, high-risk incidents, and successful hack attacks an organization will face in the next 60 days. These types of incidents differ not only in severity, but also in what features, and to which extent they are taken into account when performing a prediction. We analyze the model's performance, feature importance, and explainability in predicting various types of security incidents, noting its strengths and limitations.

5.6.1. PREDICTION PERFORMANCE

To find the optimal model configuration, we use a combination of random search and grid search, with 3-fold cross validation for each search, to tune the XGBoost hyperparameters. See Table 5.3 for the list of tuned hyperparameters.

As mentioned in Section 5.5.4, we use a type of sequential training and testing procedure to evaluate the performance of the model. All organization security incident data is aggregated into a single dataset, creating a pseudo time series with which we train the model. Though not a true time series, the dataset retains a sequential structure due to the sliding window features. We then also train and test the model sequentially. Each iteration in this process makes predictions for a week's worth of traffic, after which the model is trained with the additional data so it can learn from potential new developments in the threat landscape.

To determine the size of the time window (see Section 5.5.5), we test model performance using both a 30-day and 60-day window. We find decreased model performance using the 30-day window, e.g., obtaining R^2 scores of 0.15, 0.42, and 0.26 for low-risk incidents, high-risk incidents, and successful hack attacks, respectively. Thus, we perform all further

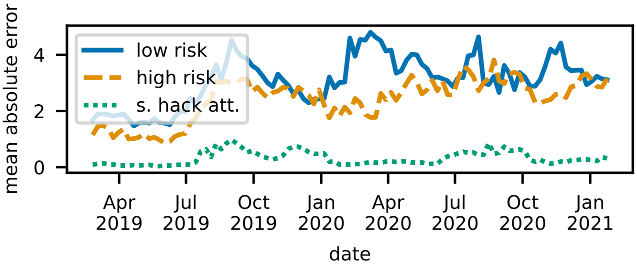


Figure 5.4: The mean absolute errors for each weekly training and testing iterations.

analysis using the 60-day window, as in [133].

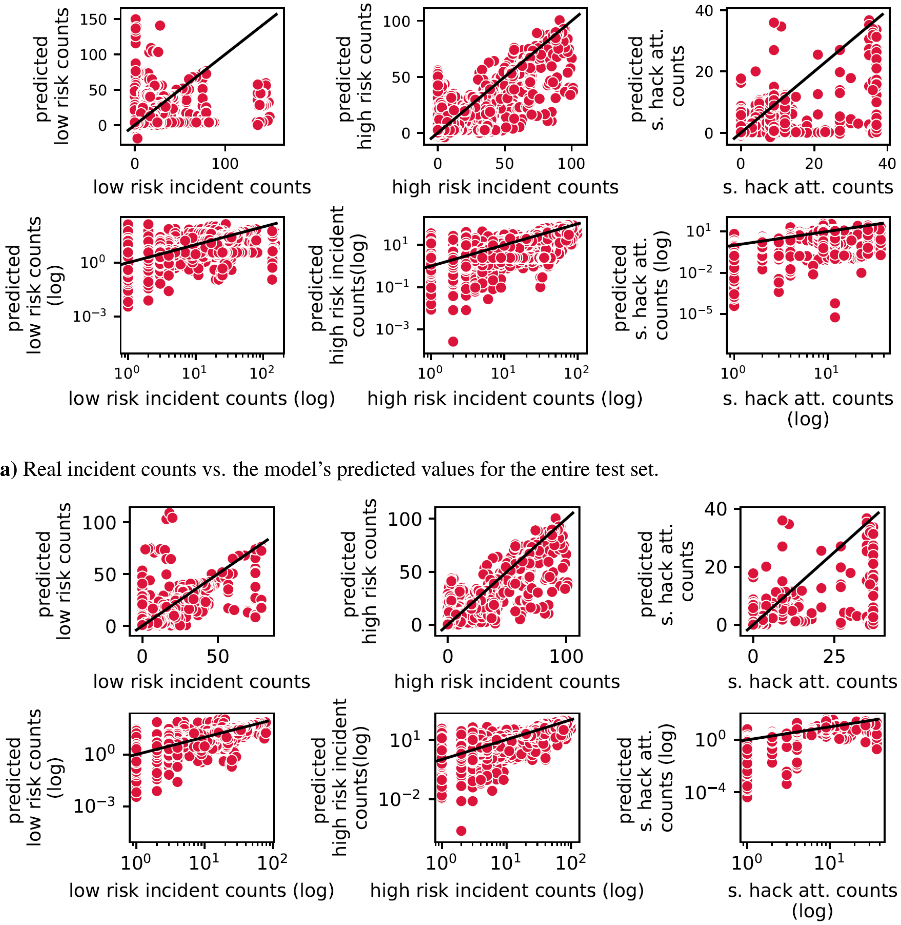
Figure 5.4 shows the mean absolute error for each week. All three curves for different incident types display similar trends and are highly correlated. Previous research has discovered that the number of different types of internal security incidents over time are highly correlated [239]. This suggests many incidents within organizations might be unaffected by external security measures, potentially explaining sudden error spikes when external data fails to predict these independent incidents.

In Figure 5.5a, the graph displays actual security incident counts for the test data, categorized into three types, alongside their log-scaled representations. A linear relationship between actual and predicted values is evident, particularly in the log-scaled plots. However, the classifier had difficulty accurately identifying many outliers in the predictions.

The model struggles with a significant number of incidents, unable to provide reliable predictions. This issue is evident from the plot of low-risk incident predictions (Figure 5.5a), characterized by lines diverging from the bottom-left corner. These problematic instances are linked to over 20,000 sparse feature vectors, as detailed in Section 5.5.2, where nearly identical vectors with the same non-zero features produce different incident counts, rendering the data ineffective for predictive modeling.

To examine prediction performance without sparse, unusable data, we remove the feature vectors from the dataset [30] that have a sparsity ratio of 0.95, yielding the graphs shown in Figure 5.5b.

Table 5.4 shows the model's mean absolute error (MAE), root mean squared error (RMSE), and R^2 scores for the testing period. Given the large range of potential incident counts, the MAE provides a good indication as to the performance of the model, since it is less sensitive to outliers and skewness in the data distribution [99]. The RMSE and R^2 scores seem to paint a bleaker picture, both presenting relatively high and low values, respectively. However, RMSE is suboptimal when the errors are not Gaussian [45], and Figure 5.6 illustrates the lack of such a distribution for all types of errors.



(a) Real incident counts vs. the model's predicted values for the entire test set.

(b) Real incident counts vs. the model's predicted values, excluding feature vectors with 95% or higher feature sparsity.

Figure 5.5: The real incident counts in the test data plotted against the value predicted by the classifier. While there are clear linear relationships between the real and predicted values, each category of incidents contains numerous outliers. The black diagonal line illustrates the $x = y$ line. The closer a dot is to this line, the better the prediction.

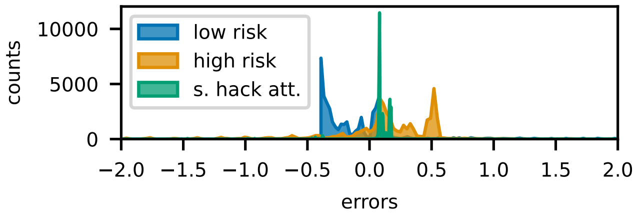


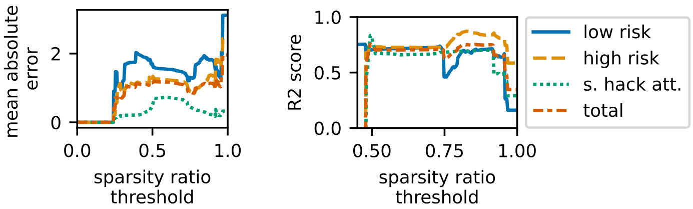
Figure 5.6: The mean normalized error distribution for all types of incidents.

The model underperforms for low-risk incidents and successful hacks but does better for high-risk incidents. The low performance metrics stem from making 20,000 predictions on sparse feature vectors. Excluding these vectors improves the scores substantially: MAE decreases by 59%, 34%, and 39%, and R^2 scores increase by 306%, 37%, and 66% for low-risk incidents, high-risk incidents, and successful hacks, respectively Nevertheless, this could be a reflection of not only the complications introduced by data sparsity, but also of the ad-hoc nature of incident labeling.

Table 5.4: The mean absolute error (MAE), root mean squared error (RMSE), and R^2 scores for the three different types of incidents. This table includes results for both the entire test set and the test set excluding all feature vectors with a 95% sparsity ratio or higher.

	Entire test set			**Excluding ≥ 95% sparse feature vectors**		
	MAE	**RMSE**	**R^2 score**	**MAE**	**RMSE**	**R^2 score**
Low risk	3.11	8.33	0.16	1.27	3.73	0.65
High risk	2.43	5.28	0.59	1.61	4.38	0.81
Successful hack attack	0.33	1.84	0.29	0.20	1.62	0.48
All incidents	1.96	5.15	0.35	1.03	3.24	0.64

Table 5.4 shows poor performance in predicting successful hack attacks, improving with less sparse data, as indicated by an R^2 score of 0.48. Figure 5.7 demonstrates that adjusting the sparsity ratio threshold enhances prediction accuracy. For instance, when setting the threshold at 0.9 (i.e., exclude all feature vectors that are over 90% sparse or higher), the successful hack attack R^2 score increases to 0.70. Likewise, prediction performance for low and high-risk incidents improves as well. Due to the threshold shifting, a mere 10% of the test set is now accessible, thereby limiting the availability of this model's functionality to a select group of organizations possessing sufficient Internet exposure.



(a) Feature vector sparsity threshold ratios vs. MAE.

(b) Feature vector sparsity threshold ratios vs. R^2 score.

Figure 5.7: Feature vector sparsity threshold ratios plotted against the model's total MAE and R^2 score. The threshold values on the x-axis represent at what sparsity ratio to discard a feature vector. For instance, at $x = 0.95$, any feature vector that is over 95% sparse is discarded from the dataset.

5.6.2. FEATURE IMPORTANCE

Examining feature importance is critical for understanding a model and interpreting predictions. The XGBoost module [248] provides feature importance functionality based on 1) feature usage in tree nodes, and 2) reduction in training loss when a feature is used. This can overestimate feature importance if features are frequently used without significant prediction changes, potentially misrepresenting true importance [139]. Additionally, it lacks functionality to explore feature interactions and individual prediction decisions.

Instead, we use SHAP values as proposed by Lundberg and Lee [139] to analyze feature importance. SHAP values take a game-theoretical approach to analyze feature attribution: each feature is a "player" that contributes to a "game", that game being a single prediction. Additionally, while other methods offer a global view of important features, SHAP values reveal the intricate interactions and effects of each feature on individual predictions.

No single feature will decide outcomes alone. Instead, different feature configurations will influence model predictions. Using this SHAP framework allows examining feature interactions, crucial for understanding how certain combinations of external network signals influence an organization's security internal state.

Figure 5.8 shows the mean SHAP values for the top 10 features influencing predictions. Different incident types are affected by different features and to varying degrees. The magnitude differences stem from varying incident counts, with low-risk and high-risk incidents occurring more frequently than successful hack attacks (see Figure 5.5). Despite the sparsity of most features, only two—unique IP addresses and the number of open ports—of the 15 less sparse features are shown. This highlights the high relevance of sparse features in security incident prediction.

Table 5.5 groups the features into seven categories as described in Section 5.4 (excluding

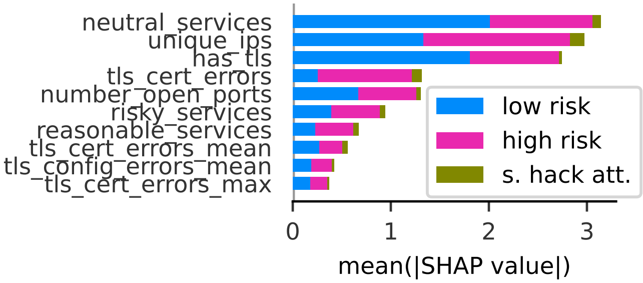


Figure 5.8: The mean SHAP values for the 10 most important features for low-risk incidents, high-risk incidents, and successful hack attacks. The mean SHAP values represent the average impact each feature has on predictions.

the size feature). The live hosts category is ranked most important due to the sparsity in other features. Second are the services features, with neutral services contributing more to predictions than reasonable and risky services [73] (see Figure 5.8). Mismanagement symptoms are third, despite being extracted from only two of the five original categories, indicating poor network hygiene and adherence to best practices reflect an organization's security state. This is notable as our mismanagement features are based on longitudinal measurements, unlike Liu et al. [133]. Malicious activity and persistence categories are less significant compared to previous work, likely due to the data sparsity, as many organizations do not appear in abuse datasets, resulting in no persistence features to extract.

Figure 5.9 shows a dependence plot on the effect of the number of open ports on high-risk incident predictions and its interaction with TLS certificate errors. Each dot represents a prediction instance with its corresponding open port count. The vertical dots at $x = 0$ occur due to the high sparsity of features, causing many features to be zero-valued. The figure

Table 5.5: Normalized feature importance based on the computed SHAP values.

Feature category	Normalized importance (mean |SHAP| value)
Live hosts	0.4332
Services	0.2986
Mismanagement symptoms	0.1430
ShadowServer features	0.0431
Persistence full time window	0.0349
Malicious activity	0.0191
Persistence last 14 days of time window	0.0281

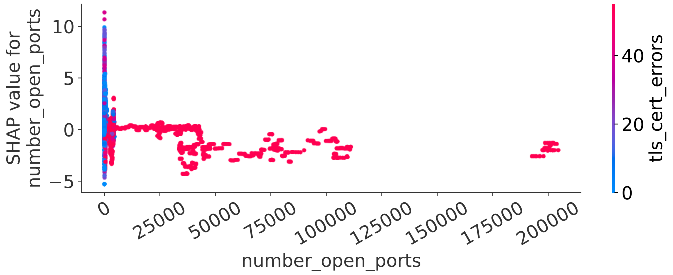


Figure 5.9: Dependence graph for high-risk incidents that plots an organization's number of open ports against the SHAP value for that specific feature value. The secondary *y*-axis, and coloring, indicates how the presence of TLS certificate errors interacts with the primary feature within the model to make a prediction: as the number of open ports increases, the occurrence of TLS certificate errors increases the predicted high-risk incident count.

shows that a small number of open ports can either increase or decrease predicted incident risk, but this effect levels out as the number of open ports increases. At higher open port counts, there is a slight, constant increase in predicted incident risk. The interaction with TLS certificate errors shows that at zero open ports, TLS certificate errors have no effect on predicted incident counts, as no TLS certificates are found in the first place. As the number of open ports increases, TLS certificate errors increase the predicted incident count.

For many feature interactions, this pattern of no secondary effects at the $x = 0$ mark holds. A notable exception we found, though, can be seen in the interaction between the number of reasonable services and the average persistence in the "normal" spam region (see Section 5.4.2). Regardless of the number of reasonable services (or lack thereof) that an organization has publicly accessible, having hosts actively sending out spam emails (and having those IP addresses present in a spam blacklist) increases the predicted incident counts. While there are many more feature interactions, it is unfeasible to examine them all.

5.6.3. Prediction explainability

In addition to a global view of the model's inner workings, we also examine the output of the model on an individual prediction level. In a real-world environment, this would allow analysts to identify vulnerabilities, patterns, or indicators of potential threats or anomalies in real-time. Thus, we further sample a number of noteworthy predictions and examine the decisions taken by the model. The figures referred to in this section (Figures 2 and 3) are located in Appendix B.2.

Figure 2 shows the decision plot for the model's false prediction of 20 high-risk incidents, when none had occurred. Most contributing features are related to insecure signals or malicious activity. At first glance, one could certainly understand the predicted number of 20, given the large values for many of these features. This specific case corresponds to a large cloud infrastructure provider. Given this type of company, it is likely that many of the ranges allocated to it are not entirely managed by the organization, but instead (sub)allocated to other entities. This could explain why no incidents were picked up by the MSSP. This, again, stresses the necessity of accurately mapping out the external networks of organizations, and, additionally, identifying IP block suballocations. Unfortunately, this task is challenging when, e.g., telecom operators or (cloud) hosting providers are involved, due to the inherent difficulty of differentiating between official allocations and (unofficial) suballocations of IP blocks.

In Figure 3 we see a decision plot for numerous successful hack attack predictions of varying accuracy. These correspond to the cluster of successful hack attacks visible as a vertical line on the graph's right-hand side in both Figure 5.5a and 5.5b. All incidents originate from a single multinational media company. Despite over a hundred responsive hosts with numerous TLS certificate and configuration errors and other problematic features, the model could not predict their occurrence. Closer examination revealed that the aforementioned problematic features were present in the organization's network since well before the occurrence of said incidents. Interestingly, the successful hack attacks led to a decreasing trend in responsive hosts and a reduction in many problematic features (e.g., TLS certificate and configuration errors) within the organization network.

5.7. Discussion

5.7.1. Organizations' Internet exposure

An organization's Internet infrastructure stands at the front line of both its success and vulnerability. As seen in Figure 5.8, the feature with the highest importance (according to SHAP values) for all three types of incidents is the `unique_ips` feature, representing the count of an organization's reachable IP addresses. This emphasizes the importance of having a complete view of an organization's Internet presence. Not only because of its high importance, but also because the accuracy of all the other features relies on the accurate measure of this particular feature.

Collecting and manually verifying the IP blocks allocated to different organizations was time-consuming but resulted in accurate features. Manual verification of a 50% sample showed that the identified IP blocks from the MaxMind GeoIP [144] database are highly accurate. This is a 50% sample containing the smallest organizations in terms of network size, and it is perhaps precisely the larger organizations that require manual verification, given the larger amount of IP blocks allocated to them. Due to manpower and time constraints, however, this was unfeasible to perform.

In this work, we surprisingly found ourselves not utilizing the feature that was the direct result of this process, specifically, the total size of an organization's allocated IP blocks. While the claim by Liu et al. that the size of an organization "to some extent captures the likelihood of an of it becoming a target of intentional attacks [133]" seems valid, they are contrary to our findings.

We found that this feature did not have the intended effect for our use case. Instead, it unexpectedly influenced the model's handling of highly sparse feature vectors. The model erroneously learned a direct relationship between the size of a network and the number of incidents that would occur within that network, assuming a larger network size meant more incidents. This inference, however, is a simplification and not reflective of reality. A myriad of factors can impact the frequency of incidents within a network, and it is not necessarily tied to its size. Therefore, while the feature might hold some predictive power under certain conditions, it is critical to understand that network size does not unilaterally determine the number of incidents. Dealing with such sparse data is discussed in Section 5.7.2.

5.7.2. Lack of external signals

One of the most striking facets we encountered during our data collection process was the considerable degree of sparsity present within the data. This notable characteristic, however, was not highlighted in the work done by Liu et al. [133] They utilized data primarily from large-scale breaches, predominantly containing larger organizations in terms of scale and network size than the ones in our own population. Such organizations, due to their scale, would naturally have a denser array of network assets, reducing the observed data sparsity.

Our scenario, however, starkly differs, since the MSSP we collaborate with boasts a diverse portfolio of clients, spanning from large multinational corporations to smaller, regionally-focused enterprises. Given this wide variation in the scale of operations, it is unsurprising that the smaller entities among our subject group have fewer network assets that are publicly accessible. Consequently, the dataset we work with exhibits a high degree of sparsity, a unique aspect that calls for specialized handling during analysis.

This somewhat calls into question the claim of being able to predict what goes on inside a network purely from external network signals. For many organizations, there were no external network signals to observe, yet they were still plagued with the same types of incidents as organizations with much higher Internet exposure. Finding links between external signals and internal incidents and security posture is impossible if there are no external signals from which to extrapolate or infer these links in the first place. Put plainly, as long as there is nothing to see, there is nothing to be learned.

Another consequence of sparse features is the increased propensity of organizations with smaller Internet exposure to share similar feature vectors (see Section 5.5.2). While they may share external network characteristics, internal security events will certainly differ. This provides a model with training data with inadequate training data: similar features, but wildly different target variables. This will have a detrimental effect on the quality of the resulting model, as is exemplified by the familiar saying “garbage in, garbage out” [195].

The performance increase that results from removing sparse feature vectors suggests either one of two possibilities: 1) not enough external features have been collected, or 2) there are security issues that are simply not noticeable through external measurements. To verify which of the two is the case, manual verification of individual incidents must be performed, which is a possibility for future work.

Be that as it may, setting the sparsity ratio threshold to match a specific desired performance is a workable method. The drawback is, of course, that the number of usable feature vectors decreases as the sparsity ratio threshold is adjusted.

This suggests, then, that a system like this is only of value to larger enterprises, or, at least, enterprises with a large degree of Internet presence. In such cases, data sparsity would cease being an obstacle. Alternatively, sparse data can be ignored altogether. Though these changes are trivial to implement, further work is needed to fully understand the scope and effect of sparse data.

5.7.3. Incident severity

In our analysis, we are tasked with forecasting three distinct values. Each of these corresponds to the different labels assigned by the MSSP to incidents that have been confirmed as true positives. These labels are not standardized, but influenced by factors such as the organizational culture and work environment, and an analyst’s subjective judgment of what constitutes low or high risk.

While analysts do seem to have a clear idea of what comprises high-risk incidents, the blurriest line seems to be that between low-risk incidents and non-incidents. This is reflected by the performance metrics of low-risk incidents in Table 5.4. A possible consequence of this subjectivity is the inadvertent inclusion of noise (i.e., non-incidents) into low-risk incidents that can muddy the relationships between external network signals and internal security events. A straightforward remedy for this is creating concrete guidelines on how to classify an incident’s severity, and applying these guidelines consistently.

The model performs much better when predicting high-risk incidents, both including and excluding sparse feature vectors, implying that external network signals carry more information about high-risk incidents than either of the other two incident types. One possible explanation can be found in the collection of incidents that are included in the low-risk category and not in others: we notice a large number of adware-related incidents exclusively classified as low risk. In contrast, high-risk incidents include Trojan traffic, internal Nmap scans, and exploit attempts, which are more directly related to an organization's security and thus more easily connected to external network signals. One could argue that the adware-related low-risk incidents are not as easily attributable to such external network signals; either more external features need to be collected to uncover the latent relationship, or they are entirely disconnected. A closer inspection of these incidents and their feature space is needed for definitive conclusions.

Another finding that stands out is the relatively poor performance of the successful hack attack predictions at higher feature sparsity thresholds. While performance improves with adapted thresholds, the substantial difference, especially compared to high-risk incidents, is intriguing. This performance increase at a different sparsity threshold suggests that predicting incidents of this severity requires a richer and more complete view of an organization's external network, provided the organization has sufficient Internet exposure, pointing once again to the same limitation that restricts the usability of incident prediction models to the aforementioned types of organizations.

Considering incident types separately allows for more tailored responses to incidents of different severity, e.g., prioritization of response, resource allocation, enhanced strategic planning. This is currently affected by the distinct model performance between the considered incident categories. Predicting high-risk incidents is, clearly, of higher value, since those (and successful hacks) as they can potentially cause significant financial and reputation damage, while low-risk incidents are often just nuisances. Discovering that high-risk incidents are more directly linked with external network signals opens the door to more effective proactive prevention measures and strategic security investments.

5.7.4. Incident counts vs. incident likelihood

We opted to predict incident counts rather than likelihood of incident occurrence due to the nature of the MSSP's incident data. Instead of taking all incidents as a whole, we investigate each category separately. Low-risk incidents are common, making binary classification pointless. Instead, targeting the counts allows us to examine potential links between them external signals, with the added benefit of making full use of the finer-grained incident data. We extend the same methodology to the other two categories. The value of our work lies in our demonstration and validation of such links. Consequently, incident count is a representation of the organization's internal state that we can infer from external data.

Predicting counts enables organizations to compare risk and security posture to their own (or sector-wide) incident baseline. Quantifying the expected number of incidents also allows for the analysis of trends and taking measures accordingly, e.g., seasonality adjustments and resource allocation. Tracking these counts over time reveals how well an organization adapts to threats, with increasing trends potentially indicating delays in security updates, and decreasing trends suggesting effective adaptations.

By demonstrating the links between external signals and internal indicators, we also open the door to the creation of risk metrics that estimate incident likelihood, especially for internal high-risk incidents. Such risk metrics make security posture comparisons more generalizable across organizations, as they provide a standard, empirical measurement of security posture, independent of size or industry, rather than a single number that may differ in magnitude from one organization to the next.

Risk metrics can function as quality labels for security. Organizations can then be incentivized to obtain better risk scores through several means, improving overall security on the Internet. For instance, just as for energy labels [76], governments can provide subsidies to more secure organizations to incentivize the attainment of better risk scores. Insurance firms leveraging risk scores to assess an organization's security posture can incentivize improvements in security measures by offering lower premiums for better scores. Market pressure can also be exerted by consumers, who may prefer to do business with organizations less liable to data breaches.

5.8. CONCLUSION

This paper presents significant findings, showcasing the vital role that external network signals can have in incident prediction. By leveraging these signals, we have demonstrated to what extent it is possible to have accurate predictions of actual security incidents, empowering proactive defense measures and enhanced risk management. This offers organizations a transformative approach to cybersecurity beyond traditional compliance-driven methods.

We have demonstrated the efficacy of employing a gradient boosting regressor to predict cybersecurity incidents. The model's ability to forecast incident counts for low-risk, high-risk, and successful hack attacks provides valuable insights for proactive security measures. In contrast to prior research, our study revealed that external features lack sufficient predictive power to accurately forecast successful attacks. However, they did demonstrate considerable accuracy in predicting incidents driven by both low and high-risk incidents. Although we identified challenges related to sparse data and feature interactions, addressing these issues could significantly enhance prediction accuracy. Notably, seemingly insignificant features hold substantial predictive power, becoming critical components in fortifying an organization’s cybersecurity resilience.

For instance, such a system can substantially enhance the effectiveness of a SOC by functioning as a proactive early warning system for client organizations. By forecasting potential security incidents, it empowers SOCs to alert client organizations ahead of time, enabling them to bolster their defenses and mitigate risks before they materialize. Additionally, not only does this predictive capability serve as a shield against potential threats but can also act as a valuable tool in forecasting the workload of the SOC by helping to anticipate the volume and complexity of incidents that may arise, allowing for better planning and management of resources.

Our study lays the foundation for integrating the predictions from our model to construct a comprehensive risk metric, enabling organizations to gain a holistic understanding of their security posture. While additional challenges exist in standardizing predictions and defining numerical differences between severity levels, overcoming these obstacles would provide organizations with a powerful tool to assess and manage their security risks effectively.

6

CONCLUSION

This dissertation studied how to make security predictions about the internal state of a network through only external network observations. In Chapters 2 to 5 we have presented four peer-reviewed studies, each of which addresses a portion of the following overarching research question:

How can internal security incidents be predicted through the leverage of external network signals?

In the final chapter of this dissertation, we sum up the work presented in the previous chapters and their contributions. Next, we discuss how the studies we conducted answer the aforementioned research question. We then elaborate on the practical implications for governance and policy that the findings of our work have uncovered. And, finally, we discuss potential avenues for future research that our work and findings have opened up.

6.1. SUMMARY OF THE FINDINGS

CHAPTER 2 – EXTERNAL ASSET DISCOVERY

In this chapter, we aimed to compile and systematize novel asset discovery techniques. Since asset discovery is typically not the main focus of network measurement research, it is overlooked, even though the process of asset discovery is crucial to many a cybersecurity task.

To address these issues, we presented a framework for asset discovery that proposes a syntax to make explicit the steps in the asset-discovery process. We then used this framework to systematize all asset discovery techniques designed or mentioned in literature published in 14 leading academic venues between 2015 and 2019. This framework and systematization provide a natural way for researchers to identify gaps in their study design, thereby creating opportunities to build on earlier efforts by broadening the set of assets discovered. Furthermore, our systematization of recent advances in active asset discovery can help researchers select relevant techniques for the individual steps of our framework based on their needs. Finally, we applied our framework to various use cases, illustrating how techniques can be combined and how to identify where gaps remain.

CHAPTER 3 – NIDS RULES AND INCIDENTS

This chapter aimed at shedding light on another oft-overlooked yet vital aspect of network security, namely signature-based NIDSs and the rulesets they employ to detect threats. Using 13 years of NIDS rule management data and internal alert and incident data from an MSSP, we analyze the evolution of said rules and rulesets, as well as how the rules influence the detection of threats and the investigation of security incidents. This analysis is complemented by a number of in-depth interviews conducted with rule writers and SOC analysts from the MSSP to verify the findings.

We find that, overall, barely a fifth of all rules are meaningfully updated throughout their lifespan (i.e., in a way that alters their detection capabilities), either to account for changes in the threat landscape, or to reduce the amount of generated alerts by making rules more specific to certain threats, thereby alleviating the workload on the SOC and its analysts. Furthermore, a half percent of rules are responsible for 80% of all alerts. Of all these alerts, we find only a 1.2% are significant enough to warrant closer investigation, of which only a fourth carry any risk to an organization.

CHAPTER 4 – NIDS RULE AND INCIDENT MANAGEMENT PROCESSES

Complementary to the previous chapter, in this chapter we studied the organizational processes surrounding the management of signature-based NIDSs and their respective rulesets. Although many of these processes are technical in nature, they are carried out manually by a team of professionals, and end up determining the types of security incidents and the degree to which they detected and resolved. To understand the way in which internal security incidents are influenced by internal management processes, we conducted 17 in-depth interviews with security professionals from different organizations and industries about their roles, workflows, and heuristics when working with the NIDSs within their organizations.

We find that there is no single manner in which all SOCs are managed. Many different factors need to be managed and balanced against each other within the context of, e.g., the SOC's rule management processes and resource availability. This, in turn, is also balanced against customer expectations. Taking these points into account, we conclude by presenting a number of recommendations for internal management processes to improve the effectiveness of SOC teams and their services.

CHAPTER 5 – SECURITY INCIDENT PREDICTION

Finally, this chapter investigated how a collection of external network signals can be used to predict the occurrence of internal network security incidents. To this end, we extract 338 features from nearly five years of external network scan data from providers such as Shodan and ShadowServer, malicious activity reports and abuse datasets, and use them to train a model to predict internal security incidents (similar dataset as used in Chapter 3). Instead of binary classifications, this system is able to predict with high accuracy the number of internal security incidents within a certain time span, thereby demonstrating the predictive power of external network data. Additionally, we find that this performance is highly dependent upon the sparsity of the input features extracted from the external data and abuse datasets. These findings allow organizations to identify troubling security signals ahead of time, thereby potentially be kick-starters for organizations to take a more proactive approach to their network security.

6.2. FROM EXTERNAL SIGNALS TO INTERNAL SECURITY

Through the findings in the previous chapters, we advance the state of the art surrounding empirical risk estimation. While previous work has demonstrated the effectiveness and predictive power of external network signals when observing and exploring large data breaches, none had examined such relationships with internal security incidents. To achieve this, we identified three primary gaps in current scientific knowledge (see Section 1.3). To address the aforementioned gaps, we reflect and elaborate on our findings and how they ultimately answer our overarching research question: *How can internal security incidents be predicted through the leverage of external network signals?*

First, we have discovered a myriad of asset discovery techniques buried within recent academic literature. These are discovery techniques that shed light upon an organization's publicly accessible digital infrastructure, both known and unknown. To our knowledge, these technological advancements in asset discovery are rarely implemented by organizations to keep track of digital assets. Unrecorded IT infrastructure is a problem that remains rampant [189]. Not only is this information useful for adversaries, but also for defenders. Adversaries often exhibit ingenuity in carrying out their plans at the expense of an organization. These are techniques developed by academics worldwide for specific research purposes, which are often forgotten afterwards. Consequently, the security community in the future is forced to reinvent the wheel whenever the ability to track those specific assets becomes necessary for the security of an organization. Being able to not only accurately identify assets and controls, but also seamlessly integrate such techniques into existing asset discovery pipelines, becomes vital for organizations. This closes the gap between themselves and potential adversaries, maintaining a balance in this ongoing cat-and-mouse game.

Second, we discover that organizations prioritize three aspects when it comes to optimizing incident detection and improving security: 1) maximize threat landscape coverage, 2) maintain NIDS rules as up-to-date as possible with current developments in security, and 3) minimize false positives as to reduce security analyst workload. These three factors do not exist in a vacuum, however, and many trade-offs are made depending on, for instance, the nature of the threat and the ability of the analysts to detect said threats. To give an example, minimizing false positives implies making rules more specific to particular threats, thereby reducing the probability of unintended triggering on non-malicious traffic, while most likely sacrificing a degree of genericness that allows it to detect a wider range of similar threats and cover more of the threat landscape. This is not a static phenomenon, but rather a dynamic process that evolves not only due to the aforementioned threat landscape, but also the shifting abilities, resources, and priorities of the team of analysts and the organization as a whole.

Different factors influence how susceptible some organizations are to certain attacks, and not everything that happens warrants swift and decisive action. Analysts react to various alerts and incidents differently, depending also on the organization. These are systems and processes managed and maintained by human professionals, making these aspects inseparable from the human factors governing them. After a thorough examination of these human factors, we find that the results of the quantitative analysis are reflected in the personal experiences of the analysts. This suggests that the maintenance and management practices implemented by organizations are intentionally and effectively carried out by the security professionals. For instance, the aforementioned aspects of threat landscape coverage, rule update processes, and the minimization of false positives through the creation of specific rules are priorities at both the technical and organizational levels. Furthermore, many of the organizational processes that ensure the effectiveness of the team and their security systems rely on the individual intuition of security analysts. However, we find that there is an industry-wide lack of strictly-defined feedback loops within the incident detection process at the organizational level. This limits the opportunity for security analysts, whether expert or novice, to gain experience from a wide range of security-related situations. Since intuition is built off experience in a certain field, this impedes the degree to which teams can effectively carry out their tasks. This calls for the implementation of organizational processes that can aid in the transfer of knowledge from more experienced analysts to novices. Such processes could include the creation of playbooks, distribution of SOC process documentation, and setting up systems for internal and external collaboration. Still, these NIDSs remain vital to the security of organizations, as evidenced by their presence in almost all security departments and their reliance on these systems for providing security to their clients.

The incidents produced by NIDSs that security analysts respond to are influenced by the corresponding organization's Internet exposure and the information extracted from such exposure. This opens the door to studying the relationships between the two. While signals do not directly indicate how secure organizations are, they can be potential symptoms of wider security issues affecting such organizations. For instance, an out-of-date SMTP server is not inherently a security issue, but it may indicate inadequate organization-wide patching practices. This, in turn, may suggest a general low level of security maturity. The phenomenon we aim to capture is not causality but rather signals indicative of an underlying cause. We find that the relationship between external network signals and the internal state of a network is significant enough for the former to have predictive power for the latter. Using a vast array of external signals collected from a variety of organizations, we have developed a system capable of predicting with high accuracy the number of security incidents that each respective organization will face during a specific future time window. The effectiveness of this prediction system varies with the size, type, and Internet exposure of the organization. Hence, similar to the NIDSs themselves, this system needs to be continually updated and retrained as each organization's threat landscape and security practices evolve.

Nevertheless, our work successfully demonstrates the feasibility of this approach. However, as with any new technology, it is necessary to consider the broader implications and potential issues that may arise from its widespread use.

6.3. IMPLICATIONS FOR GOVERNANCE

The results of this research, and the feasibility of the developed prediction system carry with it several practical implications for governance and public policy. A prediction system as described in this thesis requires two elements: 1) extensive external network scans, and 2) unrestricted access to internal network security data (in the form of NIDS alerts and SOC-labeled incidents). Given the scale of data collection necessary, it stands to reason that acceptable use-cases be appropriately and specifically delineated so as to prevent misuse, unintentional or not.

6.3.1. Scanning the Internet

External network scans are already performed on a daily basis by numerous organizations (e.g., Shodan [208], Censys [43], ShadowServer [229]). Such services can be employed by both legitimate and malicious actors for their own end goals. Nevertheless, the data obtained through these means is publicly accessible by design, and there is thus little sense in placing restrictions upon the public to prevent them from accessing such data.

Still, improper use of scanning techniques can result in undesired side-effects on the side of the receiving party. This can range from low-level alert triggers in the SOC to practically a full-blown denial of service attack. While the former can be seen as simply a nuisance, the latter can be legally actionable. Consequently, organizations and their network administrators may prefer network scans not be performed on their infrastructure. To protect both the—presumably benign—scanning organization and the organization being scanned, the scanning must be performed with proper “etiquette”. Several of them were proposed by Durumeric et al. [72], the creators of ZMap.

That is not to say that these best practices would best serve our communities enshrined in legislation and rigorously applied by law enforcement. Nor should extensive Internet scanning tools and techniques become exclusive to governments and intelligence agencies. Private institutions have used these tools and techniques for many years. Through this, they play major roles within the improvement of security on the Internet. Even though official government institutions are ultimately responsible for the take-downs, arrests, and prosecution of cybercriminals, this work is very often done in collaboration with private citizens within, e.g., universities and private organizations. For instance, the aforementioned ShadowServer Foundation [229] was instrumental in the 4 year-long international operation aimed at bringing down the Avalanche cybercrime syndicate and botnet [78].

Existing legislation surrounding the criminality of computer abuse generally does not address the issue of (large-scale) port scanning directly. However, some provisions are relevant in the context of port scans and can be applied, depending on, e.g., the circumstances surrounding the scanning and the objective of the actor performing the scans (e.g., the Netherlands [75], U.S. [127]). And since private citizens are the ones often responsible for discovering security flaws in the networks of companies, they are the ones who often take the brunt of this legislation. An example of this is AT&T’s misconfiguration of their servers that resulted in making the email addresses of iPad owners publicly accessible on the Internet. A researcher by the name of Andrew Auernheimer then independently discovered this misconfiguration, allowing him to scrape over a 100,000 emails from the unsecured servers [74]. By publishing the email addresses, AT&T was forced to fix the server’s security issues. No good deed goes unpunished, however, and Mr. Auernheimer was sentenced to three and a half years in prison for violating the U.S.' *Computer Fraud and Abuse Act* [127], which, among other things, prohibits anybody from gaining unauthorized access to computer systems. Ultimately and thankfully, this conviction was overturned in an appeal, since Auernheimer had not actually circumvented any "code-based" access restrictions [74].

Although the disclosure of the vulnerability in AT&T's servers could have been done more responsibly, or in a less public manner, this is hardly something for which Auernheimer carries the (entire) blame. Responsible vulnerability disclosure is a relatively new development, and, at best, it was still in its infancy back in 2013. In fact, the first ISO standard regarding vulnerability disclosure was published only in February 2014 [106]. It can be argued that AT&T got off easy, and it repaid the security researcher responsible for identifying a major flaw in its network by attempting to put him in prison.

Companies can be overeager to litigate and/or retaliate against security researchers who discover and disclose vulnerabilities in their systems that the companies themselves have lacked the capabilities to discover on their own. Many such cases can be found, and various organizations track their occurrences over the years [18, 69]. Extending these laws, perhaps to include the aforementioned best practices, will undoubtedly have negative repercussions for the security community. Many organizations lack the resources or the capabilities to completely map out their digital assets, and maintain them secure at all times, as the pernicious phenomenon of Shadow IT demonstrates [189]. When expensive asset discovery or penetration testing services cannot be obtained, vulnerable organizations rely on security researchers to inform them of their predicament before more malicious actors can seize that opportunity for themselves.

This is even more crucial nowadays due to the implementation of GDPR [166]. GDPR mandates organizations to safeguard personal data against unauthorized access and data breaches. The contributions of independent researchers can bolster an organization's security posture, ensuring ongoing compliance with GDPR's stringent data protection requirements. Furthermore, their involvement promotes a culture of continuous security improvement, aligning with GDPR's principle of "privacy by design" and helping organizations to mitigate risks effectively, maintain trust with their customers, and avoid the severe penalties associated with non-compliance.

In the context of this thesis, the purpose of the external network scans is to observe and determine as accurately as possible the external state of a system. The only communication performed with third-party computer systems is at most to identify potential misconfigurations, but not to act upon them. Exploiting vulnerabilities is entirely out of scope. Accessing the internal state of organizations' networks is performed through completely different means, and, therefore, requires a different type of reflection.

6.3.2. ACCESS TO INTERNAL SECURITY EVENT DATA

The second requirement of our prediction system is access to an organization's internal security events. In the specific case of this thesis, this data comes in the form of alerts generated by an NIDS installed within an organization's network, in addition to the security incidents that are investigated by the SOC. To create and train a workable and effective prediction system, it requires many years' worth of such data, which we were able to acquire this data through strict legal agreements with the collaborating MSSP. These legal processes can be long and arduous, since the data that is being shared with researchers is of a highly private and sensitive nature.

This can provide an explanation as to why these prediction systems are not more widespread, both in academia and industry. The technical requirements are not prohibitive, even for small and medium-sized enterprises (SMEs). Instead, it is the prior step of acquiring the necessary internal network data that is the biggest obstacle.

At the forefront of this issue is the privacy of the client organizations. Any entity that has access to an organization's sensitive data can discover and reveal trade secrets, business strategies, and personal identifiable information of employees and customers. This data must, therefore, be protected accordingly against unauthorized access and data breaches, as mandated by GDPR and other data protection legislation.

These constraints limit the types of organizations that are currently capable of creating such prediction systems to those that provide security services to a large network of client organizations or departments (e.g. MSSPs, consultancy firms, large banks). Due to the nature of the services such organizations provide, they will have access to their clients' internal network data. The diversity in the population of client organizations and networks will determine the generalizability of the model, although the case can be made for domain-specific models designed for specific branches of industry.

Data marketplaces are recent phenomena that can offer an alternative to organizations that lack a network of client organizations [77]. These platforms enable organizations can choose to monetize their data by putting it up for sale [161]. Though many different types of data can be published, internal network data would require additional effort to guarantee anonymity and filter out any other type of sensitive information. However, the large quantity of required data means that a large quantity of data must be purchased. Assuming that there are enough organizations that opt to sell their internal network data (which is certainly not a given), the amount of funds necessary for the purchasing of enough data may very well be way beyond the means of any SME, again limiting this technology to a select group of organizations.

Storing sensitive data brings with it the responsibility to maintain secure systems and patch vulnerable software. Although GDPR does not explicitly mention patching vulnerable software, it does require organizations to ensure the security of processing "appropriate to the risk" [107]. This can be interpreted as an obligation to keep systems and software up to date and secure against known vulnerabilities. Regularly updating and patching software can be seen as part of the requirement to take steps to secure personal data.

Given the sensitive nature of the data, it is only natural for organizations to be averse to the sharing of this data with other parties. Thus, as with all informed consent, this process must be carried out to ensure all parties are on the same page regarding issues such as data ownership and what risks are involved.

This must be done, then, at the discretion of the organizations that are providing this data. The data can be used to discover weaknesses in the networks of these organizations, as well as provide valuable insights in how to bypass and evade security controls. Therefore, access to internal network data should be limited to trusted personnel. Obviously, this includes security professionals that require access to this data on a daily basis, such as SOC analysts, (N)IDS rule writers, incident response teams, among others. Additional cases, such as for data analytics and (academic) research purposes, should require further deliberation in collaboration with data stewards and, potentially, legal personnel to delineate the scope. Opting out of the data sharing programs, particularly for the latter two cases, should always be maintained as a possibility for the client organizations.

6.3.3. RISK MEASUREMENTS

Many jurisdictions around the world mandate private and governmental entities to disclose data breaches to the public. When this type of prediction system becomes effective enough at predicting the occurrence of singular breaches, should these predictions be disclosed to the public? Or perhaps to a governing body?

We should probably avoid *Minority Report*-esque[1] situations where potential offenders are penalized before an offence has actually taken place. This does not mean, however, that prediction systems are worthless to governance efforts. These systems can be used to compute risk scores based on the probability of a breach occurring within a certain time span. The scores would be empirical measurements extracted from an organization's digital infrastructure.

One could draw parallels between such scores and energy labels given to electronic appliances, vehicles, and buildings. Just as governments can incentivize the development of energy-efficient office buildings by mandating a minimum energy label [164] or offering

[1] *Minority Report* is a 2002 movie based on a 1956 novella of the same name, set in a future where a special police unit can arrest murderers before they commit their crimes, thanks to a technology that can predict the future.

subsidies for additional sustainability and energy efficiency efforts [40], so can they put in place similar mandates and subsidies to incentivize the attainment of better risk scores. Insurance firms using these risk scores to evaluate organization's security posture can further stimulate organizations to improve the state of their security if that means having to spend less money on premiums. Finally, market pressure can also be exerted by consumers and businesses on other organizations that achieve sub-par risk scores.

A potential downside of this policy is that organizations can aim at optimizing the score itself, rather than actually improving security in a productive way. The more a metric is used for social decision-making, the more likely it will be to incentivize gaming and undermine the process it is intended to improve [41]. While this is certainly a valid observation, the same can be said about current security posture measurements that are based on compliance metrics. At least a risk score as proposed is based on empirical measurements.

The exclusive use of external network data to make predictions is a trade-off that is made due to the scale of the problem. Most indicators of low security originate from inside a network. Thus, only with access to the internal network of an organization is it possible to most accurately assess the state of an organization's security. Analyzing all internal network traffic for audits is impractical and unscalable due to the complexity and variability among organizations, posing challenges for generalizability. Using external network data as a proxy will likely result in information loss and yield less accurate results. However, if research indicates that much information about internal security can be inferred from external data, extensive internal analysis becomes unnecessary. Taking a snapshot of an organization's external digital infrastructure is simply easier.

Organizations can benefit from creating in-house prediction systems with their own data for anomaly detection. However, these systems will struggle to adapt to benign network changes due to the model's lack of generalizing knowledge. Risk scores derived from the output of in-house systems may not be as reliable as those from more advanced third-party models. Furthermore, SMEs likely lack the financial resources to develop and maintain such systems. For the aforementioned use cases, it is then seems more sensible for the majority of organizations to rely on more sophisticated models for the generation of risk scores.

Such risk scores are dependent on a more advanced type of prediction system than is developed in this thesis, though. This and other potential or necessary advancements will be discussed next.

6.4. FUTURE WORK

The chapters in this dissertation leave room for future research. In this chapter, we discuss these potential avenues.

In Chapter 2, we discuss the topic of asset discovery. This is a very broad field and, while we were able to collect and systematize many techniques, there was a noticeable lack of discovery techniques for IoT devices, or network services running on non-standard ports. As reconnaissance and asset discovery are crucial in the process of securing an organization, it is important that such techniques are developed and documented.

Chapters 3 and 4 describe the technical and organizational aspects, respectively, that govern the management of NIDS rules within a SOC. A straightforward direction for future research is performing similar investigations at additional SOCs to verify the generalizability of the results. Additionally, we have exclusively scoped our research to security incidents at network level. Including security events at host and application level will allow for a rich comparison between threat landscapes and workflows at different layers, as well as yield a more complete view of the incidents faced by organizations and SOCs.

When SOCs lack an internal threat intelligence department, they depend wholly on outside sources and publicly available data. Future research can investigate how complete reliance on these sources affects workflows within the SOC, as well as, ultimately, the network security of the organizations that the SOC is tasked with protecting. In a more general sense, each SOC has its own unique manner in which they provide security to their clients, depending on the unique blend of resources, expertise, and business model, all seemingly as effective as the rest. Future work can delve into the specific differences between different SOCs and examine how these differences affect aspects such as network security, customer satisfaction, and analyst workload.

Chapter 5 is the most critical part of this dissertation, which also opens the door to the largest amount of future research. The chapter describes the design and performance of an ML-based security incident prediction system. It is able to confidently predict the number of certain types of incidents that will occur within a specific time window. However, the model experiences difficulties when attempting to predict low-risk incidents, and sometimes successful hacks. Future research can examine this discrepancy more deeply, study the specific types of threats that cause difficulties to predict, and identify the reasons why not every incident is created equal. Furthermore, instead of using the data of third-party Internet scanning organizations (e.g. Shodan, Censys, ShadowServer), the training data can be collected in real-time using the asset discovery techniques collected and systematized in Chapter 2. Also, in addition to network-level incidents originating from NIDSs that currently form the basis of the system, security incidents at the host and application level can be included in the training data. This can potentially allow for a more complete and finer-grained picture of an organization's security posture and more accurate predictions. The discovery techniques can also be expanded on with techniques developed and published more recently in industry and academic literature. Finally, the prediction system currently works with correlated signals and says nothing about causation. Additional extensive research is still needed to establish causal links between observed external signals and security events. This step is a necessity, seeing as the goal is to create accurate outcome-based risk metrics.

BIBLIOGRAPHY

[1] 2014. United States of America vs. Ross William Ulbricht. United States District Court, Southern District of New York. Indictment 14CRIM068.

[2] abuseat.org. 2007. The CBL. Retrieved 2023-04-30 from https://web.archive.org/web/20070716052425/http://cbl.abuseat.org/

[3] David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J. Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella-Béguelin, and Paul Zimmermann. 2015. Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)* (22 ed.), Ninghui Li and Christopher Kruegel (Eds.). ACM, 5–17. https://doi.org/10.1145/2810103.2813707

[4] Enoch Agyepong, Yulia Cherdantseva, Philipp Reinecke, and Pete Burnap. 2020. Challenges and performance metrics for security operations center analysts: a systematic review. *Journal of Cyber Security Technology* 4, 3 (2020), 125–152.

[5] Atif Ahmad, Sean B Maynard, Kevin C Desouza, James Kotsias, Monica T Whitty, and Richard L Baskerville. 2021. How can organizations develop situation awareness for incident response: A case study of management practice. *Computers & Security* 101 (2021), 102122.

[6] Rami Al-Dalky, Michael Rabinovich, and Kyle Schomp. 2019. A Look at the ECS Behavior of DNS Resolvers. In *Proceedings of the 2019 Internet Measurement Conference (IMC)*, Anna Sperotto, Roland van Rijswijk-Deij, and Cristian Hesselman (Eds.). ACM, 116–129. https://doi.org/10.1145/3355369.3355586

[7] Rami Al-Dalky and Kyle Schomp. 2018. Characterization of Collaborative Resolution in Recursive DNS Resolvers. In *Proceedings of the 13th Passive and Active Measurement (PAM)* (13 ed.) *(Lecture Notes in Computer Science)*, Robert Beverly and Georgios Smaragdakis (Eds.), Vol. 10771. Springer, 146–157. https://doi.org/10.1007/978-3-319-76481-8_11

[8] Bushra A. Alahmadi, Louise Axon, and Ivan Martinovic. 2022. 99% False Positives: A Qualitative Study of SOC Analysts' Perspectives on Security Alarms. In *Proceedings of the 31st USENIX Security Symposium (USENIX Security)*. USENIX Association, Boston, MA, 2783–2800. https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi

[9] Areej Albataineh and Izzat Alsmadi. 2019. IoT and the Risk of Internet Exposure: Risk Assessment Using Shodan Queries. In *Proceedings of the 20th IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)* (20 ed.). IEEE, 1–5. https://doi.org/10.1109/WoWMoM.2019.8792986

[10] Eugene Albin and Neil C Rowe. 2012. A realistic experimental comparison of the Suricata and Snort intrusion-detection systems. In *2012 26th International Conference on Advanced Information Networking and Applications Workshops (AINAW)*. IEEE, 122–127.

[11] Alliantist Ltd. 2018. ISO 27001 Annex A.8 - Asset Management. Retrieved 2020-06-10 from https://www.isms.online/iso-27001/annex-a-8-asset-management/

[12] Omar Alrawi, Chaoshun Zuo, Ruian Duan, Ranjita Pai Kasturi, Zhiqiang Lin, and Brendan Saltaformaggio. 2019. The Betrayal At Cloud City: An Empirical Analysis Of Cloud-Based Mobile Backends. In *Proceedings of the 28th USENIX Security Symposium (USENIX Security)* (28 ed.), Nadia Heninger and Patrick Traynor (Eds.). USENIX Association, 551–566. https://www.usenix.org/conference/usenixsecurity19/presentation/alrawi

[13] S. Alrwais, X. Liao, X. Mi, P. Wang, X. Wang, F. Qian, R. Beyah, and D. McCoy. 2017. Under the Shadow of Sunshine: Understanding and Detecting Bulletproof Hosting on Legitimate Service Provider Networks. In *Proceedings of the 38th IEEE Symposium on Security & Privacy (S&P)* (38 ed.), Ulfar Erlingsson and Bryan Parno (Eds.). IEEE. https://doi.org/10.1109/SP.2017.32

[14] CAIDA: Center for Applied Internet Data Analysis. [n. d.]. Routeviews Prefix to AS Mappings Dataset (Pfx2as) for IPv4 and IPv6. Retrieved 2020-06-23 from https://www.caida.org/data/routing/routeviews-prefix2as.xml

[15] APEWS.ORG. 2021. APEWS.ORG - Anonymous Postmasters Early Warning System. Retrieved 2023-04-30 from http://www.apews.org/

[16] Hafizul Asad and Ilir Gashi. 2018. Diversity in Open Source Intrusion Detection Systems. In *Proceedings of the 37th International Conference on Computer Safety, Reliability, and Security (SAFECOMP)*, Barbara Gallina, Amund Skavhaug, and Friedemann Bitsch (Eds.). Springer International Publishing, Cham, 267–281.

[17] ASTRA IT, Inc. 2023. 160 Cybersecurity Statistics: Updated Report 2023. Retrieved 2023-08-18 from https://www.getastra.com/blog/security-audit/small-business-cyber-attack-statistics/

[18] attrition.org. 2021. Errata - Legal Threats. https://attrition.org/errata/legal_threats/

[19] Nimrod Aviram, Sebastian Schinzel, Juraj Somorovsky, Nadia Heninger, Maik Dankel, Jens Steube, Luke Valenta, David Adrian, J. Alex Halderman, Viktor Dukhovni, Emilia Käsper, Shaanan Cohney, Susanne Engels, Christof Paar, and Yuval Shavitt. 2016. {DROWN}: Breaking {TLS} Using SSLv2. In *Proceedings of the 25th USENIX Security Symposium (USENIX Security)* (25 ed.), Thorsten Holz and Stefan Savage (Eds.). USENIX Association, 689–706. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/aviram

[20] Mohammed Bashir and Nicolas Christin. 2008. Three case studies in quantitative information risk analysis. In *Proceedings of the 2008 CERT/SEI Making the Business Case for Software Assurance Workshop*. Pittsburgh, PA, 77–86. https://www.andrew.cmu.edu/user/nicolasc/publications/ash.pdf

[21] Ali Sercan Basyurt, Jennifer Fromm, Philipp Kuehn, Marc-André Kaufhold, and Milad Mirbabaie. 2022. Help Wanted-Challenges in Data Collection, Analysis and Communication of Cyber Threats in Security Operation Centers. (2022).

[22] Karyn Benson, Alberto Dainotti, kc claffy, Alex C. Snoeren, and Michael Kallitsis. 2015. Leveraging Internet Background Radiation for Opportunistic Network Analysis. In *Proceedings of the 2015 Internet Measurement Conference (IMC)*, Kenjiro Cho, Kensuke Fukuda, Vivek S. Pai, and Neil Spring (Eds.). ACM, 423–436. https://doi.org/10.1145/2815675.2815702

[23] Robert Beverly and Arthur Berger. 2015. Server Siblings: Identifying Shared IPv4/IPv6 Infrastructure Via Active Fingerprinting. In *Proceedings of the 10th Passive and Active Measurement (PAM) (Lecture Notes in Computer Science)*, Jelena Mirkovic and Yong Liu (Eds.), Vol. 8995. Springer, 149–161. https://doi.org/10.1007/978-3-319-15509-8_12

[24] Robert Beverly, Ramakrishnan Durairajan, David Plonka, and Justin P Rohrer. 2018. In the IP of the Beholder: Strategies for Active IPv6 Topology Discovery. In *Proceedings of the 2018 Internet Measurement Conference (IMC)*, Ben Y. Zhao and Ethan Katz-Bassett (Eds.). ACM, 308—321. https://doi.org/10.1145/3278532.3278559

[25] 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 26th ACM SIGSAC Conference on Computer and Communications Security (CCS)* (26 ed.), Lorenzo Carvalho, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz (Eds.). ACM, 449–464. https://doi.org/10.1145/3319535.3354230

[26] Leyla Bilge, Yufei Han, and Matteo Dell'Amico. 2017. RiskTeller: Predicting the Risk of Cyber Incidents. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*. Association for Computing Machinery, New York, NY, USA, 1299–1311. https://doi.org/10.1145/3133956.3134022

[27] Henry Birge-Lee, Yixin Sun, Anne Edmundson, Jennifer Rexford, and Prateek Mittal. 2018. Bamboozling Certificate Authorities with {BGP}. In *Proceedings of the 27th USENIX Security Symposium (USENIX Security)* (27 ed.), William Enck and Adrienne Porter Felt (Eds.). USENIX Association, 833–849. https://www.usenix.org/conference/usenixsecurity18/presentation/birge-lee

[28] Alex Biryukov, Ivan Pustogarov, and Ralf-Philipp Weinmann. 2013. Trawling for Tor hidden services: Detection, measurement, deanonymization. In *Proceedings of the 34th IEEE Symposium on Security & Privacy (S&P)* (34 ed.). IEEE, 80–94. https://doi.org/10.1109/SP.2013.15

[29] BitSight. [n. d.]. BitSight: Security Ratings Leader - Cyber Risk Management Solutions. Retrieved 2023-06-30 from https://www.bitsight.com/

[30] Martin Bland. 2015. *An Introduction to Medical Statistics* (fourth ed.). Oxford University Press, Chapter Missing data, 306.

[31] Leon Böck, Emmanouil Vasilomanolakis, Max Mühlhäuser, and Shankar Karuppayah. 2019. Next Generation P2P Botnets: Monitoring Under Adverse Conditions. In *Proceedings of the 21st International Symposium on Recent Advances in Intrusion Detection (RAID)* (21 ed.) *(Lecture Notes in Computer Science)*, Michael Bailey, Thorsten Holz, Manolis Stamatogiannakis, and Sotiris Ioannidis (Eds.), Vol. 11050. Springer, 511–531. https://doi.org/10.1007/978-3-030-00470-5_24

[32] Roland Bodenheim, Jonathan Butts, Stephen Dunlap, and Barry Mullins. 2014. *International Journal of Critical Infrastructure Protection* 7, 2 (June 2014), 114–123. https://doi.org/10.1016/j.ijcip.2014.03.001

[33] Anthony J. Bonkoski, Russ Bielawski, and J. Alex Halderman. 2013. Illuminating the Security Issues Surrounding Lights-Out Server Management. In *7th USENIX Workshop on Offensive Technologies*, Jon Oberheide and William K. Robertson (Eds.). USENIX Association. https://www.usenix.org/conference/woot13

[34] Kevin Borgolte, Tobias Fiebig, Shuang Hao, Christopher Kruegel, and Giovanni Vigna. 2018. Cloud Strife: Mitigating the Security Risks of Domain-Validated Certificates. In *Proceedings of the 25th Network and Distributed System Security Symposium (NDSS)* (25 ed.), Patrick Traynor and Alina Oprea (Eds.). Internet Society (ISOC).

[35] Kevin Borgolte, Shuang Hao, Tobias Fiebig, and Giovanni Vigna. 2018. Enumerating Active IPv6 Hosts for Large-Scale Security Scans via DNSSEC-Signed Reverse Zones. In *Proceedings of the 39th IEEE Symposium on Security & Privacy (S&P)* (39 ed.), Bryan Parno and Christopher Kruegel (Eds.). IEEE, 770–784. https://doi.org/10.1109/SP.2018.00027

[36] S. Bortzmeyer. 2016. *DNS Query Name Minimisation to Improve Privacy*. RFC 7816. RFC Editor. https://doi.org/10.17487/RFC7816

[37] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[38] Virginia Braun and Victoria Clarke. 2021. One size fits all? What counts as quality practice in (reflexive) thematic analysis? *Qualitative research in psychology* 18, 3 (2021), 328–352.

[39] Bricata. 2021. IDS is Dead! Long Live IDS! An Analyst Prediction from 2003 Remains Relevant. Retrieved 2021-05-07 from https://bricata.com/blog/ids-is-dead/

[40] Business.gov.nl. 2024. Subsidies | Business.gov.nl. https://business.gov.nl/running-your-business/environmental-impact/subsidies/

[41] Donald T. Campbell. 1979. Assessing the impact of planned social change. *Evaluation and Program Planning* 2, 1 (1979), 67–90. https://doi.org/10.1016/0149-7189(79)90048-X

[42] Frank Cangialosi, Taejoong Chung, David Choffnes, Dave Levin, Bruce M. Maggs, Alan Mislove, and Christo Wilson. 2016. Measurement and Analysis of Private Key Sharing in the HTTPS Ecosystem. In *Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security (CCS)* (23 ed.), Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (Eds.). ACM, 628–640. https://doi.org/10.1145/2976749.2978301

[43] Censys. [n. d.]. Censys. Retrieved 2020-06-02 from http://censys.io/

[44] Center for Internet Security. [n. d.]. CIS Critical Security Controls. Retrieved 2023-06-30 from https://www.cisecurity.org/controls

[45] Tianfeng Chai and Roland R Draxler. 2014. Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geoscientific model development* 7, 3 (2014), 1247–1250.

[46] Check Point Software Technologies Ltd. 2023. Check Point Research Reports a 38Global Cyberattacks - Check Point Blog. Retrieved 2023-08-18 from https://blog.checkpoint.com/2023/01/05/38-increase-in-2022-global-cyberattacks/

[47] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 785–794.

[48] Daiki Chiba, Ayako Akiyama Hasegawa, Takashi Koide, Yuta Sawabe, Shigeki Goto, and Mitsuaki Akiyama. 2019. DomainScouter: Understanding the Risks of Deceptive IDNs. In *Proceedings of the 22nd International Symposium on Recent Advances in Intrusion Detection (RAID)* (22 ed.), Thorsten Holz and Manuel Egele (Eds.). USENIX Association, 413–426. https://www.usenix.org/conference/raid2019/presentation/chiba

[49] D. Chiba, T. Yagi, M. Akiyama, T. Shibahara, T. Yada, T. Mori, and S. Goto. 2016. DomainProfiler: Discovering Domain Names Abused in Future. In *Proceedings of the 46th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Domenico Cotroneo and Cristina Nita-Rotaru (Eds.). IEEE, 491–502. https://doi.org/10.1109/DSN.2016.51

[50] Nicolas Christin. 2013. Traveling the Silk Road: A measurement analysis of a large anonymous online marketplace. In *Proceedings of the 22nd World Wide Web*

Conference (WWW) (22 ed.), Daniel Schwabe, Virgílio A. F. Almeida, Hartmut Glaser, Ricardo Baeza-Yates, and Sue B. Moon (Eds.). 213–224. https://doi.org/10.1145/2488388.2488408

[51] Taejoong Chung, David Choffnes, and Alan Mislove. 2016. Tunneling for Transparency: A Large-Scale Analysis of End-to-End Violations in the Internet. In *Proceedings of the 2016 Internet Measurement Conference (IMC)*, Phillipa Gill, John S. Heidemann, John W. Byers, and Ramesh Govindan (Eds.). ACM, 199–213. https://doi.org/10.1145/2987443.2987455

[52] Taejoong Chung, Yabing Liu, David Choffnes, Dave Levin, Bruce MacDowell Maggs, Alan Mislove, and Christo Wilson. 2016. Measuring and Applying Invalid SSL Certificates: The Silent Majority. In *Proceedings of the 2016 Internet Measurement Conference (IMC '16)*. ACM, Santa Monica, CA, USA, 527–541. https://doi.org/10.1145/2987443.2987454

[53] Cisco. 2021. Snort - Network Intrusion Detection & Prevention System. Retrieved 2021-04-16 from https://www.snort.org/

[54] Cisco. 2021. Talos - Author of the Official Snort Rule Sets. Retrieved 2021-03-24 from https://www.snort.org/talos

[55] Cisco. 2021. Why are rules commented out by default? Retrieved 2021-04-16 from https://www.snort.org/faq/why-are-rules-commented-out-by-default

[56] Cisco Talos Intelligence Group. 2023. PhishTank | Join the fight against phishing. Retrieved 2023-04-30 from https://phishtank.org/

[57] European Commission. 2021. Four eyes principle | CROS. https://ec.europa.eu/eurostat/cros/content/four-eyes-principle_en

[58] A. Continella, M. Polino, M. Pogliani, and S. Zanero. 2018. There's a Hole in that Bucket! A Large-scale Analysis of Misconfigured S3 Buckets. In *Proceedings of the 34th Annual Computer Security Applications Conference (ACSAC)* (34 ed.). ACM.

[59] CrowdStrike. 2023. What is a Purple Team? – CrowdStrike. https://www.crowdstrike.com/cybersecurity-101/purple-teaming/

[60] Chris Crowley and John Pescatore. 2019. Common and best practices for security operations centers: Results of the 2019 SOC survey. *SANS, Bethesda, MD, USA, Tech. Rep* (2019).

[61] CVE. 2023. Metrics | CVE. https://www.cve.org/About/Metrics

[62] Jakub Czyz, Matthew Luckie, Mark Allman, and Michael Bailey. 2016. Don't Forget to Lock the Back Door! A Characterization of IPv6 Network Security Policy. In *Proceedings of the 23rd Network and Distributed System Security Symposium (NDSS)* (23 ed.), Srdjan Capkun (Ed.). Internet Society (ISOC). https://doi.org/10.14722/ndss.2016.23047

[63] David Day and Benjamin Burns. 2011. A performance analysis of Snort and Suricata network intrusion detection and prevention engines. In *Proceedings of the 5th International Conference on Digital Society (ICDS)*. 187–192.

[64] Wouter B de Vries, Roland van Rijswijk-Deij, Pieter-Tjerk de Boer, and Aiko Pras. 2018. Passive Observations of a Large DNS Service: 2.5 Years in the Life of Google. In *Proceedings of the 2018 International Workshop on Traffic Monitoring and Analysis (TMA)*, Pedro Casas, Nur Zincir-Heywood, and Amogh Dhamdhere (Eds.). IEEE, 190–200. https://doi.org/10.1109/TNSM.2019.2936031

[65] Matteo Dell'Amico, Leyla Bilge, Ashwin Kayyoor, Petros Efstathopoulos, and Pierre-Antoine Vervier. 2017. Lean On Me: Mining Internet Service Dependencies From Large-Scale DNS Data. In *Proceedings of the 33rd Annual Computer Security Applications Conference (ACSAC)* (33 ed.). ACM, 449–460. https://doi.org/10.1145/3134600.3134637

[66] Amogh Dhamdhere, David D. Clark, Alexander Gamero-Garrido, Matthew Luckie, Ricky K. P. Mok, Gautam Akiwate, Kabir Gogia, Vaibhav Bajpai, Alex C. Snoeren, and Kc Claffy. 2018. Inferring Persistent Interdomain Congestion. In *Proceedings of the 2018 ACM SIGCOMM Conference (SIGCOMM)*, Sergey Gorinsky and János Tapolcai (Eds.). ACM, 1–15. https://doi.org/10.1145/3230543.3230549

[67] Constanze Dietrich, Katharina Krombholz, Kevin Borgolte, and Tobias Fiebig. 2018. Investigating System Operators' Perspective on Security Misconfigurations. In *Proceedings of the 25th ACM SIGSAC Conference on Computer and Communications Security (CCS)* (25 ed.), Michael Backes and XiaoFeng Wang (Eds.). ACM. https://doi.org/10.1145/3243734.3243794

[68] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2014. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium (USENIX Security)* (23 ed.), Matt Blaze (Ed.). USENIX Association. https://doi.org/10.5555/1251375.1251396

[69] disclose.io. 2023. Research Threats: Legal Threats Against Security Researchers | Security Research Threats. https://attrition.org/errata/legal_threats/

[70] Zakir Durumeric, David Adrian, Ariana Mirian, Michael 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)* (22 ed.), Ninghui Li and Christopher Kruegel (Eds.). ACM, 542–553. https://doi.org/10.1145/2810103.2813703

[71] Zakir Durumeric, J. Alex Halderman, David Adrian, Ariana Mirian, James Kasten, Elie Bursztein, Nicolas Lidzborski, Kurt Thomas, Vijay Eranti, and Michael Bailey. 2015. Neither Snow Nor Rain Nor MITM...: An Empirical Analysis of Email Delivery Security. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)* (22 ed.), Ninghui Li and Christopher Kruegel (Eds.). ACM, 27–39. https://doi.org/10.1145/2815675.2815695

[72] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. 2013. ZMap: Fast Internet-Wide Scanning and Its Security Applications ZMap: Fast Internet-Wide Scanning and Its Security Applications. In *Proceedings of the 22nd USENIX Security Symposium (USENIX Security)* (22 ed.), Samuel T. King (Ed.). USENIX Association. https://doi.org/10.5555/2534766.2534818

[73] Benjamin Edwards, Jay Jacobs, and Stephanie Forrest. 2019. Risky Business: Assessing Security with External Measurements. arXiv:cs.CR/1904.11052

[74] Electronic Frontier Foundation. 2014. Appeals Court Overturns Andrew “weev” Auernheimer Conviction. https://www.eff.org/press/releases/appeals-court-overturns-andrew-weev-auernheimer-conviction

[75] Arnoud Engelfriet. 2018. De portscan als strafbare voorbereiding @ iusmentis.com door Arnoud Engelfriet. https://www.iusmentis.com/beveiliging/hacken/portscans/

[76] European Commission. 2017. Ecodesign and Energy Label - European Commission. https://energy-efficient-products.ec.europa.eu/ecodesign-and-energy-label_en

[77] Joint Research Centre European Commission. 2024. Data marketplace | Joinup. https://joinup.ec.europa.eu/collection/elise-european-location-interoperability-solutions-e-government/glossary/term/data-marketplace

[78] Europol. [n. d.]. ’Avalanche’ network dismantled in international cyber operation | Europol. Retrieved 2024-01-09 from https://www.europol.europa.eu/media-press/newsroom/news/%e2%80%98avalanche%e2%80%99-network-dismantled-in-international-cyber-operation

[79] FICO. [n. d.]. The #1 Analytic Decisioning Platform to Optimize Consumer Interactions Across all Customer Decisions | FICO. Retrieved 2023-06-30 from https://www.fico.com/

[80] Tobias Fiebig, Kevin Borgolte, Shuang Hao, Christopher Kruegel, and Giovanni Vigna. 2017. Something from Nothing (There): Collecting Global IPv6 Datasets from DNS. In *Proceedings of the 12th Passive and Active Measurement (PAM)* (12 ed.) *(Lecture Notes in Computer Science)*, Mohamed Ali Kaafar, Steve Uhlig, and Johanna Amann (Eds.), Vol. 10176. Springer, 30–43. https://doi.org/10.1007/978-3-319-54328-4_3

[81] Tobias Fiebig, Kevin Borgolte, Shuang Hao, Christopher Kruegel, Giovanni Vigna, and Anja Feldmann. 2018. In rDNS We Trust: Revisiting a Common Data-Source’s Reliability. In *Proceedings of the 13th Passive and Active Measurement (PAM)* (13 ed.) *(Lecture Notes in Computer Science)*, Robert Beverly, Georgios Smaragdakis, and Anja Feldmann (Eds.), Vol. 10771. Springer, 131–145. https://doi.org/10.1007/978-3-319-76481-8_10

[82] Tobias Fiebig, Anja Feldmann, and Matthias Petschick. 2016. A one-year perspective on exposed in-memory key-value stores. In *Proceedings of the 2016 ACM Workshop on Automated Decision Making for Active Cyber Defense*. 17–22.

[83] Jason Firch. 2021. 2021 Cyber Security Statistics: The Ultimate List Of Stats, Data & Trends. Retrieved 2021-04-16 from https://purplesec.us/resources/cyber-security-statistics

[84] Pawel Foremski, Oliver Gasser, and Giovane C. M. Moura. 2019. DNS Observatory: The Big Picture of the DNS. In *Proceedings of the 2019 Internet Measurement Conference (IMC)*, Anna Sperotto, Roland van Rijswijk-Deij, and Cristian Hesselman (Eds.). ACM, 87–100. https://doi.org/10.1145/3355369.3355566

[85] Pawel Foremski, David Plonka, and Arthur Berger. 2016. Entropy/IP: Uncovering Structure in IPv6 Addresses. In *Proceedings of the 2016 Internet Measurement Conference (IMC)*, Phillipa Gill, John S. Heidemann, John W. Byers, and Ramesh Govindan (Eds.). ACM, 167–181. https://doi.org/10.1145/2987443.2987445

[86] Ian D. Foster, Jon Larson, Max Masich, Alex C. Snoeren, Stefan Savage, and Kirill Levchenko. 2015. Security by Any Other Name: On the Effectiveness of Provider Based Email Security. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)* (22 ed.), Ninghui Li and Christopher Kruegel (Eds.). ACM, 450–464. https://doi.org/10.1145/2810103.2813607

[87] Gartner, Inc. [n. d.]. Gartner Identifies Three Factors Influencing Growth in Security Spending. Retrieved 2023-06-30 from https://www.gartner.com/en/newsroom/press-releases/2022-10-13-gartner-identifies-three-factors-influencing-growth-i

[88] Oliver Gasser, Benjamin Hof, Max Helm, Maciej Korczynski, Ralph Holz, and Georg Carle. 2018. In Log We Trust: Revealing Poor Security Practices with Certificate Transparency Logs and Internet Measurements. In *Proceedings of the 13th Passive and Active Measurement (PAM)* (13 ed.) *(Lecture Notes in Computer Science)*, Robert Beverly and Georgios Smaragdakis (Eds.), Vol. 10771. Springer, 173–185. https://doi.org/10.1007/978-3-319-76481-8_13

[89] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczyński, Stephen D. Strowes, Luuk Hendriks, and Georg Carle. 2018. Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists. In *Proceedings of the 2018 Internet Measurement Conference (IMC)*, Ben Y. Zhao and Ethan Katz-Bassett (Eds.). ACM, 364–378. https://doi.org/10.1145/3278532.3278564

[90] Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle. 2016. Scanning the IPv6 Internet: Towards a Comprehensive Hitlist. In *Proceedings of the 2016 International Workshop on Traffic Monitoring and Analysis (TMA)*, Ramin Sadre, Fabiàn Bustamante, and Alessio Botta (Eds.). IFIP. http://dl.ifip.org/db/conf/tma/tma2016/tma2016-final51.pdf

[91] Manaf Gharaibeh, Anant Shah, Bradley Huffaker, Han Zhang, Roya Ensafi, and Christos Papadopoulos. 2017. A Look at Router Geolocation in Public and Commercial Databases. In *Proceedings of the 2017 Internet Measurement Conference (IMC)*, Steve Uhlig and Olaf Maennel (Eds.). ACM, 463–469. https://doi.org/10.1145/3131365.3131380

[92] Jean-François Grailet and Benoit Donnet. 2019. Revisiting Subnet Inference WISE-Ly. Retrieved 2019-06-28 from https://github.com/JefGrailet/WISE

[93] Jean-Franois Grailet, Fabien Tarissan, and Benoit Donnet. 2016. Passive Observations of a Large DNS Service: 2.5 Years in the Life of Google. In *Proceedings of the 2016 International Workshop on Traffic Monitoring and Analysis (TMA)*, Ramin Sadre, Fabiàn Bustamante, and Alessio Botta (Eds.). IFIP, 190–200. http://dl.ifip.org/db/conf/tma/tma2016/tma2016-final12.pdf

[94] Nitika Gupta, Issa Traore, and Paulo Magella Faria de Quinan. 2019. Automated Event Prioritization for Security Operation Center using Deep Learning. In *2019 IEEE International Conference on Big Data (Big Data)*. 5864–5872. https://doi.org/10.1109/BigData47090.2019.9006073

[95] Shuai Hao, Yubao Zhang, Haining Wang, and Angelos Stavrou. 2018. End-Users Get Maneuvered: Empirical Analysis of Redirection Hijacking in Content Delivery Networks. In *Proceedings of the 27th USENIX Security Symposium (USENIX Security)* (27 ed.), William Enck and Adrienne Porter Felt (Eds.). USENIX Association, 551–566. https://www.usenix.org/conference/usenixsecurity18/presentation/hao

[96] D. Harkins and D. Carrel. 1998. *The Internet Key Exchange (IKE).* RFC 2409. RFC Editor. https://doi.org/10.17487/rfc2409

[97] Luuk Hendriks, Ricardo de Oliveira Schmidt, Roland van Rijswijk-Deij, and Aiko Pras. 2017. On the Potential of IPv6 Open Resolvers for DDoS Attacks. In *Proceedings of the 12th Passive and Active Measurement (PAM)* (12 ed.) *(Lecture Notes in Computer Science)*, Mohamed Ali Kâafar, Steve Uhlig, and Johanna Amann (Eds.), Vol. 10176. Springer, 17–29. https://doi.org/10.1007/978-3-319-54328-4_2

[98] T. Hlavacek, A. Herzberg, H. Shulman, and M. Waidner. 2018. Practical Experience: Methodologies for Measuring Route Origin Validation. In *Proceedings of the 48th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Marco Vieira and Gilles Muller (Eds.). IEEE, 634–641. https://doi.org/10.1109/DSN.2018.00070

[99] Timothy O Hodson. 2022. Root-mean-square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development* 15, 14 (2022), 5481–5487.

[100] Hang Hu and Gang Wang. 2018. End-to-End Measurements of Email Spoofing Attacks. In *Proceedings of the 27th USENIX Security Symposium (USENIX Security)* (27

ed.), William Enck and Adrienne Porter Felt (Eds.). USENIX Association, 1095–1112. https://www.usenix.org/conference/usenixsecurity18/presentation/hu

[101] Qinwen Hu, Muhammad Rizwan Asghar, and Nevil Brownlee. 2018. Measuring IPv6 DNS Reconnaissance Attacks and Preventing Them Using DNS Guard. In *Proceedings of the 48th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Marco Vieira and Gilles Muller (Eds.). IEEE, 350–361. https://doi.org/10.1109/DSN.2018.00045

[102] Huawei. [n. d.]. A Brief Introduction about New IP Research Initiative. Retrieved 2021-02-18 from https://www.huawei.com/en/industry-insights/innovation/new-ip

[103] Anwar Husain, Ahmed Salem, Carol Jim, and George Dimitoglou. 2019. Development of an Efficient Network Intrusion Detection Model Using Extreme Gradient Boosting (XGBoost) on the UNSW-NB15 Dataset. In *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. 1–7. https://doi.org/10.1109/ISSPIT47144.2019.9001867

[104] IBM Security. 2023. Cost of a Data Breach Report 2022. Retrieved 2023-08-18 from https://www.ibm.com/downloads/cas/3R8N1DZJ

[105] M. Inci, G. Irazoqui, T. Eisenbarth, and B. Sunar. 2016. Efficient, Adversarial Neighbor Discovery using Logical Channels on Microsoft Azure. In *Proceedings of the 32nd Annual Computer Security Applications Conference (ACSAC)* (32 ed.). ACM.

[106] International Organization for Standardization. 2014. ISO/IEC 29147:2014 - Information technology — Security techniques — Vulnerability disclosure. https://www.iso.org/standard/45170.html

[107] intersoft consulting. 2024. Art. 32 GDPR – Security of processing - General Data Protection Regulation (GDPR). https://gdpr-info.eu/art-32-gdpr/

[108] ISACA. [n. d.]. COBIT | Control Objectives for Information Technologies | ISACA. Retrieved 2023-06-30 from https://www.isaca.org/resources/cobit

[109] ISO. [n. d.]. ISO/IEC 27001 Standard – Information Security Management Systems. Retrieved 2023-06-30 from https://www.iso.org/standard/27001

[110] Arthur S. Jacobs, Roman Beltiukov, Walter Willinger, Ronaldo A. Ferreira, Arpit Gupta, and Lisandro Z. Granville. 2022. AI/ML for Network Security: The Emperor Has No Clothes. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22)*. Association for Computing Machinery, New York, NY, USA, 1537–1551. https://doi.org/10.1145/3548606.3560609

[111] Jim Roskind. 2012. QUIC: Design Document and Specification Rationale. Retrieved 2020-06-02 from https://docs.google.com/document/d/1RNHkx_VvKWyWg6Lr8SZ-saqsQx7rFV-ev2jRFUoVD34/

[112] L. Jin, S. Hao, H. Wang, and C. Cotton. 2018. Your Remnant Tells Secret: Residual Resolution in DDoS Protection Services. In *Proceedings of the 48th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Marco Vieira and Gilles Muller (Eds.). IEEE, 362–373. https://doi.org/10.1109/DSN.2018.00046

[113] 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 (IMC)*, Steve Uhlig and Olaf Maennel (Eds.). ACM, 100–113. https://doi.org/10.1145/3131365.3131383

[114] Mattijs Jonker, Anna Sperotto, Roland van Rijswijk-Deij, Ramin Sadre, and Aiko Pras. 2016. Measuring the Adoption of DDoS Protection Services. In *Proceedings of the 2016 Internet Measurement Conference (IMC)*, Phillipa Gill, John S. Heidemann, John W. Byers, and Ramesh Govindan (Eds.). ACM, 279–285. https://doi.org/10.1145/2987443.2987487

[115] Andrew J. Kaizer and Minaxi Gupta. 2015. ∼Open Resolvers: Understanding the Origins of Anomalous Open DNS Resolvers. In *Proceedings of the 10th Passive and Active Measurement (PAM) (Lecture Notes in Computer Science)*, Jelena Mirkovic and Yong Liu (Eds.), Vol. 8995. Springer, 3–14. https://doi.org/10.1007/978-3-319-15509-8_1

[116] Kaufman, C. 2005. *Internet Key Exchange (IKEv2) Protocol*. RFC 4306. RFC Editor. https://doi.org/10.17487/RFC4306

[117] Erin Kenneally and David Dittrich. 2012. The Menlo Report: Ethical principles guiding information and communication technology research. *Available at SSRN 2445102* (2012).

[118] Rachael King. 2016. Cybersecurity Startup QuadMetrics Calculates Odds a Company Will be Breached. Retrieved 2020-05-01 from https://www.wsj.com/articles/BL-CIOB-8864

[119] Amit Klein, Haya Shulman, and Michael Waidner. 2017. Counting in the Dark: DNS Caches Discovery and Enumeration in the Internet. In *Proceedings of the 47th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Evgenia Smirni and Pascal Felber (Eds.). IEEE, 367–378. https://doi.org/10.1109/DSN.2017.63

[120] Faris Bugra Kokulu, Ananta Soneji, Tiffany Bao, Yan Shoshitaishvili, Ziming Zhao, Adam Doupé, and Gail-Joon Ahn. 2019. Matched and mismatched socs: A qualitative study on security operations center issues. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)*. 1955–1970.

[121] Platon Kotzias, Leyla Bilge, Pierre-Antoine Vervier, and Juan Caballero. 2019. Mind Your Own Business: A Longitudinal Study of Threats and Vulnerabilities in Enterprises.. In *Network and Distributed System Security Symposium (NDSS)*.

[122] Athanasios Kountouras, Panagiotis Kintis, Chaz Lever, Yizheng Chen, Yacin Nadji, David Dagon, Manos Antonakakis, and Rodney Joffe. 2016. Enabling Network Security Through Active DNS Datasets. In *Proceedings of the 19th International Symposium on Recent Advances in Intrusion Detection (RAID)* (19 ed.) *(Lecture Notes in Computer Science)*, Fabian Monrose, Marc Dacier, Gregory Blanc, and Joaquin Garcia-Alfaro (Eds.), Vol. 9854. Springer, 188–208. https://doi.org/10.1007/978-3-319-45719-2_9

[123] Thomas Krenc and Anja Feldmann. 2016. BGP Prefix Delegations: A Deep Dive. In *Proceedings of the 2016 Internet Measurement Conference (IMC)*, Phillipa Gill, John S. Heidemann, John W. Byers, and Ramesh Govindan (Eds.). ACM, 469–475. https://doi.org/10.1145/2987443.2987458

[124] Christopher Kruegel, Davide Balzarotti, William Robertson, and Giovanni Vigna. 2007. Improving signature testing through dynamic data flow analysis. In *Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC)*. IEEE, 53–63. https://doi.org/10.1109/ACSAC13565.2007

[125] D. Kumar, Z. Wang, M. Hyder, J. Dickinson, G. Beck, D. Adrian, J. Mason, Z. Durumeric, J. A. Halderman, and M. Bailey. 2018. Tracking Certificate Misissuance in the Wild. In *Proceedings of the 39th IEEE Symposium on Security & Privacy (S&P)*

(39 ed.), Bryan Parno and Christopher Kruegel (Eds.). IEEE. https://doi.org/10.1109/SP.2018.00015

[126] Albert Kwon, Mashael AlSabah, David Lazar, Marc Dacier, and Srinivas Devadas. 2015. Circuit Fingerprinting Attacks: Passive Deanonymization of Tor Hidden Services. In *Proceedings of the 24th USENIX Security Symposium (USENIX Security)* (24 ed.), Jaeyeon Jung Jung and Thorsten Holz (Eds.). USENIX Association, 287–302.

[127] Legal Information Institute. 2024. 18 U.S. Code §1030 - Fraud and related activity in connection with computers | U.S. Code | US Law | LII / Legal Information Institute. https://www.law.cornell.edu/uscode/text/18/1030

[128] Frank Li, Zakir Durumeric, Jakub Czyz, Mohammad Karami, Michael Bailey, Damon McCoy, Stefan Savage, and Vern Paxson. 2016. You've Got Vulnerability: Exploring Effective Vulnerability Notifications. In *Proceedings of the 25th USENIX Security Symposium (USENIX Security)* (25 ed.), Thorsten Holz and Stefan Savage (Eds.). USENIX Association, 1033–1050. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/li

[129] B. Liu, C. Lu, Z. Li, Y. Liu, H. Duan, S. Hao, and Z. Zhang. 2018. A Reexamination of Internationalized Domain Names: The Good, the Bad and the Ugly. In *Proceedings of the 48th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Marco Vieira and Gilles Muller (Eds.). IEEE, 654–665. https://doi.org/10.1109/DSN.2018.00072

[130] Daiping Liu, Shuai Hao, and Haining Wang. 2016. All Your DNS Records Point to Us: Understanding the Security Threats of Dangling DNS Records. In *Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security (CCS)* (23 ed.), Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (Eds.). ACM, 1414–1425. https://doi.org/10.1145/2976749.2978387

[131] Daiping Liu, Zhou Li, Kun Du, Haining Wang, Baojun Liu, and Haixin Duan. 2017. Don't Let One Rotten Apple Spoil the Whole Barrel: Towards Automated Detection of Shadowed Domains. In *Proceedings of the 24th ACM SIGSAC Conference on Computer and Communications Security (CCS)* (24 ed.), David Evans, Tal Malkin, and Dongyan Xu (Eds.). ACM, 537–552. https://doi.org/10.1145/3133956.3134049

[132] Suqi Liu, Ian Foster, Stefan Savage, Geoffrey M. Voelker, and Lawrence K. Saul. 2015. Who Is .Com? Learning to Parse WHOIS Records. In *Proceedings of the 2015 Internet Measurement Conference (IMC)*, Kenjiro Cho, Kensuke Fukuda, Vivek S. Pai, and Neil Spring (Eds.). ACM, 369–380. https://doi.org/10.1145/2815675.2815693

[133] Yang Liu, Armin Sarabi, Jing Zhang, Parinaz Naghizadeh, Manish Karir, Michael Bailey, and Mingyan Liu. 2015. Cloudy with a Chance of Breach: Forecasting Cyber Security Incidents. In *USENIX Security Symposium (USENIX Security)*. USENIX Association, Washington, D.C., USA, 1009–1024. https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/liu

[134] Yang Liu, Armin Sarabi, Jing Zhang, Parinaz Naghizadeh, Manish Karir, Michael Bailey, and Mingyan Liu. 2015. Cloudy with a Chance of Breach: Forecasting Cyber Security Incidents. In *Proceedings of the 24th USENIX Security Symposium (USENIX Security)* (24 ed.), Jaeyeon Jung Jung and Thorsten Holz (Eds.). USENIX Association, 1009–1024. https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/liu

[135] Angelique Faye Loe and Elizabeth Anne Quaglia. 2019. You Shall Not Join: A Measurement Study of Cryptocurrency Peer-to-Peer Bootstrapping Techniques. In *Proceedings of the 26th ACM SIGSAC Conference on Computer and Communications Security (CCS)* (26 ed.), Lorenzo Carvalho, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz (Eds.). ACM, 2231–2247. https://doi.org/10.1145/3319535.3345649

[136] LoRa Alliance. [n. d.]. Homepage - LoRa Alliance. Retrieved 2021-02-18 from https://lora-alliance.org/

[137] Chaoyi Lu, Baojun Liu, Zhou Li, Shuang Hao, Haixin Duan, Mingming Zhang, Chunying Leng, Ying Liu, Zaifeng Zhang, and Jianping Wu. 2019. An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come?. In *Proceedings of the 2019 Internet Measurement Conference (IMC)*, Anna Sperotto, Roland van Rijswijk-Deij, and Cristian Hesselman (Eds.). ACM, 22–35. https://doi.org/10.1145/3355369.3355580

[138] Knud Lasse Lueth. 2020. State of the IoT 2020: 12 billion IoT connections, surpassing non-IoT for the first time. Retrieved 2021-02-18 from https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/

[139] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

[140] Gordon Fyodor Lyon. 2009. *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure.

[141] Ratul Mahajan, David Wetherall, and Tom Anderson. 2002. Understanding BGP misconfiguration. *ACM SIGCOMM Computer Communication Review* 32, 4 (2002), 3–16.

[142] Yaser Mansour. 2021. Rules Authors Introduction to Writing Snort 3 Rules. Retrieved 2021-11-11 from https://www.snort.org/documents/rules-writers-guide-to-snort-3-rules

[143] Srdjan Matic, Platon Kotzias, and Juan Caballero. 2015. CARONTE: Detecting Location Leaks for Deanonymizing Tor Hidden Services. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)* (22 ed.), Ninghui Li and Christopher Kruegel (Eds.). ACM, 1455–1466. https://doi.org/10.1145/2810103.2813667

[144] MaxMind. [n. d.]. GeoIP Legacy Downloadable Databases « MaxMind Developer Site. Retrieved 2023-06-30 from https://web.archive.org/web/20210309090228/https://dev.maxmind.com/geoip/legacy/downloadable/

[145] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. 2019. Reliability and Inter-rater Reliability in Qualitative Research: Norms and Guidelines for CSCW and HCI Practice. In *Proceedings of the 2019 ACM CHI Conference on Human Factors in Computing Systems (CHI)*, Vol. 3. ACM. https://doi.org/10.1145/3359174

[146] P. Mell. 2003. Understanding Intrusion Detection Systems. In *IS Management Handbook*. Auerbach Publications, 409–418.

[147] mergr. [n. d.]. FICO Acquires QuadMetrics | Mergr M&A Deal Summary. Retrieved 2023-06-30 from https://mergr.com/fair-isaac-acquires-quadmetrics

[148] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai. 2018. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. (2018).

[149] Tyler Moore, Scott Dynes, and Frederick R Chang. 2016. Identifying how firms manage cybersecurity investment. In *Workshop on the Economics of Information Security (WEIS)*. 1–27.

[150] Jose G. Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. 2012. A unifying view on dataset shift in classification. *Pattern Recognition* 45, 1 (2012), 521–530. https://doi.org/10.1016/j.patcog.2011.06.019

[151] Kathleen Moriarty and Stephen Farrell. 2021. Deprecating TLS 1.0 and TLS 1.1. RFC 8996. https://doi.org/10.17487/RFC8996

[152] Shun Morishita, Takuya Hoizumi, Wataru Ueno, Rui Tanabe, Carlos Gañán, Michel J.G. van Eeten, Katsunari Yoshioka, and Tsutomu Matsumoto. 2019. Detect Me If You… Oh Wait. An Internet-Wide View of Self-Revealing Honeypots, Joe Betser, Carol J. Fung, Alex Clemm, Jérôme François, and Shingo Ata (Eds.). 134–143. https://ieeexplore.ieee.org/document/8717918

[153] G. C. M. Moura, M. Müller, M. Wullink, and C. Hesselman. 2016. nDEWS: A New Domains Early Warning System for TLDs, Sema Oktug, Mehmet Ulema, Cicek Cavdar, Lisandro Zambenedetti Granville, and Carlos Raniery Paula dos Santos (Eds.). IEEE, 1061–1066. https://doi.org/10.1109/NOMS.2016.7502961

[154] Austin Murdock, Frank Li, Paul Bramsen, Zakir Durumeric, and Vern Paxson. 2017. Target Generation for Internet-Wide IPv6 Scanning. In *Proceedings of the 2017 Internet Measurement Conference (IMC)*, Steve Uhlig and Olaf Maennel (Eds.). ACM, 242–253. https://doi.org/10.1145/3131365.3131405

[155] Johannes Naab, Patrick Sattler, Jonas Jelten, Oliver Gasser, and Georg Carle. 2019. Prefix Top Lists: Gaining Insights with Prefixes from Domain-Based Top Lists on DNS Deployment. In *Proceedings of the 2019 Internet Measurement Conference (IMC)*, Anna Sperotto, Roland van Rijswijk-Deij, and Cristian Hesselman (Eds.). ACM, 351–357. https://doi.org/10.1145/3355369.3355598

[156] A. Nappa, R. Munir, I. Tanoli, C. Kreibich, and J. Caballero. 2016. RevProbe: Detecting Silent Reverse Proxies in Malicious Server Infrastructures. In *Proceedings of the 32nd Annual Computer Security Applications Conference (ACSAC)* (32 ed.). ACM.

[157] National Institute of Standards and Technology. [n. d.]. Cybersecurity Framework | NIST. Retrieved 2023-06-30 from https://www.nist.gov/cyberframework

[158] National Institute of Standards and Technology. 2018. *Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1*. Technical Report NIST CSWP 04162018. National Institute of Standards and Technology, Gaithersburg, MD, USA. NIST CSWP 04162018 pages. https://doi.org/10.6028/NIST.CSWP.04162018

[159] NCSI . 2017. NCSI :: Ranking. https://ncsi.ega.ee/ncsi-index/

[160] Nmap. [n. d.]. nmap/scripts/ssl-dh-params.nse at master · nmap/nmap. Retrieved 2023-06-30 from https://github.com/nmap/nmap/blob/master/scripts/ssl-dh-params.nse

[161] Nokia. 2024. Nokia Data Marketplace | Nokia. https://www.nokia.com/networks/bss-oss/data-marketplace/

[162] Arman Noroozian, Michael Ciere, Maciej Korczynski, Samaneh Tajalizadehkhoob, 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. http://weis2017. econinfosec. org/wp-content/uploads/sites/3/2017/05/WEIS_2017_paper_60. pdf*.

[163] Carl Nykvist, Linus Sjöström, Josef Gustafsson, and Niklas Carlsson. 2018. Server-Side Adoption of Certificate Transparency. In *Proceedings of the 13th Passive and Active Measurement (PAM)* (13 ed.) *(Lecture Notes in Computer Science)*, Robert Beverly and Georgios Smaragdakis (Eds.), Vol. 10771. Springer, 186–199. https://doi.org/10.1007/978-3-319-76481-8_14

[164] The Netherlands Chamber of Commerce. 2022. Energy label C requirement: avoid a fine. https://www.kvk.nl/en/sustainability/energy-label-c-requirement-avoid-a-fine/

[165] U.S. Department of Justice. 2017. US vs. Cazes, Verified Complaint for Forfeiture in Rem. https://www.justice.gov/opa/press-release/file/982821/download.

[166] Council of the European Union. 2016. General Data Protection Regulation (GDPR) – Official Legal Text. https://eur-lex.europa.eu/eli/reg/2016/679/oj

[167] OISF. 2021. Announcing Suricata 5.0.0. Retrieved 2021-11-11 from https://suricata.io/2019/10/15/announcing-suricata-5-0-0/

[168] OISF. 2021. Suricata Update. Retrieved 2021-11-11 from https://suricon.net/wp-content/uploads/2019/01/SuriCon2018_Ish.pdf

[169] Cyril Onwubiko and Karim Ouazzane. 2019. Cyber onboarding is ‘broken’. In *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. IEEE, 1–13.

[170] Open Resolver Project. [n. d.]. Open Resolver Project. Retrieved 2020-06-02 from http://openresolverproject.org/

[171] Alina Oprea, Zhou Li, Robin Norris, and Kevin Bowers. 2018. MADE: Security Analytics for Enterprise Threat Detection. In *Proceedings of the 34th Annual Computer Security Applications Conference (ACSAC ’18)*. Association for Computing Machinery, New York, NY, USA, 124–136. https://doi.org/10.1145/3274694.3274710

[172] Lars Øverlier and Paul Syverson. 2006. Locating Hidden Servers. In *Proceedings of the 27th IEEE Symposium on Security & Privacy (S&P)* (27 ed.). IEEE, 100–114. https://doi.org/10.1109/SP.2006.24

[173] Ramakrishna Padmanabhan, Amogh Dhamdhere, Emile Aben, kc claffy, and Neil Spring. 2016. Reasons Dynamic Addresses Change. In *Proceedings of the 2016 Internet Measurement Conference (IMC)*, Phillipa Gill, John S. Heidemann, John W. Byers, and Ramesh Govindan (Eds.). ACM, 183–198. https://doi.org/10.1145/2987443.2987461

[174] Ramakrishna Padmanabhan, Zhihao Li, Dave Levin, and Neil Spring. 2015. UAv6: Alias Resolution in IPv6 Using Unused Addresses. In *Proceedings of the 10th Passive and Active Measurement (PAM) (Lecture Notes in Computer Science)*, Jelena Mirkovic and Yong Liu (Eds.), Vol. 8995. Springer, 136–148. https://doi.org/10.1007/978-3-319-15509-8_11

[175] Ramakrishna Padmanabhan, Aaron Schulman, Dave Levin, and Neil Spring. 2019. Residential Links under the Weather. In *Proceedings of the 2019 ACM SIGCOMM Conference (SIGCOMM)*, Jianping Wu and Wendy Hall (Eds.). ACM, 145–158. https://doi.org/10.1145/3341302.3342084

[176] J. Park, A. Khormali, M. Mohaisen, and A. Mohaisen. 2019. Where Are You Taking Me? Behavioral Analysis of Open DNS Resolvers. In *Proceedings of the 2019 USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Jay R. Lorch and Minlan Yu (Eds.). ACM, 581–598. https://www.usenix.org/conference/nsdi19/presentation/jin

[177] J. Park, A. Khormali, M. Mohaisen, and A. Mohaisen. 2019. Where Are You Taking Me? Behavioral Analysis of Open DNS Resolvers. In *Proceedings of the*

49th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Karthik Pattabiraman and Fernando Pedone (Eds.). IEEE, 493–504. https://doi.org/10.1109/DSN.2019.00057

[178] Paul Pearce, Ben Jones, Frank Li, Roya Ensafi, Nick Feamster, Nick Weaver, and Vern Paxson. 2017. Global Measurement of {DNS} Manipulation. In *Proceedings of the 26th USENIX Security Symposium (USENIX Security)* (26 ed.), Engin Kirda and Thomas Ristenpart (Eds.). USENIX Association, 307–323. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/pearce

[179] Giancarlo Pellegrino, Onur Catakoglu, Davide Balzarotti, and Christian Rossow. 2016. Uses and Abuses of Server-Side Requests. In *Proceedings of the 19th International Symposium on Recent Advances in Intrusion Detection (RAID)* (19 ed.) *(Lecture Notes in Computer Science)*, Fabian Monrose, Marc Dacier, Gregory Blanc, and Joaquin Garcia-Alfaro (Eds.), Vol. 9854. Springer, 393–414. https://doi.org/10.1007/978-3-319-45719-2_18

[180] Ponemon Institute LLC. 2019. Improving the Effectiveness of the Security Operations Center. http://www.surfline.com/surf-news/maldives-surf-access-controversy-update_75296/

[181] Adrienne Porter Felt, Richard Barnes, April King, Chris Palmer, Chris Bentzel, and Parisa Tabriz. 2017. Measuring {HTTPS} Adoption on the Web. In *Proceedings of the 26th USENIX Security Symposium (USENIX Security)* (26 ed.), Engin Kirda and Thomas Ristenpart (Eds.). USENIX Association, 551–566. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/felt

[182] The Zeek Project. 2020. The Zeek Network Security Monitor. Retrieved 2021-04-16 from https://zeek.org/

[183] Proofpoint. 2021. Daily Ruleset Update Summary 2020/02/24. Retrieved 2021-11-11 from https://www.proofpoint.com/us/daily-ruleset-update-summary-20200224

[184] Proofpoint. 2021. Daily Ruleset Update Summary 2020/02/25. Retrieved 2021-11-11 from https://www.proofpoint.com/us/daily-ruleset-update-summary-20200225

[185] Proofpoint. 2021. Daily Ruleset Update Summary 2020/02/26. Retrieved 2021-11-11 from https://www.proofpoint.com/us/daily-ruleset-update-summary-20200226

[186] Proofpoint. 2021. Emerging Threats Pro Ruleset | Proofpoint. Retrieved 2021-03-24 from https://www.proofpoint.com/us/threat-insight/et-pro-ruleset

[187] Proofpoint. 2021. Proofpoint Emerging Threats Rules. Retrieved 2021-04-28 from https://rules.emergingthreats.net/

[188] psbl.org. 2023. Passive Spam Block List. Retrieved 2023-04-30 from https://psbl.org/

[189] Christopher Rentrop and Stephan Zimmermann. 2012. Shadow IT: Management and Control of Unofficial IT. In *Proceedings of the The 6th International Conference on Digital Society (ICDS)* (6 ed.). International Academic, Research, and Industry Association (IARIA), 98–102.

[190] RIPE. 2019. Routing Information Service (RIS). Retrieved 2020-06-22 from https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/routing-information-service-ris

[191] RIPE NCC. 2021. RIPEstat docs | Historical Whois | Docs. Retrieved 2023-04-30 from https://m.stat.ripe.net/docs/02.data-api/historical-whois.html

[192] RouteViews. [n. d.]. Routeviews – University of Oregon Route Views Project. Retrieved 2020-06-22 from http://www.routeviews.org/routeviews/

[193] RouteViews. 2023. RouteViews – University of Oregon RouteViews Project. Retrieved 2023-06-30 from https://www.routeviews.org/routeviews/

[194] Jan Rüth, Ingmar Poese, Christoph Dietzel, and Oliver Hohlfeld. 2018. A First Look at QUIC in the Wild. In *Proceedings of the 13th Passive and Active Measurement (PAM)* (13 ed.) *(Lecture Notes in Computer Science)*, Robert Beverly and Georgios Smaragdakis (Eds.), Vol. 10771. Springer, 255–268. https://doi.org/10.1007/978-3-319-76481-8_19

[195] Hillary Sanders and Joshua Saxe. 2017. Garbage in, garbage out: how purportedly great ML models can be screwed up by bad data. *Proceedings of Blackhat* 2017 (2017).

[196] Armin Sarabi and Mingyan Liu. 2018. Characterizing the Internet Host Population Using Deep Learning: A Universal and Lightweight Numerical Embedding. In *Proceedings of the Internet Measurement Conference 2018 (IMC '18)*. Association for Computing Machinery, New York, NY, USA, 133–146. https://doi.org/10.1145/3278532.3278545

[197] Armin Sarabi, Parinaz Naghizadeh, Yang Liu, and Mingyan Liu. 2016. Risky business: Fine-grained data breach prediction using business profiles. *Journal of Cybersecurity* 2, 1 (2016), 15–28.

[198] Sarah Scheffler, Sean Smith, Yossi Gilad, and Sharon Goldberg. 2018. The Unintended Consequences of Email Spam Prevention. In *Proceedings of the 13th Passive and Active Measurement (PAM)* (13 ed.) *(Lecture Notes in Computer Science)*, Robert Beverly and Georgios Smaragdakis (Eds.), Vol. 10771. Springer, 158–169. https://doi.org/10.1007/978-3-319-76481-8_12

[199] Quirin Scheitle, Oliver Gasser, Theodor Nolte, Johanna Amann, Lexi Brent, Georg Carle, Ralph Holz, Thomas C. Schmidt, and Matthias Wählisch. 2018. The Rise of Certificate Transparency and Its Implications on the Internet Ecosystem. In *Proceedings of the 2018 Internet Measurement Conference (IMC)*, Ben Y. Zhao and Ethan Katz-Bassett (Eds.). ACM, 343–349. https://doi.org/10.1145/3278532.3278562

[200] Quirin Scheitle, Oliver Gasser, Minoo Rouhi, and Georg Carle. 2017. Large-Scale Classification of IPv6-IPv4 Siblings with Variable Clock Skew. In *Proceedings of the 2017 International Workshop on Traffic Monitoring and Analysis (TMA)*, Marco Mellia, Emir Halepovic, and David Malone (Eds.). IEEE, 1–9. https://doi.org/10.23919/TMA.2017.8002901

[201] Kyle Schomp, Tom Callahan, Michael Rabinovich, and Mark Allman. 2013. On Measuring the Client-Side DNS Infrastructure. In *Proceedings of the 2013 Internet Measurement Conference (IMC)*, Konstantina Papagiannaki, P. Krishna Gummadi, and Craig Partridge (Eds.). ACM, 77–90. https://doi.org/10.1145/2504730.2504734

[202] Will Scott, Thomas Anderson, Tadayoshi Kohno, and Arvind Krishnamurthy. 2016. Satellite: Joint Analysis of CDNs and Network-Level Interference, Ajay Gulati and Hakim Weatherspoon (Eds.). USENIX Association, 195–208. https://www.usenix.org/conference/atc16/technical-sessions/presentation/scott

[203] SecurityScorecard. [n. d.]. Security Ratings & Cybersecurity Risk Management | SecurityScorecard. Retrieved 2023-06-30 from https://securityscorecard.com/

[204] SecurityScorecard. 2023. CVSS Score Distribution Reports and Trends Over Time. https://www.cvedetails.com/cvss-score-charts.php?fromform 1&vendor_id &product_id &startdate 2021 - 01 - 01&enddate 2022 - 12 - 31&groupbyyear1

[205] Soumya Sen. 2006. Performance characterization & improvement of snort as an IDS. *Bell Labs Report* (2006).

[206] Ankit Shah, Rajesh Ganesan, Sushil Jajodia, and Hasan Cam. 2018. Understanding tradeoffs between throughput, quality, and cost of alert analysis in a CSOC. *IEEE Transactions on Information Forensics and Security* 14, 5 (2018), 1155–1170.

[207] Syed Ali Raza Shah and Biju Issac. 2018. Performance comparison of intrusion detection systems and application of machine learning to Snort system. *Future Generation Computer Systems* 80 (2018), 157–170.

[208] Shodan. [n. d.]. Shodan. Retrieved 2023-05-01 from https://www.shodan.io/

[209] N Shone, T.N. Ngoc, V.D. Phai, and Q. Shi. 2018. A Deep Learning Approach to Network Intrusion Detection. *IEEE Trans. on Emerging Topics in Computational Intelligence (TETCI)* 2, 1 (2018), 41–50.

[210] Haya Shulman and Michael Waidner. 2017. One Key to Sign Them All Considered Vulnerable: Evaluation of DNSSEC in the Internet. In *Proceedings of the 2017 USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Aditya Akella and Jon Howell (Eds.). ACM, 131–144. https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/shulman

[211] Matthew Shutock and Glenn Dietrich. 2022. Security Operations Centers: A Holistic View on Problems and Solutions. In *Proceedings of the 55th Hawaii International Conference on System Sciences*.

[212] Sessika Siregar and Kuo-Chung Chang. 2019. Cybersecurity agility: antecedents and effects on security incident management effectiveness. In *23rd Pacific Asia Conference on Information Systems (PACIS 2019)*. 8–12.

[213] Robin Sommer and Vern Paxson. 2003. Enhancing byte-level network intrusion detection signatures with context. In *Proceedings of the 10th ACM Conference on*

Computer and Communications Security (CCS). Association for Computing Machinery, 262–271. https://doi.org/10.1145/948109

[214] J. Sonchack, A. Dubey, A. Aviv, J. Smith, and E. Keller. 2016. Timing-based Reconnaissance and Defense in Software-defined Networks. In *Proceedings of the 32nd Annual Computer Security Applications Conference (ACSAC)* (32 ed.). ACM.

[215] Awalin Sopan, Matthew Berninger, Murali Mulakaluri, and Raj Katakam. 2018. Building a Machine Learning Model for the SOC, by the Input from the SOC, and Analyzing it for the SOC. In *2018 IEEE Symposium on Visualization for Cyber Security (VizSec)*. 1–8. https://doi.org/10.1109/VIZSEC.2018.8709231

[216] Kyle Soska and Nicolas Christin. 2014. Automatically Detecting Vulnerable Websites Before They Turn Malicious. In *USENIX Security Symposium (USENIX Security)*. USENIX Association, San Diego, CA, USA, 625–640. https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/soska

[217] Kyle Soska and Nicolas Christin. 2015. Measuring the Longitudinal Evolution of the Online Anonymous Marketplace Ecosystem. In *Proceedings of the 24th USENIX Security Symposium (USENIX Security)* (24 ed.), Jaeyeon Jung Jung and Thorsten Holz (Eds.). USENIX Association, 33–48. https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/soska

[218] Drew Springall, Zakir Durumeric, and J. Alex Halderman. 2016. FTP: The Forgotten Cloud. In *Proceedings of the 46th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Domenico Cotroneo and Cristina Nita-Rotaru (Eds.). IEEE, 503–513. https://doi.org/10.1109/DSN.2016.52

[219] Drew Springall, Zakir Durumeric, and J. Alex Halderman. 2016. Measuring the Security Harm of TLS Crypto Shortcuts. In *Proceedings of the 2016 Internet Measurement Conference (IMC)*, Phillipa Gill, John S. Heidemann, John W. Byers, and Ramesh Govindan (Eds.). ACM, 33–47. https://doi.org/10.1145/2987443.2987480

[220] N. Srivastav and R.K. Challa. 2013. Novel Intrusion Detection System Integrating Layered Framework with Neural Network. In *Proceedings of the 2013 IEEE Advance Computing Conference (IACC)*. IEEE, IEEE, 682–689.

[221] Sathya Chandran Sundaramurthy, Alexandru G Bardas, Jacob Case, Xinming Ou, Michael Wesch, John McHugh, and S Raj Rajagopalan. 2015. A human capital model for mitigating security analyst burnout. In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*. 347–359.

[222] Sathya Chandran Sundaramurthy, John McHugh, Xinming Ou, Michael Wesch, Alexandru G Bardas, and S Raj Rajagopalan. 2016. Turning contradictions into innovations or: How we learned to stop whining and improve security operations. In *Twelfth Symposium on Usable Privacy and Security ({SOUPS} 2016)*. 237–251.

[223] Suricata. 2021. Suricata | Open Source IDS / IPS / NSM engine. Retrieved 2021-04-16 from https://suricata-ids.org/

[224] Samaneh Tajalizadehkhoob, Maciej Korczynski, Arman Noroozian, Carlos Ganan, and Michel van Eeten. 2016. Apples, Oranges and Hosting Providers: Heterogeneity and Security in the Hosting Market, Sema Oktug, Mehmet Ulema, Cicek Cavdar, Lisandro Zambenedetti Granville, and Carlos Raniery Paula dos Santos (Eds.). IEEE, 289–297. https://doi.org/10.1109/NOMS.2016.7502824

[225] Samaneh Tajalizadehkhoob, Tom Van Goethem, Maciej Korczyński, Arman Noroozian, Rainer Böhme, Tyler Moore, Wouter Joosen, and Michel van Eeten. 2017. Herding Vulnerable Cats: A Statistical Approach to Disentangle Joint Responsibility for Web Security in Shared Hosting. In *Proceedings of the 24th ACM SIGSAC Conference on Computer and Communications Security (CCS)* (24 ed.), David Evans, Tal Malkin, and Dongyan Xu (Eds.). ACM, 553–567. https://doi.org/10.1145/3133956.3133971

[226] Nassim Nicholas Taleb. 2007. *The black swan: The impact of the highly improbable*. Vol. 2. Random house.

[227] Emerging Threats Research Team. 2021. Emerging Threats: Announcing Support for Suricata 5.0. Retrieved 2021-11-11 from https://www.proofpoint.com/us/corporate-blog/post/emerging-threats-announcing-support-suricata-50

[228] The Shadowserver Foundation. 2023. Network Reporting | The Shadowserver Foundation. Retrieved 2023-04-30 from https://www.shadowserver.org/what-we-do/network-reporting/

[229] The Shadowserver Foundation. 2023. The Shadowserver Foundation. Retrieved 2023-04-30 from https://www.shadowserver.org/

[230] The Spamhaus Project SLU. 2023. XBL - Exploit and Botnet Filter - The Spamhaus Project. Retrieved 2023-04-30 from https://www.spamhaus.org/xbl/

[231] The ZMap Project. 2016. Zmap/Zgrab2: Fast Go Application Scanner. The ZMap Project. Retrieved 2020-06-02 from https://github.com/zmap/zgrab2

[232] Kittikhun Thongkanchorn, Sudsanguan Ngamsuriyaroj, and Vasaka Visoottiviseth. 2013. Evaluation studies of three intrusion detection systems under various attacks and rule sets. In *Proceedings of the 2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)*. 1–4. https://doi.org/10.1109/TENCON.2013.6718975

[233] U.S. Attorney's Office, Southern District of New York. 2014. Dozens Of Online "Dark Markets" Seized Pursuant To Forfeiture Complaint Filed In Manhattan Federal Court In Conjunction With The Arrest Of The Operator Of Silk Road 2.0. http://www.justice.gov/usao/nys/pressreleases/November14/DarkMarketTakedown.php.

[234] Ivo Vacas, Ibéria Medeiros, and Nuno Neves. 2018. Detecting Network Threats using OSINT Knowledge-Based IDS. In *2018 14th European Dependable Computing Conference (EDCC)*. 128–135. https://doi.org/10.1109/EDCC.2018.00031

[235] Olivier van der Toorn, Roland van Rijswijk-Deij, Tobias Fiebig, Martina Lindorfer, and Anna Sperotto. 2020. TXTing 101: Finding Security Issues in the Long Tail of DNS TXT Records. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 544–549.

[236] Benjamin VanderSloot, Allison McDonald, Will Scott, J. Alex Halderman, and Roya Ensafi. 2018. Quack: Scalable Remote Measurement of Application-Layer Censorship. In *Proceedings of the 27th USENIX Security Symposium (USENIX Security)* (27 ed.), William Enck and Adrienne Porter Felt (Eds.). USENIX Association, 187–202. https://www.usenix.org/conference/usenixsecurity18/presentation/vandersloot

[237] Matteo Varvello, Kyle Schomp, David Naylor, Jeremy Blackburn, Alessandro Finamore, and Konstantina Papagiannaki. 2016. Is the Web HTTP/2 Yet?. In *Proceedings of the 10th Passive and Active Measurement (PAM) (Lecture Notes in Computer Science)*, Thomas Karagiannis and Xenofontas Dimitropoulos (Eds.), Vol. 9631. Springer, 218–232. https://doi.org/10.1007/978-3-319-30505-9_17

[238] Verizon Security Research & Cyber Intelligence Center. [n. d.]. The VERIS Framework. Retrieved 2023-06-30 from http://veriscommunity.net/index.html

[239] Mathew Vermeer, Michel van Eeten, and Carlos Gañán. 2022. Ruling the rules: Quantifying the evolution of rulesets, alerts and incidents in network intrusion detection. In

Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security. Association for Computing Machinery, 799–814.

[240] Giovanni Vigna. 2010. Network Intrusion Detection: Dead or Alive?. In *Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC '10)*. Association for Computing Machinery, New York, NY, USA, 117–126. https://doi.org/10.1145/1920261.1920279

[241] Giovanni Vigna, William Robertson, and Davide Balzarotti. 2004. Testing network-based intrusion detection signatures using mutant exploits. In *Proceedings of the 11th ACM Conference on Computer and Communications Security*. Association for Computing Machinery, 21–30. https://doi.org/10.1109/WAINA20374.2012

[242] Thomas Vissers, Tom Van Goethem, Wouter Joosen, and Nick Nikiforakis. 2015. Maneuvering Around Clouds: Bypassing Cloud-Based Security Providers. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)* (22 ed.), Ninghui Li and Christopher Kruegel (Eds.). ACM, 1530–1541. https://doi.org/10.1145/2810103.2813633

[243] Hao Wang, Somesh Jha, and Vinod Ganapathy. 2006. NetSpy: Automatic generation of spyware signatures for NIDS. In *2006 22nd Annual Computer Security Applications Conference (ACSAC)*. IEEE, 99–108.

[244] Tian Wang, Chen Zhang, Zhigang Lu, Dan Du, and Yaopeng Han. 2019. Identifying Truly Suspicious Events and False Alarms Based on Alert Graph. In *2019 IEEE International Conference on Big Data (Big Data)*. 5929–5936. https://doi.org/10.1109/BigData47090.2019.9006555

[245] Rolf van Wegberg, Samaneh Tajalizadehkhoob, Kyle Soska, Ugur Akyazi, Carlos Hernandez Gáñan, Bram Klievink, Nicolas Christin, and Michel van Eeten. 2018. Plug and Prey? Measuring the Commoditization of Cybercrime via Online Anonymous Markets. In *Proceedings of the 27th USENIX Security Symposium (USENIX Security)* (27 ed.), William Enck and Adrienne Porter Felt (Eds.). USENIX Association. https://www.usenix.org/conference/usenixsecurity18/presentation/van-wegberg

[246] Joshua S. White, Thomas Fitzsimmons, and Jeanna N. Matthews. 2013. Quantitative analysis of intrusion detection systems: Snort and Suricata. In *Cyber Sensing 2013*, Igor V. Ternovskiy and Peter Chin (Eds.), Vol. 8757. International Society for Optics and Photonics, SPIE, 10 – 21. https://doi.org/10.1117/12.2015616

[247] Maarten Wullink, Giovane C. M. Moura, and Cristian Hesselman. 2018. Dmap: Automating Domain Name Ecosystem Measurements and Applications. In *2018 Network Traffic Measurement and Analysis Conference (TMA)*. 1–8. https://doi.org/10.23919/TMA.2018.8506521

[248] "xgboost developers". [n. d.]. XGBoost Documentation — xgboost 1.7.6 documentation. Retrieved 2023-06-30 from https://xgboost.readthedocs.io/en/stable/index.html

[249] Chaowei Xiao, Armin Sarabi, Yang Liu, Bo Li, Mingyan Liu, and Tudor Dumitras. 2018. From Patching Delays to Infection Symptoms: Using Risk Profiles for an Early Discovery of Vulnerabilities Exploited in the Wild. In *USENIX Security Symposium (USENIX Security)*. USENIX Association, Baltimore, MD, 903–918. https://www.usenix.org/conference/usenixsecurity18/presentation/xiao

[250] Haitao Xu, Fengyuan Xu, and Bo Chen. 2018. Internet Protocol Cameras with No Password Protection: An Empirical Investigation. In *Proceedings of the 13th Passive and Active Measurement (PAM)* (13 ed.) *(Lecture Notes in Computer Science)*, Robert Beverly and Georgios Smaragdakis (Eds.), Vol. 10771. Springer, 47–59. https://doi.org/10.1007/978-3-319-76481-8_4

[251] Bahador Yeganeh, Ramakrishnan Durairajan, Reza Rejaie, and Walter Willinger. 2019. How Cloud Traffic Goes Hiding: A Study of Amazon's Peering Fabric. In *Proceedings of the 2019 Internet Measurement Conference (IMC)*, Anna Sperotto, Roland van Rijswijk-Deij, and Cristian Hesselman (Eds.). ACM, 202–216. https://doi.org/10.1145/3355369.3355602

[252] Vinod Yegneswaran, Jonathon T Giffin, Paul Barford, and Somesh Jha. 2005. An Architecture for Generating Semantic Aware Signatures.. In *Proceedings of the 14th USENIX Security Symposium (USENIX Security)*. 97–112.

[253] Jing Zhang, Zakir Durumeric, Michael Bailey, Mingyan Liu, and Manish Karir. 2014. On the Mismanagement and Maliciousness of Networks. In *Proceedings of the 2014 Network and Distributed System Security Symposium*. Internet Society, San Diego, CA, USA. https://doi.org/10.14722/ndss.2014.23057

[254] Jing Zhang, Zakir Durumeric, Michael Bailey, Mingyan Liu, and Manish Karir. 2014. On the Mismanagement and Maliciousness of Networks.. In *Network and Distributed System Security Symposium (NDSS)*, Vol. 14. 23–26.

APPENDIX A

A.1. INTERVIEW STUDY PROTOCOL

1) Welcome

2) Short overview of the study

3) Explanation of the interview

4) Informed consent

5) Start interview

6) Debriefing

A.2. INTERVIEW QUESTIONS

A.2.1. WORKFLOWS

1) Could you describe to me your workflow? a. Probe about routine and non-routine tasks

2) What do you see as the main objectives of your work? a. Probe on incentives that they have to reach this objective

3) Can you walk me through the process of the acquiring, creating, changing, and deactivating rules? a. Probe on when a rule is added into the sensor b. Probe on all the testing that is done before rules are added into the sensor

4) How many rules do you investigate every day? a. Probe on how they perceive this task (creative / procedure / workload) b. Do you have any tasks that you don't have enough time for? c. Probe rule evaluation

A.2.2. MANAGEMENT

5) How do you work together with other colleagues on making or changing rules? a. Probe on working together or separation of tasks in specific client's rulesets? b. Probe if they ever had a disagreement on certain rules

6) How do you seek additional information in order to assess a rule? a. Probe on who they asked, what advice they received. b. What kind data they were looking for.

7) Did someone ever follow up with you after you made or changed a rule?

8) In your experience, what is the most severe thing that could go wrong with the rulesets you are using? a. Probe on fear of FN b. What are potential consequences of a ruleset that isn't functioning properly c. Probe on the amount of risk that they perceive on missing TP d. Probe on the amount of FP and their perception and definition of a FP e. Probe on how likely they think consequences might happen

9) Have there been made any mistakes while adapting rulesets? a. Probe on how this came to light.

10) What procedures does the organization have on making or changing rules?

11) Who is responsible for the quality of the rules? a. Probe on differences between the responsibility of individual, senior, manager.

12) How is a client involved in the creation of rules? a. Probe on relationship with clients

13) Can you give an example of feedback that you received from clients?

14) In your opinion, what could be improved in the management of rules?

A.2.3. OBJECTIVES

15) In your opinion, what is a good ruleset? a. Probe on the influence of the volume of a ruleset

16) How do you optimize a ruleset as a whole?

17) What is the best ruleset that is achievable in practice?

A.2.4. EVALUATING RULES

18) Can you walk me through the process on how you determine whether a rule is good or bad?

19) Can you give an example of a good rule? a. Probe on why this is a good rule

20) Can you give an example of a bad rule? a. Probe on why this is a bad rule

21) Which data do you use for evaluating rules? a. Probe on what they think is the most important data

22) Is there additional data that you would like to have?

23) What do you do when you have doubts on a rule?

24) How do you deal with rules that do not or no longer generate any alerts?

25) Is there anything else you would like to tell me that could benefit our research?

26) Live evaluation of sample rule #1.

27) How would you evaluate the previous rule if it is known to have generated 4000 false positive alerts in the last month?

28) Live evaluation of sample rule #2.

29) How would you evaluate the previous rule if it is known to have generated 20 true positives in last month?

30) Live evaluation of sample rule #3.

31) How would you evaluate the previous rule, taking its performance impact into account, if it generates a single true positive in a year?

A.2.5. CLOSING DEMOGRAPHICS

32) What is your name?

33) What is your job title?

34) How old are you?

35) What is your educational level?

36) How many years have you been doing this work?

37) Do you know anyone else who we could interview for this research?

A.3. INTERVIEW SAMPLE RULES

Sample rule 1

```
alert http $HOME_NET any -> $EXTERNAL_NET any (
  msg:"ET POLICY Vulnerable Java Version 1.8.x Detected";
  flow:established,to_server;
  content:" Java/1.8.0_"; http_user_agent;
  content:!"251"; within:3; http_user_agent;
  flowbits:set,ET.http.javaclient.vulnerable;
  threshold: type limit, count 2, seconds 300, track by_src;
  metadata: former_category POLICY;
  reference:url,www.oracle.com/technetwork/java/javase/8u-relnotes-2225394.html;
  classtype:bad-unknown;
  sid:2019401;
  rev:30;
  metadata:affected_product Java, attack_target Client_Endpoint, deployment Perimeter,
      ↪  deployment Internal, signature_severity Informational, created_at 2014
      ↪ _10_15, performance_impact Low, updated_at 2020_04_27;
)
```

Sample rule 2

```
alert tcp $HOME_NET any -> any any (
  msg:"ET EXPLOIT Possible OpenSSL? HeartBleed? Large HeartBeat? Response (Client Init
      ↪  Vuln Server)";
  flow:established,to_client;
  content:"|18 03|";depth:2;byte_test:1,<,4,2;
  flowbits:isset,ET.HB.Request.CI;
  flowbits:isnotset,ET.HB.Response.CI;
  flowbits:set,ET.HB.Response.CI;
  flowbits:unset,ET.HB.Request.CI;
  byte_test:2,>,150,3;
  threshold:type limit,track by_src,count 1,seconds 120;
  metadata: former_category CURRENT_EVENTS;
  reference:cve,2014-0160;
  reference:url,blog.inliniac.net/2014/04/08/detecting-openssl-heartbleed-with-
      ↪ suricata/;
  reference:url,heartbleed.com/;
  reference:url,blog.fox-it.com/2014/04/08/openssl-heartbleed-bug-live-blog/;
  classtype:bad-unknown;
  sid:2018377;
  rev:4;
  metadata:created_at 2014_04_09, updated_at 2014_04_09;
)
```

Sample rule 3

```
alert http $HOME_NET any -> $EXTERNAL_NET any (msg: "ET TROJAN [PTsecurity] Tinba (
    ↪ Banking Trojan) Check-in";
  flow: established,to_server;
  content:!"Referer|3a|";
  http_header;
  content: "|0d0a0d0a|";
  depth: 2000;
  byte_extract: 2, 0, byte0, relative;
  byte_extract: 2, 0, byte1, relative;
  byte_test: 2, =, byte1, 6, relative;
  byte_test: 2, !=, byte1, 7, relative;
  byte_test: 2, =, byte1, 10, relative;
  byte_test: 2, !=, byte1, 11, relative;
  byte_test: 2, !=, byte1, 23, relative;
  byte_test: 2, !=, byte0, 25, relative;
  byte_test: 2, !=, byte1, 27, relative;
  byte_test: 2, =, byte0, 40, relative;
```

```
    byte_test: 2, =, byte1, 42, relative;
    byte_test: 2, =, byte0, 44, relative;
    byte_test: 2, =, byte1, 46, relative;
    byte_test: 2, =, byte0, 48, relative;
    byte_test: 2, =, byte1, 50, relative;
    content:!"|0000|";depth:30; http_client_body;
    content: "|0000|";offset:34;depth:2; http_client_body; fast_pattern;
    content: "|0000|";distance:2;within:2; http_client_body;
    content: "|0000|";distance:2;within:2; http_client_body;
    metadata: former_category TROJAN;
    reference:md5,be312fdb94f3a3c783332ea91ef00ebd;
    classtype:trojan-activity;
    sid:10003433;
    rev:1;
    metadata:affected_product Windows_XP_Vista_7_8_10_Server_32_64_Bit, attack_target
        ↪ Client_Endpoint, deployment Perimeter, tag Banker, signature_severity Major,
        ↪  created_at 2018_08_07, malware_family Tinba, performance_impact High;
)
```

APPENDIX B

B.1. ORGANIZATION AND IP BLOCK MAPPING FLOWCHART

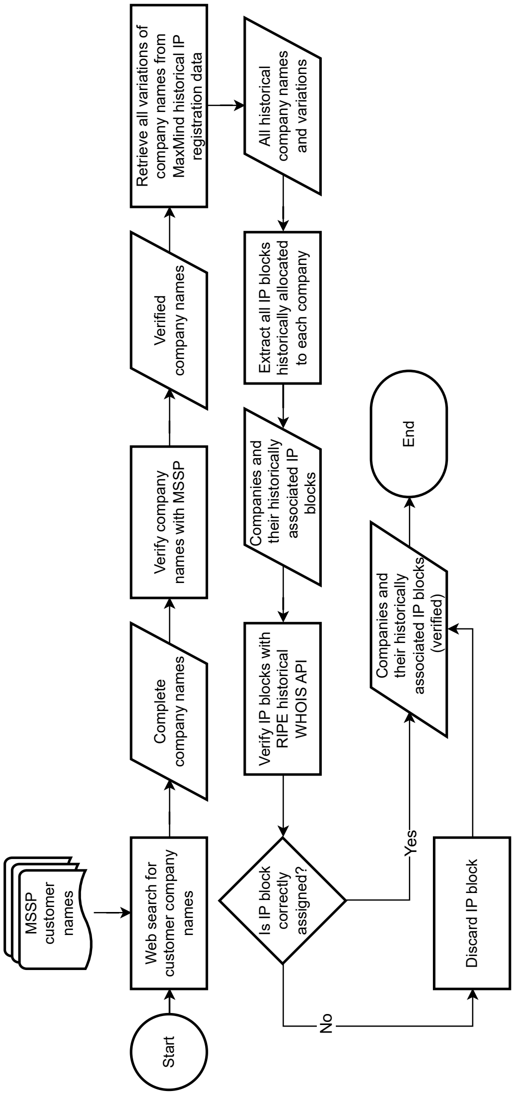

Start
MSSP customer names
Web search for customer company names
Complete company names
Verify company names with MSSP
Verified company names
Retrieve all variations of company names from MaxMind historical IP registration data
All historical company names and variations
Extract all IP blocks historically allocated to each company
Companies and their historically associated IP blocks
Verify IP blocks with RIPE historical WHOIS API
Is IP block correctly assigned?
Yes
No
Discard IP block
Companies and their historically associated IP blocks (verified)
End

Figure 1: Flowchart of the manual process conducted to map organizations to their corresponding IP blocks.

B.2. SHAP DECISION PLOTS

This appendix contains decision plots for several of the model's predictions at specific points in time. The vertical gray line indicates the expected value (i.e., the mean of the data) at the moment that prediction was made.

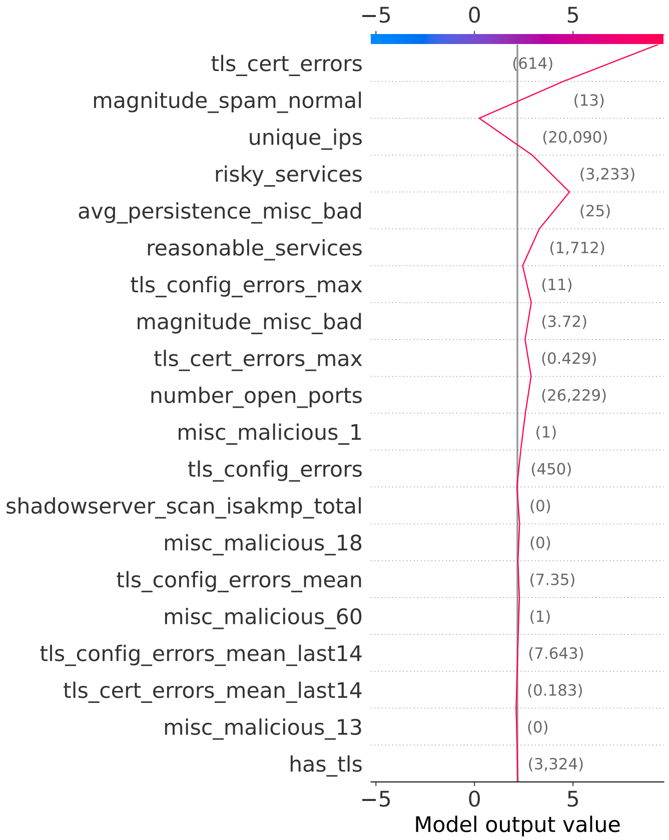


Figure 2: Decision plot of an inaccurate high risk incident prediction. The actual high risk incident count is 0.

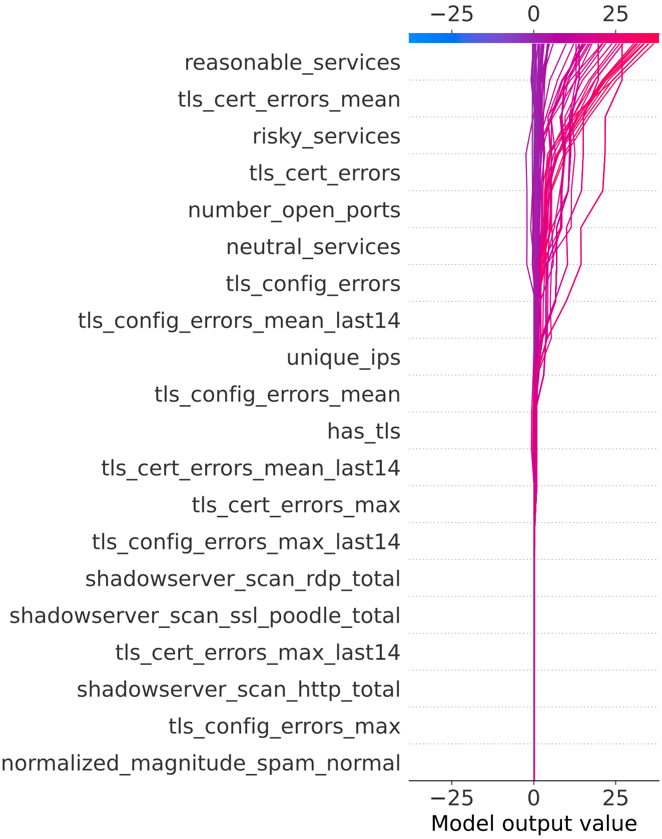


Figure 3: Decision plot of a cluster of successful hack attacks corresponding to the same organization. Even though the incident counts remain elevated throughout this period, the model is not able to perform accurate or consistent predictions.

B.3. List of features

Table 1: All 338 features extracted and used.

Feature	Count
Malicious activity	***180***
- Number of IPs in spam datasets for each day in time window	60
- Number of IPs in phishing datasets for each day in time window	60
- Number of IPs in misc. malicious activity datasets for each day in time window	60
Persistence	***72***
- Magnitude of persistence in good region of spam datasets in time window	1
- Magnitude of persistence in normal region of spam datasets in time window	1
- Magnitude of persistence in bad region of spam datasets in time window	1
- Normalized magnitude of persistence in good region of spam datasets in time window	1
- Normalized magnitude of persistence in normal region of spam datasets in time window	1
- Normalized magnitude of persistence in bad region of spam datasets in time window	1
- Average duration of persistence in good region of spam datasets in time window	1
- Average duration of persistence in normal region of spam datasets in time window	1
- Average duration of persistence in bad region of spam datasets in time window	1
- Frequency at which time series enters good region of spam datasets in time window	1
- Frequency at which time series enters normal region of spam datasets in time window	1
- Same features, spam datasets, but for last 14 days of time window	*12*
- Same features, phishing datasets	*12*
- Same features, phishing datasets, but for last 14 days of time window	*12*
- Same features, misc. malicious datasets	*12*
- Same features, misc. malicious datasets, but for last 14 days of time window	*12*
Mismanagement symptoms	***15***
- Number of open recursive resolvers in time window	1
- Maximum and mean open recursive resolvers in time window	2
- Maximum and mean open recursive resolvers in last 14 days time window	2
- Same features for TLS configuration errors	*5*
- Same features for TLS certificate errors	*5*
Live hosts	***3***
- Number of unique IPs with open ports in time window	1
- Number of open ports in time window	1
- Number of TLS-enabled services in time window	1
Types of services	***3***
- Number of reasonable services in time window	1
- Number of neutral services in time window	1
- Number of risky services in time window	1
ShadowServer features (IPs present in report)	***65***
- Blacklist report	1
- Block list report	1
- Drone/Botnet-Drone Report	1
- CAIDA IP Spoofer report	1
- Command and Control Report	1
- Accessible Cisco Smart Install Report	1
- Compromised Website Report	1
- Sandbox URL Report	1
- Darknet Report	1
- Amplification DDoS Victim Report	1
- DNS Open Resolvers Report	1
- Brute Force Attack Report	1
- Honeypot HTTP Scanner Events Report	1
- Honeypot ICS Scanner Events Report	1
- Microsoft Sinkhole Report	1
- Netcore/Netis Router Vulnerability Scan Report	1
- Accessible SSL Report	1
- Open Proxy Report	1
- Outdated DNSSEC Key Report	1
- Accessible ADB Report	1
- Accessible AFP Report	1
- Accessible Apple Remote Desktop (ARD) Report	1
- Accessible CharGen Report	1
- Accessible CoAP Report	1
- Open CWMP Report	1
- Open DB2 Discovery Service	1
- Accessible DNS Report	1
- Open Elasticsearch Report	1
- Accessible FTP Report	1
- Accessible Hadoop Report	1
- Accessible HTTP Report	1
- Vulnerable HTTP Report	1
- Open IPMI Report	1
- Open IPP Report	1
- Vulnerable ISAKMP Report	1
- Open LDAP TCP Report	1
- Open LDAP Report	1
- Open mDNS Report	1
- Open Memcached Report	1
- Open MongoDB Report	1
- Open MQTT Report	1
- Open MS-SQL Server Resolution Service Report	1
- Open NAT-PMP Report	1
- NTP Version Report	1
- NTP Monitor Report	1
- Open QOTD Report	1
- Accessible Radmin Report	1
- Accessible RDP Report	1
- Accessible MS-RDPEUDP	1
- Open Redis Report	1
- Accessible Rsync Report	1
- Accessible SMB Report	1
- Open SNMP Report	1
- Open SSDP Report	1
- SSL FREAK Report	1
- SSL POODLE Report	1
- Accessible Telnet Report	1
- Open/Accessible TFTP	1
- Open Ubiquiti Report	1
- Accessible VNC Report	1
- Accessible XDMCP Service Report	1
- Sinkhole DNS report	1
- Sinkhole HTTP Drone Report	1
- Spam URL Report	1
- Vulnerable Exchange Servers Special Report	1

ACKNOWLEDGEMENTS

“If there’s one thing I know for sure, it’s that I don’t want to do a PhD.” Those were my words back in the spring of 2019 when I was still working on my Master’s thesis. Though it took some convincing, and many conversations with many people that had an opinion on the matter, I now stand at the tail end of that same PhD journey I was so certain of never wanting to embark on. On that journey I was blessed by the presence of many that, in some way or another, helped me reach that final objective. These few pages go out to them.

To my promotor Michel, the many Thursday beers definitely paid off: I’m getting my diploma and you got your name in this book. But on a serious note, I want to express my gratitude to you for the warm welcome you gave me to the team, and the guidance I received throughout the PhD, especially during the nightmare that was the SoK. Starting out, you asked me where I would ideally like to present a paper. Having never been more certain of anything in my life, I quickly answered “Japan,” after which you calmly responded with “we can make that happen.” Never sure if that was still part of the PhD marketing talk, I’m still grateful that we were indeed able to make that happen.

Carlos, thank you for all the support, supervision, and willingness to sign any document I put in front of you. The team-wide coffee breaks you organized when you were still around at the office were some of my favorite moments during my first year and always came at the right time. I appreciate you involving me in Master student supervision and encouraging me to recruit my own. They were enjoyable experiences that helped me develop professionally in ways I didn’t know were useful. I also need to tell you that I consider it a bit of an achievement that I never once heard from you that my written work was Somewhat Heavily Inadequate Text, considering your reputation as reviewer and supervisor.

Tobias and Samaneh, you were instrumental in my decision to pursue a PhD. Tobias, your sales pitches were what initially intrigued me about the possibility of going down this path. Thank you for taking the time to convince me. And Samaneh, thanks for taking time out of your busy day to chat with me about your own journey, and reassuring me that Michel isn’t as scary as I was probably imagining.

A big thank you my fellow PhD candidates of the Cybersecurity group, past and present—Aksel, Arwa, Boy, Cécile, Donald, Elsa, Evi, Fieke, Gerbrand, Hugo, Lorenz, Radu, Ronak, Sandra, Swaathi, Szu-Chun, Veerle, and Xander—for making this group truly feel like a team. I also want to thank past members of the team—Arman, Kate, Natalia, Qasim, and Ugur—for all the guidance I received when I was still starting out. Special thanks to Natalia for the initiation rituals and subsequent cultural indoctrination.

Elsa, I think I’ve told you this already many times, but our morning gossip session was my favorite part of the day. I really do believe that set the rest of the day up for success. It also helped me practice my Spanish, something my soul needed deeply. I never thought that would be part of our daily routine when I first met you during your first colloquium when I was still browsing the PhD team. Your many cat-eye requests made me feel useful, and I'm happy we were able to help each other throughout our journey.

Boy, thank you for all of our fantastic conversations about the spirit, quantum entanglement, the meaning of life, and all the other hippie stuff. You have a talent for no-nonsense, straight-to-the-point encouragement, and without your inspiring words I would surely still deadlift unacceptable numbers. Though the time you spent at the faculty wasn't that long, you made the days there much brighter.

Arwa, I'm glad that you found our group after spending a year doing your PhD somewhere else. Thank you for being so funny all the time. Also, many thanks for helping me prepare for my presentation in Nagasaki and for road tripping with me around Japan, even though you left (i.e., abandoned me) halfway through. Still, you made up for that by preventing my deportation later, so I'll let it slide.

María, thanks a ton for your friendship and kindness through the good times and the struggles. It almost makes up for you completely ignoring my existence until you heard we shared a country of birth. Just kidding, of course. Thank you for organizing all the dinners, get-togethers, and *Velitas* nights. You make everyone feel like family.

Sandra, thank you for bringing a hefty dose of Latin energy to the team. Nobody realizes that they need that until they've experienced it. I'm relieved that you arrived when you did, because I was in need of another language buddy. Too bad you left our city, but I suppose you can't have everything in life. I look forward to many more salsa parties, bouldering sessions, and game nights. Except for Exploding Kittens.

Lorenz, thanks for your friendship and hospitality. Always fun to find somebody with a similar fascination for the land of the rising sun, and even more so someone who spent time there around the same period that I did. It sometimes surprises me how small the world can be.

Szu, thank you for allowing me to continue my morning ritual of bothering people in your office, and thanks for not hating me when Sandra and I go off on our conversations in Spanish in front of you while you're busy working. Luckily for us, the first impression you had of Michel, Arwa and I in Nagasaki wasn't enough to scare you off. You've integrated well into the team, and I wish you the best of luck in these years to come.

Anni, you were one of my first TPM friends, and I want to thank you for welcoming me into your Thursday social club. Your friendship got me through some tough times, especially during the lockdown. Thank you for the celebratory Schnapps, and I'm looking forward to many more. So get on with your thesis and defense! That way I can finally print out your booklet of quotes, too.

To my esteemed and illustrious co-authors Natalia and Simon, I want to thank you for your outstanding contributions to what probably were the most backbreaking 15 pages of text I've ever written. Without you, that paper would've never gotten off the ground, let alone win a Distinguished Paper Award. Sometimes I feel like Leonardo DiCaprio when I see that Oscar standing on my nightstand. And I hope you do too.

Swaathi and Veerle, thank you for being the coolest of office mates and letting me put the heater in the office on its highest setting. I always enjoyed our conversations about love, the state of the world, and the youth nowadays, even though we most likely should've spent that time working.

Yury, thank you for all of your enlightening feedback, help with interview transcriptions, and career advice. There's been a notable change for the better ever since you joined the team, and I'm sure it'll continue in that direction with your help.

Savvas, I'm sure you know what I'm going to write here. Thank you for showing me that academic presentations are allowed to be fun, instead of exclusively informative. And even though sometimes the audience can't appreciate the selection of memes, at least I, as the presenter, am enjoying my time on the stage.

Rolf, thanks a ton for all the pep talks and guidance navigating the confusing and complicated worlds of TPM and TU Delft. Your talent of bending the rules and (mis)using technicalities to achieve a goal or obtain what you want is incredibly inspirational and something I will strive to emulate in my own endeavors.

Ruud, Sanne, and Erik, I am very grateful to you for making this entire project possible in the first place. Whether it was answering my tedious questions, providing valuable feedback, or handling endless bureaucratic processes so I didn't have to, your contributions to this thesis were more than substantial.

Japhet, my man, thank you for your friendship and support, during both the highs and the lows. I have our Coconuts excursions (and your gift of bouncer-whispering) to thank for much of my energy and mental stability. Wherever life may take us, know that you can count on me for whatever you need.

To Esther, here's a fun fact: the last two chapters of this thesis were fueled entirely by sushi, bagels, quadruple cappuccinos, and chamomile tea. And, of course, I have you to thank for that. Working from home was never as fun or productive as it was with you keeping me company. Even with your horrible romcoms playing in the background.

To my brother Daniel, thank you for always being interested in the esoteric work that I do. I appreciate every question you ask to try to make sense of it. They definitely help, because sometimes even I barely understand what I'm supposed to be doing. You keep telling everybody I'm the smartest guy you know, and now I've actually got to live up to that standard.

Tía Lynn, all those years of you calling me Bill are finally paying off. I'm glad I'm living up to your standards as well now! Thank you for all the love and support you send from all the way across the ocean. I can feel it every day.

A mi querida abuelita, muchísimas gracias por tu amor y tu apoyo. Este logro también te pertenece a tí, ya que fuiste tú la persona que me convenció a iniciar este doctorado con tus palabras sabias: «Al final lo único que a uno le queda es el conocimiento».

To my mom and dad, Olga and Michel, thank you for supporting me and my goals, something you've been doing for my entire life. You made this all possible, either through you believing in me, or through the food deliveries during crunch time when deadlines were approaching. Simple words cannot describe my gratitude. I dedicate this achievement to you!

Mathew Vermeer
Rotterdam, January 2024

AUTHORSHIP CONTRIBUTIONS

This thesis is founded on four academic papers, each a product of teamwork with several co-authors. Throughout these research projects, I benefited from insightful input and diverse contributions made by my collaborators. In the following paragraphs, I will detail the specific contributions of each co-author to every study.

In the first study (Chapter 2, my co-authors Jonathan West, Alejandro Cuevas, Shuonan Niu, Nicolas Christin, Michel van Eeten, Tobias Fiebig, Carlos Gañán, and Tyler Moore, and I were all involved to varying degrees in the framework design, literature search, and systematization of asset discovery techniques. Nicolas Christin, Michel van Eeten, Tobias Fiebig, and Tyler Moore contributed to the introduction, framework definitions, and case studies. Tobias Fiebig assisted in the writing of the concluding remarks and the trimming down of the text to conform to the venue's page limitations.

For the second study (Chapter 3, my co-authors Michel van Eeten and Carlos Gañán both contributed to the structure and contents of the introduction and conclusion. They also assisted in the sharpening of the discussion and the improvement of the general structure of the numerous draft versions of the paper. Data analysis and interviews were performed by me.

In the third study (Chapter 4), Natalia Kadenko assisted in the carrying out of numerous interviews. I was responsible for the design of the interview protocol. Together we transcribed and coded all of the interviews. Natalia Kadenko also contributed to portions of the Results section. Simon Parkin contributed to the collection of background literature, analysis of the findings, and improvements to the paper's argumentation and discussion. Michel van Eeten and Carlos Gañán helped improving the paper's overall structure and formatting.

For the fourth and final study (Chapter 5), choices regarding data collection, model selection and evaluation methodology were the result of in-depth discussions with my co-authors Michel van Eeten and Carlos Gañán. Michel van Eeten also contributed by improving and restructuring the introduction of the paper. Carlos Gañán wrote the conclusion and abstract, as well as proofread, corrected mistakes, and polished the final text.

I have been fortunate throughout all of these studies to have received the help and guidance from my many co-authors. To my co-authors, I am immensely grateful for your contributions that are undoubtedly part of the foundation of this thesis. In particular, I would like to highlight the role of my promotors Michel van Eeten and Carlos Gañán, whose aid, from the first study to the last, have allowed me to fly halfway across the globe to talk about my work.

List of Publications

- **Vermeer, M.**, West, J., Cuevas, A., Niu, S., Christin, N., Van Eeten, M., Fiebig, T., Ganán, C. and Moore, T. 2021. SoK: A Framework for Asset Discovery: Systematizing Advances in Network Measurements for Protecting Organizations. In Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P '21), Vienna, Austria, 2021, 440-456, https://doi.org/10.1109/EuroSP51992.2021.00037
- **Vermeer, M.**, van Eeten, M., Gañán, C. 2022. Ruling the Rules: Quantifying the Evolution of Rulesets, Alerts and Incidents in Network Intrusion Detection. In Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security (ASIA CCS '22). Association for Computing Machinery, New York, NY, USA, 799–814. https://doi.org/10.1145/3488932.3517412
- **Vermeer, M.**, Kadenko, N., van Eeten, M., Gañán, C., and Parkin, S. 2023. Alert Alchemy: SOC Workflows and Decisions in the Management of NIDS Rules. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23). Association for Computing Machinery, New York, NY, USA, 2770–2784. https://doi.org/10.1145/3576915.3616581
- *Under submission at IEEE S&P '25:*
  Vermeer, M., van Eeten, M., Gañán, C. 2024. Network Whispers: Deciphering Incident Predictions with External Signals.

DATASETS

Table 2: Dataset availability

Publication	Dataset(s)
Vermeer, M., van Eeten, M., Gañán, C. 2022. Ruling the Rules: Quantifying the Evolution of Rulesets, Alerts and Incidents in Network Intrusion Detection. In Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security (ASIA CCS '22). Association for Computing Machinery, New York, NY, USA, 799–814. https://doi.org/10.1145/3488932.3517412	MSSP NIDS alert and incident data, even anonymized, cannot be shared.
Vermeer, M., Kadenko, N., van Eeten, M., Gañán, C., and Parkin, S. 2023. Alert Alchemy: SOC Workflows and Decisions in the Management of NIDS Rules. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23). Association for Computing Machinery, New York, NY, USA, 2770–2784. https://doi.org/10.1145/3576915.3616581	Interview transcripts, even anonymized, cannot be shared.
Under submission at IEEE S&P '25: **Vermeer, M.**, van Eeten, M., Gañán, C. 2024. Network Whispers: Deciphering Incident Predictions with External Signals..	MSSP NIDS alert and incident data, even anonymized, cannot be shared. Commercial IPv4 scan data, IP-to-organization mappings, and abuse datasets can be acquired through their respective vendor. BGP routing data is publicly available.

About the author

Mathew Vermeer (1994) was born in Cali, Colombia. He enrolled at Delft University of Technology in 2012. After completing his minor at Tokyo Institute of Technology, he received his BSc degree in Computer Science and Engineering in 2015. In 2015 he again joined Delft University of Technology to pursue his MSc degree in Computer Science. Having started his own company in software development in 2013, he was involved in various software engineering projects while he pursued his MSc degree. He obtained his MSc degree in 2019 with an additional specialization in cyber security. His thesis examined the flaws in the design, training, and evaluation of "state-of-the-art" machine learning-based network intrusion detection systems, and how to properly evaluate their performance and effectiveness in real-world environments.

In the spring of 2019, he rejoined Delft University of Technology as a PhD candidate. His research was embedded in the CyRIE project - a DHS-funded project in collaboration with Carnegie Mellon University and The University of Tulsa. The project focused on studying the feasibility of security incident prediction and creation of outcome-based cyber risk metrics for organizations. In this project, he investigated how to optimally detect an organization's Internet exposure, and the technical and organizational processes that govern how internal network incidents are managed by organizations. Additionally, he studied manners in which an organization's internal security incidents can be predicted using exclusively external Internet exposure. During his time as a PhD candidate at the university, he supervised two MSc students throughout the duration of their thesis projects on related topics.