Analyzing the CoAP DDoS Amplification Attack Ecosystem: A Honeypot Study

Deploying the First CoAP-Specific Amplification Honeypot

Anton Kalpakchiev[1]

Supervisor: Dr. Harm Griffioen[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Anton Kalpakchiev
Final project course: CSE3000 Research Project
Thesis committee: Dr. Harm Griffioen, Mitchell Olsthoorn

Abstract

DDoS amplification attacks exploit connectionless protocols to overwhelm victims with traffic far exceeding the attacker's own bandwidth. The Constrained Application Protocol (CoAP), designed for IoT devices, has emerged as an amplification vector with over 1.7 million exposed reflectors and an amplification factor of approximately 34x, yet no prior work has studied CoAP amplification through honeypot-based observation. We hypothesize that CoAP's extreme reflector instability - 80% of IP addresses change within two weeks - imposes a higher sophistication floor on adversaries compared to previously studied protocols. To evaluate this, we deploy the first CoAP-specific amplification honeypot with eight protocol configurations testing blockwise transfer at four block sizes and rate limiting at three thresholds across 18 IP addresses for 23 days. The scanning traffic reveals behaviors undocumented in prior honeypot research: multi-phase CBOR fingerprinting that distinguishes genuine Eclipse Californium devices from other implementations, sustained reliability validation over repeated scanning cycles, and efficient list management where probing ceases after device cataloguing. Notably, our honeypot was likely filtered out after failing this fingerprinting test - a finding that contrasts sharply with prior work on other protocols, where attackers were largely indifferent to honeypot responses. No amplification attacks were observed, preventing empirical evaluation of the tested defenses. However, our findings provide the first evidence that CoAP's structural properties demand higher adversary sophistication than previously studied amplification protocols, and suggest that threat models for CoAP must account for fingerprinting-capable actors.

1 Introduction

DDoS amplification attacks remain a persistent threat to internet infrastructure, exploiting connectionless protocols to reflect and amplify traffic toward victims through IP spoofing. While traditional amplification protocols such as DNS and NTP have been extensively studied, the Constrained Application Protocol (CoAP) [11] has emerged in recent years as a new amplification vector. CoAP is a lightweight protocol designed for resource-constrained IoT devices, and its lack of mandatory authentication makes it susceptible to spoofing abuse. In 2018-2019, NETSCOUT observed the first CoAP amplification attacks in the wild, reporting an average amplification factor of approximately 34x across over 315,000 exposed devices [6] - a number that has since grown to over 1.7 million [1]. Nearly all attack traffic targeted the `/.well-known/core` resource discovery endpoint - a small request of roughly 23 bytes yielding responses of 700+ bytes. Moreover, an analysis of TU Delft's network telescope dataset in 2026 confirms the threat still exists: 365,643 well-formed CoAP packets were observed in a single day, originating from multiple distinct actors.

Prior honeypot studies on amplification protocols - Rossow [9] (14 protocols), Krämer et al. [4] (AmpPot, 21 instances), and Griffioen et al. [3] (549 honeypots) all found largely unsophisticated adversary behavior: attacks arrived within hours of deployment, scanners used the same payload as attacks, most actors did not detect honeypots, and stale reflector lists were used for months [3]. However, these findings are based on protocols with stable, server-based reflector pools. CoAP's reflector population is fundamentally different: more than 80% are mobile phones running QLC Chain (a defunct blockchain app), not servers [6, 12]. Therefore, 80% of reflectors change their IP within 2 weeks [6]. We hypothesize that this structural constraint requires higher actor sophistication to successfully execute the lifecycle of a CoAP attack compared to previously studied protocols.

Moreover, the IETF recommends blockwise transfer as a mitigation for CoAP amplification [5, 2], and rate limiting has been proposed for other amplification protocols [4]. However, both defenses require tradeoffs: small block sizes degrade performance for legitimate device communication, while aggressive rate limiting risks throttling real clients. Without empirical data on what thresholds are safe, CoAP developers must make these tradeoffs blindly.

To evaluate our hypothesis that CoAP has high adversary sophistication requirements and to understand how effective different blockwise transfer and ratelimitting thresholds are, we deploy a CoAP honeypot with eight different protocol configurations (four blockwise transfer variants, three rate limiting thresholds, one baseline) across 18 IP addresses for 23 days. We investigate two research questions:

1. **RQ1:** What actors scan for CoAP reflectors, how do they operate, and how do their techniques compare to previously documented adversary behavior?
2. **RQ2:** How effective are blockwise transfer and rate limiting at mitigating CoAP amplification attacks?

The remainder of this paper is organized as follows. Section 2 provides background on CoAP and the existing DDoS amplification attack research. Section 3 describes the experiment's methodology - the honeypot design and deployment. Section 4 presents the results. Section 5 discusses responsible research. Section 6 discusses the results. Section 7 concludes and outlines future work.

2 Background & Related Work

2.1 DDoS Amplification Attacks and Honeypots

In an amplification attack, the attacker spoofs the victim's IP address and sends a small request to a UDP-based reflector. The reflector, unable to verify the source address, sends its (larger) response to the victim [9]. The effectiveness of this technique is measured by the Bandwidth Amplification Factor (BAF), defined as the ratio of response payload size to request payload size [9]. A BAF of 34x, for example, means that a single 23-byte request generates a 782-byte response directed at the victim.

To study this phenomenon without operating real vulnerable infrastructure, honeypots are deployed - servers that intentionally emulate vulnerable services to attract and observe scanning and attack traffic [4]. By analyzing the traffic these honeypots receive, researchers can characterize adversary behavior, identify scanning patterns, and evaluate potential defenses.

2.2 Related Work and Open Gaps

Amplification attacks have been extensively characterized for established protocols. Rossow [9] measured the amplification potential of 14 protocols, finding BAFs of 556.9x for NTP, 54x for DNS, and 30.8x for SSDP. Krämer et al. [4] deployed AmpPot and observed over 1.5 million attacks across 9 protocols. Griffioen et al. [3] scaled to 549 honeypots covering

6 protocols. Together, these studies produced detailed threat profiles including adversary behavior, scanning patterns, and attack tactics.

These protocols share common characteristics: server-based reflectors with static IPs, well-understood exploitation tooling, and large stable reflector pools. In this context, Griffioen proposed a six-phase attack chain model (Capability Development → Infrastructure Reconnaissance → Target Reconnaissance → Weaponization → Testing → Execution) and found that most adversaries were unsophisticated - they used the same payload for scanning and attacks, did not detect honeypots, and reused stale reflector lists for months [3].

However, CoAP's reflector population is structurally unlike any previously studied protocol. CoAP emerged as an attack vector in 2018, with real attacks reaching 320 Gbps [6]. The IETF has since published drafts analyzing CoAP amplification scenarios and recommends blockwise transfer as mitigation [5, 2]. Yet no honeypot research has studied CoAP. Its reflectors are mobile phones rather than servers, thus 80% of IPs churn within two weeks [6].

Effective security requires accurate threat profiles. Griffioen's attack chain model describes adversary behavior for stable-reflector protocols, but it is unclear whether this model applies to CoAP given its fundamentally different infrastructure. If CoAP's structural constraints produce different adversary behavior, defenses built on the assumptions of existing models may be inadequate. Understanding whether and how CoAP adversaries differ is therefore a prerequisite for building effective CoAP-specific defenses.

2.3 CoAP Protocol Specifics

CoAP [11] is a UDP-based protocol operating on port 5683, designed for resource-constrained IoT devices. It defines two request message types: Confirmable (CON), which requires an acknowledgement, and Non-confirmable (NON), which follows a fire-and-forget messaging approach. Responses are carried in ACK or RST messages. Common request codes include 0.01 (GET) and 0.02 (POST), with response codes such as 2.05 (Content), 4.04 (Not Found), and 4.05 (Method Not Allowed). Messages carry options including Uri-Path (option 11), Content-Format (option 12), and Block2 (option 23).

The `/.well-known/core` endpoint [10] provides resource discovery, returning a CoRE Link Format listing of all device resources (see Appendix A for an example). This endpoint is the primary amplification vector: a roughly 21-byte request yields a response of approximately 700 bytes.

2.4 CoAP Reflector Population

Approximately 80% of all CoAP reflectors are located in China [6]. The dominant device among them is QLC Chain (formerly Qlink), a blockchain-based WiFi/VPN-sharing app that ran on Android phones. The app used CoAP over Eclipse Californium, a Java CoAP framework, for peer-to-peer communication, requiring public reachability on port 5683. Of 315,594 Chinese CoAP devices scanned, all but 382 returned QLC Chain responses [6, 12]. The project is now defunct (website offline since November 2024), meaning the vast majority of the global reflector pool is unmaintained zombie infrastructure with no path to remediation.

2.5 Challenges of CoAP Attacks

The most distinctive property of CoAP's reflector population is its instability. NETSCOUT reports that "comparing scans performed two weeks apart, only 20% of the addresses appear in both scans" [6]. This is because CoAP reflectors are predominantly mobile phones behind cellular CGNAT, in contrast to NTP, DNS, and Memcached reflectors which run on servers with static IP addresses.

Despite having more reflectors than NTP and Memcached combined (1.7 million vs approximately 33,000), CoAP produces only 163,000 attacks compared to DNS's 14 million [1]. These structural challenges may contribute to this disparity, suggesting a higher skill floor for effective CoAP exploitation - a hypothesis we evaluate through our honeypot deployment.

2.6 Proposed Mitigations for CoAP Amplification

The IETF recommends blockwise transfer as the primary mitigation for CoAP amplification [5, 2]. The principle is straightforward: if the server splits its response into small blocks, a spoofed request only triggers the first block. Subsequent blocks require follow-up requests that an attacker using a forged source address cannot send. This theoretically reduces the amplification factor from approximately 34x to as low as 1.2x, depending on block size.

Rate limiting per source IP is a complementary defense, used in other amplification protocols such as DNS RRL [9]. However, both defenses involve tradeoffs: blockwise transfer with small block sizes degrades performance for genuine device communication, while overly aggressive rate limiting risks throttling legitimate clients. The optimal configuration that balances security with usability remains unexplored.

Neither defense has been empirically evaluated against real CoAP attackers. Krämer et al. [4] identified rate limiting threshold evaluation as future work, a gap that remains unfilled.

3 Methodology

3.1 Honeypot Design

The goal of our honeypot is to emulate a vulnerable CoAP reflector that responds to `/.well-known/core` with a realistic CoRE Link Format payload, while varying protocol implementations (ratelimitting and blockwise configuration specifically) across IPs to enable comparison.

The honeypot is implemented as a single Go binary. It serves a static CoRE Link Format response listing 15 resources that emulate a typical IoT device (698-byte payload, see Appendix A). GET requests to `/.well-known/core` receive a 2.05 Content response; all other paths return 4.04 Not Found, and non-GET methods return 4.05 Method Not Allowed. Both CON and NON message types are supported per RFC 7252 [11].

3.2 Protocol Variants

Each public IP address is deterministically mapped to a specific variant configuration in the application layer. Based on the destination IP of each incoming packet, the packet is routed to the corresponding protocol variant. From the perspective of external clients, each IP appears to be a separate CoAP device with its own behavior - but all are served by a single process.

We test eight variants across two defense dimensions. Table 1 summarizes the configurations.

Table 1: Honeypot variant configurations. BAF computed for a 23-byte request against the 698-byte payload.

Variant	Block size (B)	Response Size (B)	BAF	Rate limit	IPs
baseline	–	707	30.7x	500 reqs/min	4
block-smallest	16	27	1.2x	500 reqs/min	2
block-small	128	139	6.0x	500 reqs/min	2
block-medium	256	267	11.6x	500 reqs/min	2
block-large	512	523	22.7x	500 reqs/min	2
ratelimit-5	–	707	30.7x	5 reqs/min	2
ratelimit-20	–	707	30.7x	20 reqs/min	2
ratelimit-100	–	707	30.7x	100 reqs/min	2

The blockwise variants test the IETF's recommended mitigation at varying granularity [5, 2]. The rate limiting variants test the complementary defense identified as an open gap by AmpPot [4]. The baseline provides a control with no defenses.

Rate limiting is implemented using a leaky bucket algorithm (GCRA) keyed by source IP address (the address of the potential victim). Excess requests are silently dropped with no error response sent. Rate limit variants serve full unblocked responses, so each variant tests one defense dimension in isolation. All variants, including blockwise and baseline, additionally have a 500 req/min (42 MB/hour bandwidth) safety cap to prevent the honeypot from being used for real amplification attacks.

3.3 Deployment

The honeypot was deployed on a single host at on an academic network, with 18 IP addresses across two network interfaces: eth0 (145.220.183.35, 1 IP) and enp6s20 (17 IPs in the 145.220.231.x and 172.110.0.x ranges).

A deployment limitation affected the experiment: Linux's default routing sent all responses via eth0 regardless of which interface received the request, causing responses to enp6s20 IPs to carry eth0's source address. As a result, only 145.220.183.35 (mapped to ratelimit-100) responded with the correct source IP. The implications of this are discussed in Section 6.

The experiment ran for 23 days (May 28 – June 20, 2026). Zero downtime was confirmed via a cron heartbeat log.

3.4 Data Collection

Three data sources were used. Prometheus scraped honeypot metrics every 15 seconds, collecting 49 metrics including per-variant request counts, response codes, rate-limited drops, and response sizes. tcpdump captured packets on eth0 only, recording 3,626 outbound responses and 277 inbound requests with payload to 145.220.183.35. Application logs were minimal by design (startup messages and unexpected errors only), with all operational data routed through Prometheus.

A limitation of this setup is that inbound traffic to the enp6s20 IPs was not captured in the pcap. However, Prometheus provides complete aggregate coverage across all 18 IPs.

The full source code, pcap capture, and Prometheus data are available in the TU Delft research repository for inspection and reproducibility.

4 Results

4.1 Scanner Ecosystem Composition

Over the 23-day experiment, we observed 37 unique scanner IP addresses. Approximately 70% were identifiable as research scanners belonging to organizations such as Censys, Shodan, Palo Alto Networks, NETSCOUT, and Hurricane Electric. For comparison, Griffioen et al. found 30–37% research scanners in their 2019 deployment [3].

One abuse-flagged actor was identified: Intelligence Hosting LLC (AS400328, Netherlands). While the scanning IPs themselves had limited abuse history, neighboring IPs in the same allocation have between 14,000 and 52,000 abuse reports on AbuseIPDB.

Additionally, five previously unseen scanners arrived within a 90-minute window on day 5 of the experiment, each exhibiting identical behavior: 30–45 repeated probes over approximately 15 hours. This pattern may suggest a coordinated campaign or consumption of a shared list of CoAP reflectors that was updated around that time.

4.2 Probe Types Observed

The honeypot received 3,628 total requests, with 3,626 outbound responses captured in the pcap. Response code analysis of the full sample (n=3,626) shows:

- 87.4% received a 2.05 Content response (valid GET `/.well-known/core`)
- 10.9% received a 4.04 Not Found response (GET without correct Uri-Path)
- 1.7% received a 4.05 Method Not Allowed response (non-GET methods)

From the 277 inbound payloads captured on 145.220.183.35 (the only IP with full packet capture), we identified three non-standard probe types:

- **CBOR probe** (13 bytes): a GET request with Content-Format set to CBOR, a 2-byte payload, and no Uri-Path option. 13 packets, all from Intelligence Hosting

LLC. This payload is documented in AMP-Research [7] as a fingerprinting probe for Eclipse Californium, the CoAP framework used by QLC Chain devices. A genuine Californium instance returns a 2.05 Content response (458 bytes), while our honeypot returns 4.04 Not Found (8 bytes) - allowing the sender to distinguish real reflectors from non-standard implementations.

- **Blockwise probe** (23 bytes): a GET request to `/.well-known/core` with a Block2 option (SZX=2). 4 packets from Shodan.
- **Minimal probe** (4 bytes): a header-only GET with no options or payload. 4 packets.

Notably, 12.6% of responses in the full sample were non-standard (4.04 or 4.05), compared to only 6% in the captured inbound traffic on 145.220.183.35. This suggests the 145.220.231.x IPs (which had source address mismatch issues) received proportionally more non-standard probes.

4.3 Request Distribution Across Variants

Table 2 shows the total requests received by each variant over the 23-day experiment. Normalizing by the number of IPs per variant, scanning traffic was approximately uniform (about 210 requests/IP) across all configurations. Notably, no rate-limited drops, malformed packets, or write errors were recorded for any variant.

Table 2: Total requests per variant over the 23-day experiment. Requests/IP normalizes for variants with different numbers of IPs.

Variant	IPs	Total Requests	Requests/IP
baseline	4	840	210.0
block-smallest	2	420	210.0
block-small	2	424	212.0
block-medium	2	417	208.5
block-large	2	418	209.0
ratelimit-5	2	414	207.0
ratelimit-20	2	420	210.0
ratelimit-100	1	275	275.0
Total	**17**	**3,628**	

The ratelimit-100 variant was allocated two IPs (145.220.183.35 and 172.110.0.2), but 172.110.0.2 is in a different subnet (172.110.0.x vs 145.220.231.x) and received zero requests — scanners targeting the 145.220.x.x range never discovered it. Furthermore, 145.220.183.35 was indexed by Shodan early in the experiment, after which scanning effectively stopped (2 probes in 18 days). This explains the higher initial per-IP count compared to the non-indexed IPs (about 210 requests/IP), which continued receiving probes throughout the experiment.

4.4 Scanning Timeline and Behavioral Patterns

The honeypot was first discovered within 17 hours of deployment. Research scanners trickled in over days 1–4, followed by a validation burst on day 5 (the 90-minute arrival cluster

described above). After this initial period, scanning of the indexed IP effectively ceased.

Intelligence Hosting LLC exhibited a distinctive scanning protocol: 13 cycles over 4.5 days, with each cycle consisting of a standard probe followed 30–45 minutes later by a simultaneous standard and CBOR probe. These cycles repeated approximately every 8 hours.

The most striking pattern was the post-cataloguing silence. After Shodan indexed 145.220.183.35 with port 5683, this IP received only 2 probes over the remaining 18 days, compared to 2,654 requests to the non-indexed IPs during the same period.

4.5 Absence of Attacks

No amplification attacks were observed during the entire 23-day collection period. All 3,628 requests were scanning probes - none involved spoofed source IPs or targeted a third-party victim. This contrasts with prior honeypot studies on other protocols, where attacks typically arrived within hours of deployment [3, 4]. The implications of this result are discussed in Section 6.

5 Responsible Research

To prevent hurting victims during real attacks, the honeypot applies per-victim-IP rate limiting. The baseline configuration (500 requests per minute) produces at most approximately 42 MB/hour of reflected traffic per victim IP. While not fully negligible, this volume is insufficient to cause meaningful denial of service against modern internet-connected hosts. The baseline rate limit is intentionally set relatively high to serve as a control condition - lower thresholds (5, 20, 100 requests per minute) are the experimental conditions being evaluated, so the baseline must be higher by definition.

Furthermore, generative AI has been used in an ethical manner. It has been used to assist with implementation of the honeypot and to provide feedback on the writing, while maintaining the essence of what was written in the original text. The main ways, in which AI helped included pointing out areas of the text that lacked context or sources, simplifying phrasing, and looking up sources for claims, which were later manually verified.

6 Discussion

6.1 Why Were No Attacks Observed?

No amplification attacks were observed during the 23-day collection period. This contrasts with prior honeypot studies on other protocols, where attacks typically arrived within hours of deployment [3, 4]. Several factors likely contributed to this result.

Most directly, our honeypot failed the CBOR fingerprinting probe sent by Intelligence Hosting LLC. As described in Section 4, this probe produces a differential response between genuine Californium devices and other CoAP implementations. Our honeypot returned 4.04 Not Found instead of the expected 2.05 Content, which would have identified it as a

non-genuine device. If this actor maintains a curated reflector list, our honeypot was likely not included after failing this test.

Additionally, an implementation error led to 17 of 18 IPs responding with an incorrect source address, meaning scanners received replies from a different IP than the one they probed. Only one IP (145.220.183.35) responded correctly and was indexed by Shodan. However, reduced deployment scale alone is unlikely to explain the absence of attacks. Griffioen et al. found that attacks arrived within hours of deployment even on individual honeypot IPs, and that attackers were largely indifferent to the responses they received - even honeypots that explicitly identified themselves as fake still attracted attack traffic [3]. This suggests that a single correctly responding IP should have been sufficient to observe attacks, had our honeypot not been filtered out during reconnaissance. Taken together, fingerprint-based filtering appears to be the primary explanation for the absence of attacks, rather than deployment scale.

As a result, we cannot empirically evaluate the effectiveness of blockwise transfer or rate limiting defenses against real attackers, leaving RQ2 as future work.

6.2 Is High Actor Sophistication Required?

We hypothesized that CoAP's structural constraints - particularly its extreme reflector instability - require more sophisticated scanning behavior compared to protocols with stable reflector pools. We evaluate this by mapping observed behaviors to the structural constraints that may explain them.

Firstly, Intelligence Hosting LLC probed the same IP 13 times over 4.5 days, which may point to a reliability validation strategy: verifying that a reflector remains online before committing it to a list that decays rapidly. For comparison, Griffioen et al. found that unsophisticated actors reused stale NTP and SSDP lists for months and attacks still landed [3] - an approach that is only viable because server IPs are stable. That strategy cannot work efficiently for CoAP.

Since the vast majority of real CoAP reflectors are QLC Chain devices running Eclipse Californium, an actor seeking quality reflectors needs a way to verify that a device is genuinely running this framework. The CBOR probe serves exactly this purpose: it exploits a difference in how Californium handles an unexpected Content-Format option. A genuine Californium instance parses the CBOR payload and returns a 2.05 Content response (458 bytes), while any other implementation - including our honeypot - returns 4.04 Not Found (8 bytes) [7]. This is a binary pass/fail test that reliably identifies genuine reflectors.

This is a notable finding. In Griffioen et al.'s study of six protocols, the closest equivalent was DNS BIND version fingerprinting, where scanners simply asked the server to report its own software version [3]. The CBOR probe is qualitatively different: rather than asking the device to self-identify, it tests edge-case implementation behavior that only the real framework exhibits. This requires knowledge of the target ecosystem - specifically, that the reflector population runs Californium, and how Californium handles CBOR-encoded requests to non-existent resources. Whether the intent is specifically to filter out honeypots or simply to select for quality is unclear, but either way it demonstrates deep understanding of the CoAP reflector ecosystem, consistent with the hypothesis that CoAP's structural constraints demand higher adversary sophistication compared to traditional amplification

protocols.

These probing behaviors stand in contrast to the spray-and-pray approach documented in prior work. Griffioen et al. found that for protocols like NTP and SSDP, attackers typically used the same payload for both scanning and attacks - there was no distinct reconnaissance phase, no fingerprinting, and no quality filtering [3]. In our observations, scanning and reconnaissance are clearly differentiated: Intelligence Hosting LLC uses a multi-phase protocol (standard probe, then CBOR fingerprint), and Shodan sends Block2 probes with SZX=2 to test whether a device supports blockwise transfer [2]. The Block2 probe is particularly significant because it tests a protocol-specific capability that directly affects amplification potential - a device that supports blockwise transfer is a less effective reflector. This level of protocol-aware reconnaissance has no equivalent in the scanning behavior documented for older amplification protocols.

Another indication of actor sophistication is that after one of our IPs was catalogued in Shodan, scanning effectively stopped: 2 probes in 18 days, compared to 2,654 requests to non-indexed IPs. This pattern is consistent with efficient list management - once a device is catalogued, re-probing it is wasteful - and aligns with the sophisticated actor profile described by Griffioen et al. [3]. However, with only one correctly responding honeypot IP, we cannot rule out other explanations for this disparity.

Furthermore, the scanning patterns we observe may suggest that Phases 2 (Infrastructure Reconnaissance) and 5 (Testing) of Griffioen's six-phase model have merged. Each of Intelligence Hosting LLC's 13 cycles combines discovery, validation, and fingerprinting in a single pass, rather than the sequential scan-then-test approach Griffioen described. However, our honeypot was likely filtered out during this merged phase, preventing observation of later phases (Weaponization, Execution).

Finally, public scanning infrastructure may serve as a force multiplier. Shodan tests Block2 capability and publishes results publicly, which could allow adversaries to query Shodan instead of scanning themselves, effectively decoupling reconnaissance from exploitation. Reflector lists are also known to be tradeable commodities between actors [3].

Taken together, these observations provide consistent evidence that CoAP scanning demands higher adversary sophistication than documented for previously studied amplification protocols. Repeated validation cycles, ecosystem-specific fingerprinting, efficient list management, and protocol-aware capability testing all represent behaviors absent from the spray-and-pray approaches that characterize NTP, DNS, and SSDP scanning [3].

6.3 Limitations

Our deployment consisted of one effectively working IP over 23 days at a single location (Netherlands, academic network). This is substantially smaller than Griffioen et al.'s 549 honeypots across 5 cloud providers over 3 months [3]. Furthermore, approximately 80% of CoAP reflectors are located in China [12], and our European deployment may not reflect scanning patterns directed at that primary reflector population.

Moreover, the source address mismatch on 17 of 18 IPs reduced the effective sample. The pcap was captured on the primary interface only, limiting full payload analysis to 277 inbound packets. However, not all request information was lost, as Prometheus metrics provide complete aggregate coverage across all 18 IPs.

CoAP was not included in Griffioen's study, which covered NTP, DNS, SSDP, CharGen, RIP, and QOTD. Comparing our findings to theirs involves comparing across different protocols, time periods, and deployment scales.

Additionally, we cannot definitively prove that Intelligence Hosting LLC is adversarial. The abuse reports on neighboring IPs are suggestive but not conclusive.

Finally, there may be alternative explanations for the observed behavior. The sophistication we observe may reflect CoAP's niche status attracting inherently more capable actors, rather than an ecosystem-wide evolution in adversary behavior. However, either interpretation supports the hypothesis that CoAP's structural properties shape scanning behavior.

7 Conclusions and Future Work

We hypothesized that the challenges in executing a CoAP attack - most notably, extreme reflector instability (80% IP churn in two weeks) impose a higher sophistication floor for scanning and exploitation compared to traditional, previously-studied amplification protocols. To evaluate this, we deployed the first CoAP-specific amplification honeypot with eight implementation variants (four blockwise transfer sizes, three rate limiting thresholds, one baseline) across 18 IP addresses for 23 days.

Firstly, our observations show that the CoAP scanning ecosystem exhibits behaviors not documented in prior honeypot research on other protocols. We observed multi-phase fingerprinting using CBOR probes that produce differential responses between genuine Californium devices and non-standard implementations, sustained reliability validation (13 scanning cycles over 4.5 days), and efficient list management where scanning ceases after a device is catalogued. These behaviors are consistent with the structural pressures CoAP imposes: when 80% of reflectors churn within two weeks, actors must verify reliability and filter non-genuine devices before committing resources to exploitation.

However, no amplification attacks were observed during the 23-day collection period, preventing empirical evaluation of the blockwise transfer and rate limiting defenses. Theoretical amplification factor reductions were confirmed (30.7x baseline down to 1.2x with 16-byte blocks). Scanning traffic was uniform across all variants - defenses did not differentially attract or repel scanners.

More broadly, our findings challenge the assumption from prior work that amplification adversaries are largely unsophisticated [3]. The CoAP scanning ecosystem we observed operates at a qualitatively different level: actors deploy ecosystem-specific fingerprinting, validate reflector reliability over multiple cycles, and leverage protocol-aware capability testing - none of which have equivalents in the scanning behavior documented for NTP, DNS, or SSDP. Threat models and defenses for CoAP must account for these fingerprinting-capable adversaries who actively distinguish genuine reflectors from honeypots and other non-standard implementations. Nevertheless, our findings are suggestive but not definitive: a single deployment location far from the primary reflector population (80% of which is in China [12]), one effectively working IP, and a 23-day window limit generalizability. The results support the hypothesis that CoAP's structural properties shape adversary behavior, but stronger confirmation requires the broader deployments outlined below.

7.1 Future Work

Several directions emerge from these limitations. Most critically, the honeypot should be extended with high-fidelity Californium emulation - closely replicating QLC Chain device behavior, including correct CBOR responses - so that it passes fingerprinting and progresses to later attack chain phases (Weaponization and Execution in Griffioen's model [3]). The existing A/B test configurations should be retained so that defense effectiveness (RQ2) can be evaluated once attacks are observed.

Geographic distribution is another priority. Deploying honeypots across multiple regions, particularly in Asia, would account for the non-uniform concentration of QLC Chain reflectors in China. Our single Netherlands location may not reflect scanning patterns directed at the primary reflector population. A longer deployment duration and more IPs would further increase statistical power and enable meaningful per-variant comparisons.

Additionally, a controlled reflector disappearance experiment could shed light on how attackers handle churn: deploying a high-fidelity honeypot, allowing it to be catalogued, then taking it offline temporarily to study re-scanning frequency, list eviction timing, and whether previously validated reflectors receive priority re-checks.

Finally, deploying CoAP alongside traditional, older amplification protocols, such as NTP and DNS, on the same infrastructure would enable direct sophistication comparison under identical conditions, eliminating the cross-study methodology caveats present in our current analysis. Monitoring whether CoAP appears in DDoS-for-hire service offerings would provide independent evidence of exploitation complexity - if CoAP is absent from booters despite 1.7 million reflectors, this further supports the high sophistication floor hypothesis.

A Honeypot Response Payload

The following is the CoRE Link Format payload returned by the honeypot's `/.well-known/core` endpoint (698 bytes), emulating a typical IoT device:

```
</sensors/temp>;rt="temperature-c";if="sensor";ct=0,
</sensors/humidity>;rt="humidity-%rh";if="sensor";ct=0,
</sensors/light>;rt="light-lux";if="sensor";ct=0,
</sensors/pressure>;rt="pressure-pa";if="sensor";ct=0,
</sensors/motion>;rt="pir";if="sensor";ct=0,
</sensors/battery>;rt="battery-pct";if="sensor";ct=0,
</sensors/gps>;rt="location";if="sensor";ct=0,
</actuators/led>;rt="led";if="actuator";ct=0,
</actuators/buzzer>;rt="buzzer";if="actuator";ct=0,
</actuators/relay>;rt="power-relay";if="actuator";ct=0,
</actuators/valve>;rt="water-valve";if="actuator";ct=0,
</config/network>;rt="config";ct=0,
</config/firmware>;rt="firmware";ct=0,
</ota/status>;rt="ota-status";ct=0,
</device>;rt="device-info";ct=0
```

References

[1] A10 Networks. DDoS threat intelligence report. https://www.a10networks.com/resources/reports/ddos-threat-report/, 2024. Accessed: 2026-06-10.

[2] Carsten Bormann and Zach Shelby. Block-Wise Transfers in the Constrained Application Protocol (CoAP). RFC 7959, August 2016.

[3] Harm Griffioen, Kris Oosthoek, Paul van der Knaap, and Christian Doerr. Scan, test, execute: Adversarial tactics in amplification DDoS attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 940–954. ACM, 2021.

[4] Lukas Krämer, Johannes Krupp, Daisuke Makita, Tomomi Nishizoe, Takashi Koide, Katsunari Yoshioka, and Christian Rossow. AmpPot: Monitoring and defending against amplification DDoS attacks. In *Proceedings of the 18th International Symposium on Research in Attacks, Intrusions, and Defenses (RAID)*, pages 615–636. Springer, 2015.

[5] John Preuß Mattsson, Göran Selander, and Christian Amsuess. Amplification attacks using the Constrained Application Protocol (CoAP). Internet-Draft draft-irtf-t2trg-amplification-attacks-05, 2024.

[6] NETSCOUT ASERT. CoAP attacks in the wild. https://www.netscout.com/blog/asert/coap-attacks-wild, 2019. Accessed: 2026-06-10.

[7] Phenomite. AMP-Research: Amplification DDoS attack payloads. https://github.com/Phenomite/AMP-Research. Accessed: 2026-06-10.

[8] Rapid7. Metasploit module library. https://www.rapid7.com/db/modules/. Accessed: 2026-06-10.

[9] Christian Rossow. Amplification hell: Revisiting network protocols for DDoS abuse. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2014.

[10] Zach Shelby. Constrained RESTful Environments (CoRE) Link Format. RFC 6690, August 2012.

[11] Zach Shelby, Klaus Hartke, and Carsten Bormann. The Constrained Application Protocol (CoAP). RFC 7252, June 2014.

[12] The Shadowserver Foundation. Accessible CoAP devices. https://www.shadowserver.org/. Accessed: 2026-06-10.