Freie Universität Berlin

Department of Mathematics and Computer Science
Institute of Computer Science

Doctoral Dissertation

DDoS Attacks: Coverage, Mitigation, and Prevention

Dissertation zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.) am
Fachbereich Mathematik und Informatik der Freien Universitaet Berlin

vorgelegt von

M.Sc. Marcin Nawrocki

Berlin 2023

First Examiner
Prof. Dr. Matthias Wählisch
Technische Universität Dresden, Ex Freie Universität Berlin

Second Examiner
Prof. Dr. Thomas C. Schmidt
Hochschule für Angewandte Wissenschaften Hamburg

Third Examiner
Prof. Dr. Kimberly C. Claffy
CAIDA, University of California San Diego

Day of the Defense
12$^{\text{th}}$ September 2024

Abstract

The Internet is a complex system of autonomous but cooperating networks that constitute a critical Infrastructure with a vast socio-economic significance. Any disruption of the Internet and its services has detrimental effects to its users, be it in the private sector or the industry. This is why Internet research aims for observing, mitigating, and ultimately preventing attacks.

In this thesis, we provide methodologies to evaluate and extend the coverage of attack observations, we assess the efficacy of current and emerging attack mitigation solutions, and we identify new opportunities for attack prevention. We do so by utilizing two major vantage point positions, the Internet core and the Internet edge. Our contributions have an operational impact on today's Internet but also its future deployment.

Zusammenfassung

Das Internet ist ein komplexes System autonomer, aber kooperierender Netzwerke, das eine kritische Infrastruktur mit großer sozioökonomischer Bedeutung darstellt. Jede Störung des Internets und seiner Dienste hat schädliche Auswirkungen auf seine Benutzer, sei es im privaten Sektor oder in der Industrie. Daher strebt die Internetforschung an, Angriffe zu beobachten, abzuwehren und letztendlich vollständig zu verhindern.

In dieser Arbeit stellen wir Methoden zur Bewertung und Erweiterung von Angriffsbeobachtungen bereit, wir überprüfen die Wirksamkeit aktueller und angehender Lösungen zur Angriffsabwehr und wir identifizieren neue Möglichkeiten zur Verhinderung von Angriffen. Wir nutzen hierbei zwei wesentliche Beobachtungspunkte: Den Kern und den Rand des Internets. Unsere Beiträge haben operative Auswirkungen auf das heutige Internet, aber auch auf dessen zukünftige Konfigurationen.

Statement of Integrity

I hereby declare to the Freie Universität Berlin that I have prepared this dissertation independently and have done so without using sources or aids other than those indicated. The present work is free of plagiarism. All excerpts taken verbatim or in content from other publications have been clearly identified as such. This dissertation has not been submitted in the same or similar form in any previous doctoral programs. I agree to have my work examined by plagiarism detection software.

Selbstständigkeitserklärung

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Dissertation selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertig habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht. Diese Dissertation wurde in gleicher oder ähnlicher Form noch in keinem früheren Promotionsverfahren eingereicht. Mit einer Prüfung meiner Arbeit durch ein Plagiatsprüfungsprogramm erkläre ich mich einverstanden.

Berlin, den 15. November 2023, Marcin Nawrocki

Acknowledgments

Matthias. Thank you for your dedicated support over the years. Especially, thank you for putting trust into my work during the extremely intricate beginning of my PhD journey. I am really excited and also motivated by your academic growth, which always served as a role model. Unaware of my future path, all this started during my Bachelors with a rather unimpressive homework on Android security mechanisms for which you (back then "just" a research assistant) required a lot of red ink... Suddenly, I found myself being a PhD student writing my first IMC submission with you, the Prof, during the middle of the night! And now, I am exactly where I wanted to be, thanks to your continuous supervision and this PhD program. **Thomas.** Thank you for your thoughtful supervision, which always brings a change of perspectives, be it methodologically or linguistically. You agreed with Matthias in the right moments; but you also disagreed with him in the right moments. I promise to deepen my knowledge in (Polish) art and literature to hopefully provide more recommendations in the future. **KC.** Thank you for agreeing to review this manuscript. Your detailed feedback is highly appreciated! Also, I admire your courage to organize and host the quiet but wild DDoS experts' plenary discussions. It has introduced me to numerous professionals in the field and holds promise for fruitful collaborations in the future.

Raphael. Thank you for being a skillful and very likable companion since the beginning of my PhD program (remember that last-minute demo debugging of *VAST*?). I deeply admire your engineering skills and always enjoyed your amazing cover slides during our meetings! **Andreas and Samir.** Thank you for changing jobs, seriously, as it made me think of what I really want in life. I guess we will never be able to forget these borderline sandwiches at this very renowned measurements conference... **Clemens.** Thank you for bringing a lot of cheeky youngster energy into the office, during a time where this was very much needed. I enjoyed every Club Mate and always smile when I think about how the couch made it into our office. You immediately absorbed feedback and I am proud to see that you are already writing your own PhD story! PS: *hjkl*. **Pouyan.** Thank you for introducing plants in our office. It seems like it escalated a little bit – we now have our private jungle! I always enjoyed your flood of ideas that questions each and every (not only technical) system. Ah, also kudos for serving as the cookie monster. **Maynard.** Thank you for extending my DNS work with so much care. I am certain, that you will be a successful PhD student, balancing out your discipline (clearly from martial

arts!) and the sudden need for four scooters in Hamburg, because of *reasons*. **Jonas.** Thank you for the great collaboration on QUIC, especially during the IETF hackathon. I witnessed how you repeatedly called the support hot-line after our flight got canceled. If you keep this perseverance, the PhD will be definitely yours!

Mattijs. Thank you for being one of my first contacts at a scientific conference (and thanks to **Alberto** for the nice introduction). I am still puzzled by the fact that I had reviewed your unpublished, anonymized work at a shadow PC and then met you a couple of days later in real life. I highly value your feedback and I am impressed by the level of technical detail you are still involved in. **Timm, Gareth, Ignacio.** Thank you for giving me the chance to rehearse my first conference talk in London, it helped so much. Also, thanks for making all the social events even more fun – which definitely highlights what *really* matters. **Philipp and Lars.** Thank you for the last-mile motivation and for continuing to ask the forbidden PhD questions (*when will you finish your ..., when can we read your..., etc.*), ouch! **John.** Thank you for the very humble but amazingly effective *You should talk with...*, I am absolutely looking forward to be part of *ASERT*!

Erik and Sebastian. Thank you for your infinite, unconditional support as friends and best flat mates ever. The discussions of our brotherhood always put things into perspective, as we explore the complete spectrum of almost everything. **Mel and Carlson.** Thank you for all the understanding and compassion. Progressing knowledge is indeed very time consuming. We should now get another ball and enjoy nature, without all the technology.

Real life is messy, inconsistent, and it's seldom when anything ever really gets resolved. It's taken me a long time to realize that. — Alan Moore, Watchmen

Bibliographical Notes

The chapters of this dissertation are based on the following research papers. Overall, the author of this dissertation was the principal investigator and first author of these publications, *i.e.,* identified research questions and goals, designed the methodology, conducted the experiments and analyses, and also significantly contributed to the writing.

Part I, Coverage of Attack Monitoring

Chapter 2 and Chapter 3 are based on:

[113] M. Nawrocki, J. Kristoff, C. Kanich, R. Hiesgen, T. C. Schmidt, and M. Wählisch, "SoK: A Data-driven View on Methods to Detect Reflective Amplification DDoS Attacks Using Honeypots," in *Proc. of IEEE Euro S&P*, Delft, Netherlands: IEEE, 2023, pp. 576–591. URL: https://doi.org/10.1109/EuroSP57164.2023.00041

[111] M. Nawrocki, M. Jonker, T. C. Schmidt, and M. Wählisch, "The Far Side of DNS Amplification: Tracing the DDoS Attack Ecosystem from the Internet Core," in *Proc. of ACM IMC*, Virtual Event: ACM, 2021, pp. 419–434. URL: https://doi.org/10.1145/3487552.3487835

Part II, Efficacy of Attack Mitigation

Chapter 4 and Chapter 5 are based on:

[109] M. Nawrocki, J. Blendin, C. Dietzel, T. C. Schmidt, and M. Wählisch, "Down the Black Hole: Dismantling Operational Practices of BGP Blackholing at IXPs," in *Proc. of ACM IMC*, Amsterdam, Netherlands: ACM, 2019, pp. 435–448. URL: https://doi.org/10.1145/3355369.3355593

[110] M. Nawrocki, R. Hiesgen, T. C. Schmidt, and M. Wählisch, "QUICsand: Quantifying QUIC Reconnaissance Scans and DoS Flooding Events," in *Proc. of ACM IMC*, Virtual Event: ACM, 2021, pp. 283–291. URL: https://doi.org/10.1145/3487552.3487840

Part III, Opportunities for Attack Prevention

Chapter 6, Chapter 7, and Chapter 8 are based on:

[115] M. Nawrocki, T. C. Schmidt, and M. Wählisch, "Industrial Control Protocols in the Internet Core: Dismantling Operational Practices," *Wiley Int. J. Netw. Manag.*, vol. 32, no. 1, 2022. URL: https://doi.org/10.1002/nem.2158

[112] M. Nawrocki, M. Koch, T. C. Schmidt, and M. Wählisch, "Transparent Forwarders: An Unnoticed Component of the Open DNS Infrastructure," in *Proc. of ACM CoNEXT*, Virtual Event: ACM, 2021, pp. 454–462. URL: https://doi.org/10.1145/3485983.3494872

[116] M. Nawrocki, P. F. Tehrani, R. Hiesgen, J. Mücke, T. C. Schmidt, and M. Wählisch, "On the Interplay between TLS Certificates and QUIC Performance," in *Proc. of ACM CoNEXT*, Rome, Italy: ACM, 2022, pp. 204–213. URL: https://doi.org/10.1145/3555050.3569123

Awards

Several publications referenced above have been distinguished with awards. Paper [115] is a fast-tracked journal extension that was previously nominated as 1 of the best 15 papers at IEEE NOMS 2020 conference. Paper [112] received the best presentation award at ACM CoNEXT 2021. Paper [116] received the ACM CoNEXT 2022 Best Paper Award as well as the Community Award. Moreover, the author was awarded a silver medal during the SIGCOMM 2018 student research competition, while presenting collaborative work [54].

Contents

Chapter 1

Introduction

In this chapter, we present the overarching problem context, introduce related research questions and highlight the challenges in answering them. The goal of this chapter is to provide a meta-methodology to enable the reader to understand the structure of this manuscript. By presenting a short background on attacks in the Internet, the stakeholders, and possible vantage points, we enable a first overview. Context-specific background, methodologies, *etc.* are then shown in the respective chapters. Most importantly, we display our key contributions and link them to research questions and chapters, which also functions as an outline of this manuscript.

1.1 Attacks in the Internet

The Internet constitutes a critical infrastructure. It functions as a basis for business activities, private communication, leisure activities. Any disruption of the Internet and its services has detrimental effects to its users. Unfortunately, inadvertent mis-configurations are not the only reason for such disruptions. Attacks in the Internet, more precisely Distributed-Denial-of-Service (DDoS) attacks, are conducted by threat actors. Therefore, we not only have to foster an Internet enabling efficient communication, but also make the Internet secure.

The original design of the Internet unfortunately attributed high levels of trust to its network participants, which led to security being only a (by this time necessary) afterthought. To this day, we deal with conceptual implications that made the Internet a performant but vulnerable communication medium. This is why Internet research aims for observing, mitigating, and ultimately preventing attacks. Only a deep understanding of the DDoS landscape facilitates an effective improvement.

1.2 An Itinerary for DDoS Research

DDoS research is a game of cat-and-mouse with the attackers. We want to (*i*) disarm vulnerabilities before they get misused, (*ii*) develop effective mitigation whenever disarming is not possible, and (*iii*) precisely describe attack activities and potentially trace back the attackers.

However, making any wide-reaching, meaningful observations and changes in the Internet is difficult. This is because the Internet is a network of inter-connected but autonomous networks.

Nobody is able to place measurement probes in all networks and gain complete visibility. This applies also to DDoS research. Selecting appropriate vantage points, usually in edge networks, and carefully extrapolating from local to global observations is the prevailing norm [64], [113], [249]. But also central vantage points, which allegedly provide a unique view of the Internet, stumble upon large data volumes and fall back to traffic sampling and truncation, limiting their view [3], [24], [70], [108], [109]. An effective correlation of orthogonal methodologies is necessary to evaluate vantage point distortion [54], [64], [111].

Almost no ground-truth on DDoS attacks further impedes DDoS research. This is because the Internet follows socio-economical dynamics. Victims of DDoS attacks do not directly gain anything from sharing attack information – sharing data with the community for altruistic reasons is not enough. Especially, if the victim could face a financial loss due to damages in its reputation. Also non-disclosure agreements part of business relations further complicate this.

Despite these high-level challenges, we aim for advancing the reliability and efficacy of measurement-based methods used in DDoS research, namely attack detection, mitigation, and prevention. This includes (*i*) a systematic understanding of the potentials and limits of attack monitoring systems by comparing commonly deployed sensors with new vantage points, (*ii*) assessing the efficacy of attack mitigation by analysing currently deployed as well as novel DDoS mitigation proposals, and (*iii*) identifying opportunities of new measurement methods to guide protocol design and real-world deployment to reduce the attack surface.

Figure 1.1 illustrates the taxonomy of the chapters in this thesis. We differentiate by vantage points, *i.e.,* Internet core and the edge. More precisely, our measurements focus on data and control plane traffic at Internet Exchange Points (IXPs). These enable networks to exchange traffic without crossing the traditional Internet core, a clique of tier-1 transit providers. Moving to the edge, we focus on measurement tools deployed in edge networks, such as honeypots and telescopes, but also review the security implications of endhost deployments, *e.g.,* CDN servers. We also classify the chapters into observing, mitigating, and preventing attacks. Please note that this classification also indicates the criticality of the measurements, with increasing significance from left to right. Although mis-quantifying the total of attacks is poor practice, it does not require *immediate* action. Detecting systems vulnerable to attacks, however, should lead to an instant reaction and efforts to remediate.

1.3 Challenges and Research Questions

In this section, we present the challenges and research questions of this thesis.

1.3.1 Research Endeavor 1: Reviewing the Coverage of Attack Monitoring

A holistic view on the DDoS landscape is essential for understanding the current threat situation. Preferably, measurement methods should observe all attacks to enable an absolute quantification, or at least enable a complete qualitative assessment, *i.e.,* capture all kind of

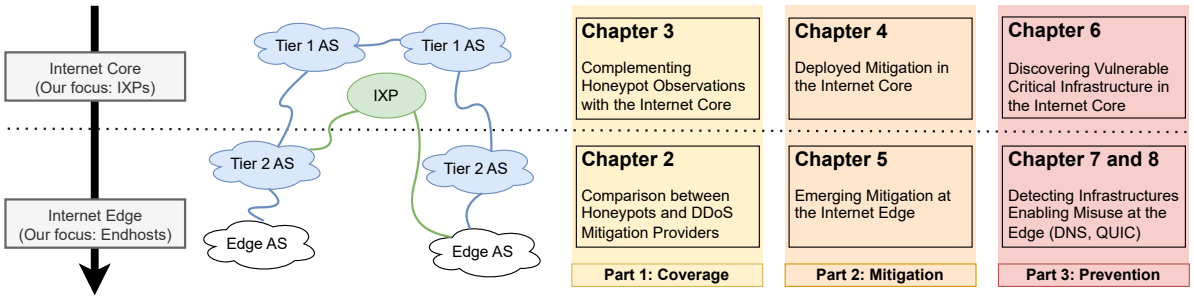


Figure 1.1: Chapter taxonomy. We differentiate vantage points (Internet core and edge), and research goals with their criticality (attack coverage, mitigation and prevention).

attack types and attack sources. Such information can be used to focus remediation efforts on the heavy-hitters. We will review and explore DDoS attack monitoring methodologies for the Internet core and edge, and examine the potential differences.

Comparison between Honeypot Sensors and DDoS Scrubbing Providers

Honeypots are a major measurement tool to observe volumetric DDoS attacks at the edge of the Internet [50], [76], [81], [133], [146]. By emulating attractive amplifiers, *i.e.,* hosts running open services that enable reflective amplification, they infer a set of DDoS victims. However, honeypots require careful design of both data collection and data analysis including cautious threshold inference. As of today, by deploying *enough* sensors and by using *correct* thresholds, we assume a complete picture of the reflective DDoS attack landscape [50]. Due to the lack of ground-truth data and reference points (*i.e.,* no honeypot platform should be selected as the prime platform), this assumption remains largely unchallenged. A driving factor for the completeness assumption is the convergence of observations, *i.e.,* from a certain point on new honeypots do not improve the attack visibility significantly. Publications also tend to focus on attack quantification than threshold justification [76], [81], [113], [146], so effects of specific threshold configurations remain open.

By receiving a baseline data set from a major DDoS mitigation provider, we are able to validate whether the lack of interaction between attackers and honeypots indeed leads to incomplete observations. Moreover, we can systematically explore the complete threshold space and describe effects on attack visibility. This leads to the following research questions:

Research Questions

- ➜ How complete are attack observations based on reflective amplification honeypot measurements?
- ➜ Do current (honeypot) attack detection thresholds indicate DDoS attacks?

Complementing Honeypot Observations with the Internet Core

Internet Exchange Points (IXPs) are hubs in the core of the Internet at which large quantities of inter-domain traffic are exchanged. Observing the complete traffic is not possible due to resource constraints [3], [24], [129]. Therefore packet sampling, flow monitoring, or a combination of both, is deployed [36]. Despite this limited view on traffic, these methods still aim for a representative sample. In contrast to honeypots, IXPs are originally fully passive, neutral vantage points that carry production *and* attack traffic. Hence, detecting reflective amplification attacks at IXPs requires a distinction of between traffic types. It remains open how feasible this is, especially in the face of no external or internal (*e.g.,* blackholing) data sources.

In theory, IXPs are able to observe both, spoofed requests to and amplified responses from amplifiers [45], [85], [108]. This potentially allows for quantifying total attack intensities, a property invisible to honeypots. Furthermore, spoofed traffic transits at the IXPs over physical links, which could enable to trace back attackers. Especially the DNS protocol, as it offers many options, *i.e.,* different queries, to amplify traffic [41], [93], [96], [131], provides opportunities to learn how other amplifiers are utilized. This leads to the following research questions:

Research Questions

- ➜ Can we detect and trace back attacks at the Internet core, i.e., Internet Exchange Points (IXPs)?
- ➜ Does an IXP-centric view contribute additional insights into reflective amplification attacks?

1.3.2 Research Endeavor 2: Assessing the Efficacy of Attack Mitigation

Upon attack detection, mitigation is activated to limit adverse effects of the attack. Mitigation solutions are required to prevent overloading the target but also to protect adjacent network infrastructure. Efficient mitigation should perfectly differentiate between attack and productive traffic, *i.e.,* keep collateral damage low, and take immediate effect as well as introduce a minimal performance impact. Reviewing mitigation solutions is an important part of the battle against DDoS attacks. We will evaluate a common DDoS mitigation solution in the Internet core as well as a mitigation extension of an emerging protocol, deployed in the edge.

Deployed Mitigation at the Internet Core

Some IXPs offer Remote Triggered Black Hole filtering (RTBH) [33], [34], [49], [65] as an additional service to their members. This enables autonomous systems to indicate that they do *not* want to receive traffic for specific prefixes from other members. Filtering in the Internet core is beneficial since attack traffic is dropped early, even before it enters the respective network. Overall, it protects not only end systems but also the Internet infrastructure.

At IXPs, blackholing signals are propagated over BGP via routerservers [129]. Autonomous systems attract traffic with a less-specific prefix announcements, and then temporarily block traffic with a more-specific prefix RTBH announcement for the addresses under attack. First, since RTBH blocks all traffic to the respective prefix, hyper-specific prefixes are used to limit adverse effects to uneffected network segments. Second, as BGP does not originally support RTBH, specific fields are semantically overloaded (`next hop, communities`) with this new meaning [33]. Both issues may lead to misconfigurations and impede filter efficacy. Lastly, RTBH introduces collateral damage by dropping all, *i.e.,* also legitimate, traffic. These research questions arise:

Research Questions

- ➜ How effectively does remotely-triggered blackholing mitigate DDoS attacks at IXPs?
- ➜ Do we need fine-grained filtering solutions to prevent collateral damage?

Emerging Mitigation at the Internet Edge

QUIC as an emerging protocol has seen wide adoption, driven by large content delivery networks [27], [28], [67], [83], [121], [135]. It promises a fast connection setup, *i.e.,* handshakes, between clients and servers, both situated in different edge networks. This new protocol was designed including a DDoS protoction mechanism similar to TCP `SYN` cookies, the `RETRY` token [203]. In theory, `RETRYs` prevent server-side resource exhaustion attacks, more specifically randomly spoofed, state-building DDoS attacks.

Measuring such attacks without being on-path is difficult because they induce state in remote networks. As the connection establishment is similar to TCP, as well as the attack vector, previously used methods such as network telescopes may be a fit to infer such attacks [64], [102]. However, little is known whether QUIC attacks indeed already exist and whether previously applied inference thresholds still hold for QUIC. Also, the practical consequences based on real-world deployments and implementations `RETRY` remain untested. This leads to the following research questions:

Research Questions

- ➜ Are emerging protocols such as QUIC already affected by DDoS attacks?
- ➜ Can QUIC deployments be made more resilient against attacks with built-in protection mechanisms?

1.3.3 Research Endeavor 3: Identifying Opportunities for Attack Prevention

Preventing attacks altogether is the ultimate goal of DDoS research, albeit the most difficult to achieve. Internet measurements are performed to discover unprotected, insecure systems that can be either attacked directly or misused to foster attacks on third-parties. Wrong deployment and implementations bugs, often hand in hand, facilitate the exploitation. Identifying vulnerabilities before the attackers prevents harmful attacks. Based on Internet-wide measurements, we now identify new opportunities for attack prevention utilizing the Internet core and edge.

Discovering Vulnerable Critical Infrastructure

Industrial control systems (ICS) are managed by dedicated protocols. These protocols were originally designed without security considerations because of isolated deployment scenarios [12], [97], [104], [142], [161]. But ICS protocols have nowadays been adapted to Internet transport which potentially leads to unprotected, inter-domain traffic, if no further security mechanism are deployed [132]. Hence, attackers on-path, *e.g.,* in the Internet core, could eavesdrop or manipulate crucial steering messages. We explore the feasability of such Man-In-The-Middle (MITM) attacks.

ICS protocols are utilized in specialized niche deployments, such as power plants [151]. This means that common tools for traffic classification do not exist or require very careful usage, especially in the Internet core at IXPs, where traffic underlies sampling and truncation. Moreover, since Internet-wide scans seek for ICS deployments, we have to exclude such traces to only detect operational, unprotected ICS traffic. This leads to the following research question:

Research Questions

- ➜ Are current deployments of Industrial Control Systems (ICS) vulnerable to Man-in-the-Middle attacks?

Detecting Infrastructure That Enables Misuse

Reflective amplification attacks are one of the most common attack vectors [64], [218], [250], [251]. Removing infrastructure that enables misuse towards third parties, *e.g.,* amplifiers, is very beneficial to the Internet ecosystem as a whole. This is why Internet-wide scans are periodically searching for new amplifiers and automatically contact the respective network operators [37],

[38], [101]. We now look for systems of traditional (DNS) but also emerging protocols (QUIC) that enable misuse.

For DNS, we focus on transparent forwarders, which were anecdotally documented but disregarded by our research community [81], [238]. Their ability to forward but spoof DNS requests makes them difficult to detect by common Internet-wide scanning techniques. Overall, we aim for a comprehensive view of the open DNS threat landscape.

QUIC is based on UDP, which makes it also susceptible to amplification attacks [203], [91]. By design, the standard dictates an anti-amplification limit, which limits first server responses to `3x` of received client data. However, implementations may inadvertently deviate from this limit. Due to QUIC handshake optimizations, there is now a dependence between this limit, Web PKI certificate chains, TLS certificate compression, padding, and even retransmissions for loss correction [157]. As QUIC is the transport protocol for HTTP/3, understanding these dependencies and real-world deployments is crucial to prevent future misuse. This leads us to the following research questions:

Research Questions

- ➜ Do current scanning campaigns detect all open DNS components?
- ➜ To which extent does the PKI ecosystem influence transport layer (e.g., QUIC+TLS) performance and security?

1.4 Key Contributions

We summarize our research questions and key contributions in Table 1.1 and Table 1.2.

Comparison between Honeypot Sensors and DDoS Scrubbing Providers. In Chapter 2, we show that a large honeypot platform only observes ≤5% of baseline reflective amplification attacks, as received from the customers of a major DDoS mitigation provider. We make this observations although the honeypot platform is subject to attack convergence, *i.e.,* the observations appear saturated and additional honeypots to not significantly improve attack visibility. This challenges fundamental completeness assumptions of reflective DDoS measurements. Moreover, although related work adapts various attack thresholds, we find no significant differences between these configurations. The thresholds correctly classify the visible events into attacks and scans, the latter event type showing high congruence with scans from telescopes. However, we need to improve the overall attack visibility of honeypots, most likely by increasing the attacker interaction. This is why we recommend to focus efforts on understanding the attacker behavior.

Complementing Honeypot Observations with the Internet Core. In Chapter 3, we introduce a passive attack detection method for the Internet core, *i.e.,* IXPs. Utilizing this method, we detect DNS amplification attacks, of which 96% were not visible to a sizable

honeypot platform. By following the physical ports of spoofed traffic, we find that such traffic originates from autonomous systems with large customer cones. This inhibits the trace back of the attack origin. We fingerprint a major attack entity using DNS query properties. This attack entity dominates the DNS attack ecosystem and makes use of inefficient DNSSEC key rollovers present in the `.gov` zone. We recommend the pre-publish key rollovers.

Deployed Mitigation at the Internet Core. In Chapter 4, we present the first in-depth analysis of all RTBH events at a large European IXP. By combining data and control plane measurements, we find that only one third of RTBH events correlate with indicators of DDoS attacks. Blackholing causes on average dropping of only 50% of unwanted traffic because announcements with hyper-specific prefixes (more specific than `/24` for IPv4) are ignored by IXP members. We quantify the collateral damage introduced by blackholing. However, we also find that ISP clients are commonly under attack which makes fine-grained filtering more difficult, because allow-listing of stable, legitimate traffic patterns is not possible. For IXP environments, we recommend exemptive BGP policies that ultimately incorporate hyper-specific RTBH announcements.

Emerging Mitigation at the Internet Edge. In Chapter 5, we show that Internet telescopes can be utilized to detect state-building QUIC attacks, *i.e.,* `INITIAL` floods. Our telescope captures malicious scans as well as backscatter from randomly spoofed attacks with QUIC, often part of multi-vector attacks. We find that CDN servers at the Internet edge are a common target of such attacks. We perform a synthetic test of the NGINX QUIC server and demonstrate that implementations are indeed vulnerable to QUIC floods, however only if the `RETRY` handshake option is not used. Since `RETRYs` challenge the design goal of 1-RTT handshakes, they are rarely deployed. We recommend a load-based `RETRY` which is dynamically turned on in case of critically high server utilization.

Discovering Vulnerable Critical Infrastructure in the Internet Core. In Chapter 6, we uncover unprotected, operational inter-domain ICS traffic at two central Internet vantage points, an IXP and an ISP. By correlating with data from honeypots and scan campaigns, we are able to filter out non-operational ICS traffic, usually belonging to Internet-wide scanners. We are indeed able to show that ICS traffic transits the Internet core, vulnerable to Man-In-The-Middle attacks. Such attacks include eavesdropping and traffic manipulation. The affected systems were invisible to scan projects which indicates firewalls as a means of protection, however, these systems blindly trust the communication channel over the Internet. We recommend common VPN technologies or recent ICS security extensions for ICS deployments.

Detecting Infrastructure that Enables Misuse at the Edge. In Chapter 7, we systematically measure and analyze open DNS components deployed in the Internet Edge. We find that the open DNS ecosystem consists of 26% of transparent forwarders, which can be misused to launch reflective amplification attacks. Worryingly, the numbers are still increasing because of new deployments, especially in countries with emerging markets. Transparent forwarders are

missed by Internet-wide scanning campaigns due to common optimizations. We validate this by deploying honeypots acting as transparent forwarders. We recommend to examine all new DNS deployments for transparent forwarding behavior. A large ISP followed our recommendation and disarmed ∼200k transparent forwarders.

In Chapter 8, we analyze over 1M Web domains with 272k QUIC-enabled services. We find that 35% of server Web certificates exceed the QUIC anti-amplification limit, which was designed to restrict response data to unverified clients, effectively preventing reflective amplification attacks. For server implementations respecting the limit, this leads to multi-RTT handshakes and thus undermines the performance of the connection setup. Certificate compression serves as a short-term remedy, however, long-term solutions should reduce the Web certificate chain sizes. In the case of faulty server implementations which do not respect the limit, large certificates lead to a higher amplification factor. We detect factors $\geq 30\times$ in IP spoofing scenarios for a major CDN, ready to be weaponized. We recommend careful analysis of the anti-amplification behavior in the case of large certificates, padding, and resends. The major CDN disarmed the faulty servers.

1.5 On How to Read This Thesis

This thesis scrutinizes the DDoS phenomenon in the global Internet across multiple dimensions and vantage points. Figure 1.1 shows the most prevalent split in this thesis: (*i*) By Internet topology, *i.e.,* core or edge, and (*ii*) by research target, *i.e.,* observing, mitigating, or preventing attacks. Rather than having separate, overarching chapters dedicated solely to background, methodology, *etc.*, this thesis follows an approach of local coherence. We present the respective information close to the related sections in self-contained chapters. For each chapter, main research questions, main vantage points, and key results are shown in Table 1.1 and Table 1.2.

Table 1.1: Thesis overview for *Part I* and *Part II*: Research questions, vantage points, key results, and additional operational outreach.

Chapter	Research Questions	Measurement Vantage Points			Key Results	Operational Outreach
		Topo.	Type	Data		
Part I: Reviewing the Coverage of Attack Monitoring						
Chapter 2	How complete are attack observations based on reflective amplification honeypots?	Edge	Active	Honeypot traffic	Honeypots observe ≤5% of baseline attacks, which challenges fundamental completeness assumptions.	CCC honeypot operators [234]
	Do current attack detection thresholds indicate DDoS attacks?	Edge	Passive	DDoS mitigation alerts, Telescope traffic	Correct classification of attacks and scans. But we need better attacker models to improve visibility.	
Chapter 3	Can we detect and trace back attacks at the Internet core, *i.e.,* Internet Exchange Points (IXPs)?	Core	Passive	Sampled IXP traffic	Amplification attacks (DNS) are detectable at IXPs. But large customer cones inhibit the trace back.	APNIC blog [229]
	Does an IXP-centric view contribute additional insights into reflective amplification attacks?	Edge	Active	Honeypot traffic, DNS scans	We fingerprint a major attack entity not visible to honeypots misusing prior unseen DNS inefficiencies.	
Part II: Assessing the Efficacy of Attack Mitigation						
Chapter 4	How effectively does remotely-triggered blackholing mitigate DDoS attacks at IXPs?	Core	Passive	BGP routeserver data, IXP flow data	Blackholing causes on average dropping of only 50% of unwanted traffic due to inapt BGP configurations.	RIPE meeting [224] and blog [225], APNIC blog [226], MIX Salottino [227]
	Do we need fine-grained filtering to prevent collateral damage?				Blackholing leads to collateral damage but fine-grained filtering is intricate due ISP clients under attack.	
Chapter 5	Are emerging protocols such as QUIC already effected by DDoS attacks?	Edge	Passive	Telescope traffic	QUIC Internet background radiation indicates scans and randomly spoofed multi-vector attacks.	Caida DUST [228]
	Can QUIC deployments be made more resilient with built-in protection mechanisms?	—	Simulation	QUIC DDoS floods	QUICs `RETRY` handshake option prevents DoS but challenges the design goal of fast connection setup.	IETF MAPRG [236]

Table 1.2: Thesis overview *Part III*: Research questions, vantage points, key results, and additional operational outreach.

Chapter	Research Questions	Measurement Vantage Points			Key Results	Operational Outreach
		Topo.	Type	Data		
Part III: Identifying new Opportunities for Attack Prevention						
Chapter 6	Are current deployments of Industrial Control Systems (ICS) vulnerable to Man-in-the-Middle attacks?	Core	Passive	Sampled IXP traffic	We detect unprotected, operational ICS traffic in the Internet core, prone to traffic manipulation.	MIT Technology Review [244]
		Edge	Active	Honeypot traffic, Scan data		
		Edge	Passive	Telescope traffic		
Chapter 7	Do current scanning campaigns detect all open DNS components?	Edge	Active	Honeypot traffic, DNS scans	Due to common optimizations, campaigns miss 26% of all DNS amplifiers (transparent forwarders).	APRICOT meeting [230], large ISP
Chapter 8	To which extent does the PKI ecosystem influence transport layer (QUIC) performance and security?	Edge	Active	QUIC scans, HTTPS/TCP scans	Large TLS data inflicts prolonged, multi-RTT QUIC handshakes and facilitates volumetric attacks with faulty server implementations.	IETF MAPRG [231] and QUICWG [233], APNIC blog [232], large CDN
		Edge	Passive	Telescope traffic		

Part I

Coverage of Attack Monitoring

Chapter 2

SoK: A Data-driven View on Methods to Detect Reflective Amplification DDoS Attacks Using Honeypots

Abstract

In this chapter, we revisit the use of honeypots for detecting reflective amplification attacks. These measurement tools require careful design of both data collection and data analysis including cautious threshold inference. We survey common amplification honeypot platforms as well as the underlying methods to infer attack detection thresholds and to extract knowledge from the data. By systematically exploring the threshold space, we find most honeypot platforms produce comparable results despite their different configurations. Moreover, by applying data from a large-scale honeypot deployment, network telescopes, and a real-world baseline obtained from a leading DDoS mitigation provider, we question the fundamental assumption of honeypot research that convergence of observations can imply their completeness. Conclusively we derive guidance on precise, reproducible honeypot research, and present open challenges.

2.1 Introduction

Distributed Denial of Service (DDoS) attacks are a serious threat to the Internet infrastructure. Reflective amplification attacks [133], [266], a specific DDoS type, are a unique burden since they allow an attacker to trigger large traffic volumes from third parties by exploiting protocol mechanics rather than hijacking hosts. Over the last many years, amplification attacks have been responsible for a significant number of attacks [251].

A common approach to detect amplification attacks in the wild is the deployment of honeypots [249]. They mimic application protocols such as DNS and NTP that are susceptible to amplification attacks, wait for attackers to interact, and then log attack traffic attempting to abuse them as amplifiers. Amplification honeypots may be able to infer the size of attacks based on traffic patterns as well as identify the victims they are instructed to reflect toward.

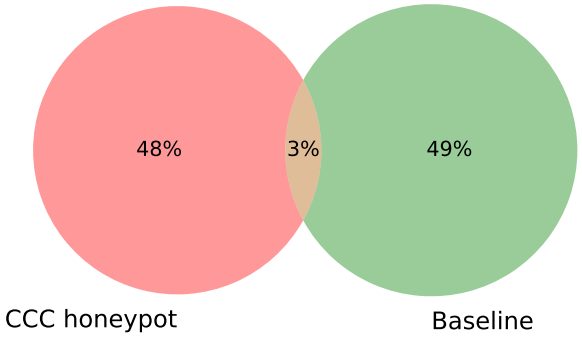


Figure 2.1: Relative shares of victims observed at a large-scale amplification honeypot and confirmed at a large DDoS mitigation provider (Baseline).

Research on amplification honeypots is usually guided by three questions to evaluate whether honeypots are a viable tool. First, which heuristics identify packets that correspond to an attack in a train of packets captured by honeypots (*attack detection*)? Second, how many honeypot sensors are necessary to capture a stable amount of events (*honeypot convergence*[1])? Third, do sensors capture a representative view of Internet-wide attacks (*completeness*)? These aspects should be considered separately. Attack detection, for example, might be accurate on a given data set, while the data set does not include all attacks.

Surprisingly, our community mixes detection, convergence, and completeness. For more than ten years, we have been holding the common belief "[t]he more honeypots we deploy, the more likely one of them is contacted" [260]. Even with the advent of amplification honeypots we still believe that we can nearly achieve completeness: "This shows that—per mode—we had enough honeypots to cover most attacks out there."[76], "[...] as many as 150 honeypots are needed to capture 99% of actor behavior" [50], "[...] our reflectors can see between 85.1% and 96.6% of UDP reflection attacks" [146]. A key insight of this chapter is that reality is different.

In this chapter, we revisit the long-held beliefs about the visibility and attack detection precision of honeypots. We combine two different methods by (*i*) systematizing and contextualizing existing knowledge and (*ii*) implementing a data-driven approach, which clearly shows that common beliefs do not hold.

Based on an extensive literature study, we select six amplification honeypots and compare them. The six honeypot platforms were used in security studies when analyzing reflective-amplification attacks based on honeypot data. They have been published, cited recently, and had a notable impact on security research. We implement three steps. (*i*) We survey the honeypot deployment configurations that enable observations, *e.g.,* the number of honeypot sensors deployed and the geographical and topological distribution of the platform, (*ii*) we

[1]Throughout this chapter, we use the terms honeypot convergence and attack convergence interchangeably for the phenomenon of allegedly saturated attack inferences made with reflective amplification honeypots.

describe the attack definitions that are used to understand the observations, and (*iii*) we assess the rationale behind the argument that the deployed honeypot achieves completeness.

To bolster our arguments, we conduct a data-driven approach. Our data corpus covers three months and includes measurements from a large-scale honeypot, four network telescopes distributed in the US and Europe, and baseline real-world alerts from a leading DDoS mitigation provider. Figure 2.1 motivates this approach. It shows the overlap of victims under attack monitored by a well-known research honeypot project and a baseline of attacks against customers of a leading DDoS mitigation provider. The overlap is small, and most importantly the honeypots do not capture a significant portion of attacks targeting real-world networks, even though a honeypot could capture those incidents in principle.

Contributions. In a nutshell, our systematization of knowledge stresses that the research community could benefit from a framework that allows for algorithmic assessment of honeypot deployments and, to assemble packets captured by honeypots to malicious flows, from attack detection heuristics that adaptively incorporate deployment properties. Our key contributions are:

1. We explore the comparability of the attack detection thresholds used by six honeypot platforms, and place them in the complete threshold space. All thresholds but one produce similar results.

2. We present a systematic approach to analyze data collected by honeypots. We identify the key properties that should be considered and documented to improve reproducibility of future honeypot research.

3. We show that honeypot convergence, a frequently used measure, is a poor indicator for the completeness of observations. This metric is statistically unstable. Sizable honeypot platforms only observe up to 11% of baseline attacks.

4. We find that current honeypot deployments do not significantly benefit from better attack detection thresholds because attackers simply do not interact with honeypots. This may help to improve the placement of honeypot sensors in the future.

5. We discuss how amplification features of protocols can influence honeypot observations and analysis.

Outline. The remainder of this chapter is guided by our research questions, see Table 2.1. We present basic background in Section 2.2, introduce our method in Section 2.3, and survey common honeypot platforms in Section 2.4. In Section 2.5, we present the data sets that we use for our data-driven analysis, We revisit attack detection, convergence, and completeness in Section 2.6 to Section 2.8. In Section 2.9, we present further deployment dimensions of honeypots. We discuss our findings comprehensively and provide further guidance in Section 2.10, and conclude in Section 2.12.

Table 2.1: Our SoK addresses the following research questions, guiding (*i*) knowledge contextualization, (*ii*) data-driven evaluation, and (*iii*) further discussions.

SoK	Research Question	Section
Introduce	Which kinds of attacks and monitoring exist?	2.2
Compare	How are amplification honeypots deployed?	2.4
Compare	How are attacks inferred?	2.6,2.6.1
Compare	How are comprehensive measurements justified?	2.6.3,2.7 ,2.7.1
Evaluate	Do different attack thresholds skew the results?	2.6.2
Evaluate	Do honeypots observe all attacks?	2.7.2,2.7.3,2.7.4,2.8.1
Evaluate	Do we need more precise attack thresholds?	2.8.2,2.8.3
Discuss	What makes measurements prone to errors?	2.9
Discuss	What do we recommend for future work?	2.10

2.2 Problem Statement and Background

2.2.1 Distributed Denial of Service (DDoS) Attacks

Denial of Service (DoS) attacks impair the network availability of their victims. This is achieved by resource exhaustion caused by overloading the infrastructure with excessive traffic volume or connection state at the victim. Attackers either set up genuine communication channels with the victims or spoof IP source addresses to obfuscate their attacks. Both methods are typically conducted using a distributed botnet. Two attack types exist, each of which take advantage of the first round-trip time when a server responds to client requests.

(*i*) State-building, randomly-spoofed attacks such as TCP SYN or QUIC Initial floods. Each spoofed request can trick the server into setting up a new connection context for non-existent clients. The network stack will maintain all currently active connections, including those from spoofed sources, which fill up the connection queues and cause legitimate requests to fail. Since the server tries to respond to each connection request, it will send *backscatter*, *e.g.,* TCP SYN/ACK or QUIC (server-) Initial packets, to the spoofed addresses. TCP SYN cookies and QUIC RETRYs may mitigate those attacks [169], [110].

(*ii*) Distributed Reflective amplification attacks (DRDoS) combine targeted address spoofing and protocol mechanics of public services such as DNS and NTP to amplify response traffic to the victim. In a DRDoS attack, request packets with the spoofed source address of the victim are sent to public third-party servers. These servers act as amplifiers since responses to the victim can be be many times larger than the original request [133]. For example, a typical DNS query packet is about 100 bytes, but a response to an `IN ANY` query can often exceed 2000 bytes in practice. Attackers seek to minimize the request volume towards amplifiers whilst

maximizing the response volume reflected to the victim. This may congest network links along the path to the victim.

Attack Popularity over Time. Conceptually, DRDoS was already utilized in 1997 with ICMP smurf attacks. However, direct-path SYN-floods remained the most popular DDoS attack vector from 1996 to 2018 and were then overtaken by DNS reflection-amplification in 2018. This popularity was due to (*i*) the commercialization of this attack type by booter services, making it available to the non-tech-savvy public, and (*ii*) easier and faster detection of amplifiers based on ready-to-use tools implementing state-less, Internet-wide scans.

2.2.2 Honeypots and Network Telescopes

Honeypots. Honeypots are decoy computer resources whose value lies in being probed, inciting interaction with attackers, and possibly getting compromised [260]. They are not a preventive countermeasure such as firewalls but a way to detect the presence of actions that harm a system. Since honeypots do not offer production-critical services, all connections to the honeypot are inherently suspicious. This enables easy detection of an unauthorized probe, scan, or attack, because malicious actions are not buried in the vast amount of legitimate production activities.

Honeypots can be classified along two dimensions, based on the level and type of interaction they offer. First, based on the level of interaction the delineation is (*i*) low-interaction honeypots, (*ii*) medium-interaction honeypots and (*iii*) high-interaction honeypots. Low-interaction honeypots offer only a minimal response-behavior, *e.g.,* they only perform transport-layer handshakes. Medium-interaction honeypots extend this behavior by emulating vulnerable services or partially exposing vulnerable components, *i.e.,* they produce valid replies for specific applications. Given the reduced interaction capabilities in low- and medium-interaction honeypots, the chances of compromise are minimal, which eases deployment. High-interaction honeypots offer unrestricted, real operating system environments. They are more complex to implement, deploy, and maintain. They enable, however, forensics to fully observe the behavior of malware, *e.g.,* bots, or ransomware.

Second, based on the type of interaction they offer, honeypots are classified into (*i*) server and (*ii*) client honeypots. Server honeypots wait for an incoming connection. They may not advertise services explicitly, more likely they are discovered before the attack, usually using lightweight scanning or probing that involves higher layer protocols. In contrast, client honeypots actively search for suspicious entities and solicit interaction with them, such as web crawlers visiting malicious websites.

Honeypot classification is largely academic. Since many honeypot variants exist, a distinction is not always possible, nor practical. In practice, the terms for low- and medium-interaction honeypots are often used interchangeably.

Methods to distinguish attacks from other types of traffic collected at honeypots have been proposed. With the advent of reflective amplification attacks, server honeypots for the the sole purpose of capturing DRDoS attacks have been designed, implemented, and deployed. We discuss amplification honeypot platforms in detail in Section 2.4.

Network telescopes. Network telescopes [1], [22], [31], [48], [128], [158] are an unsolicited traffic measurement approach that captures incoming traffic to otherwise unused address space within a larger network segment. These typically cover between a `/8` and `/24` of IPv4 address space. Originally, network telescopes were fully passive and the network segments were never used to originate any traffic. They capture both backscatter traffic (*i.e.,* replies to spoofed addresses of the telescope) and scan traffic. With the increased deployment of malicious two-phase scanners [62], *i.e.,* attackers that first check whether a TCP service is available before they initiate application requests, reactive telescopes have been proposed [53]. Reactive network telescopes implement the TCP connection handshake to gain additional knowledge about the attacker, since the attacker will proceed with an application layer request.

2.2.3 Monitoring Spoofed DDoS Attacks

When monitoring traffic two crucial questions arise. (*i*) Where should network probes be deployed? (*ii*) Which packets belong to which type of event (*e.g.,* scan, attack)?

Non-spoofed traffic, or direct-path attacks, can only be observed by systems that are deployed between the attack source, the destination target, or at the endpoints. For example, an appliance to mirror traffic might be located alongside a victim service, at a network ingress port, or within an Internet exchange point. Collecting on-path observations is a challenge for most researchers and the ability to capture related but distinct direct-path attacks can be difficult. In contrast, reflective attacks allow for broader observations because they involve triangular packet flows with the host sending a spoofed packet, a reflector (*e.g.,* honeypot) of the spoofed packet, and the victim host receiving the response to a spoofed request.

Many reflective amplification attacks rely on amplifier lists to quickly and successfully conduct attacks. The lists are commonly curated by third parties and sold to attackers. These lists may contain a subset of all known and currently active amplifiers. When monitoring amplification attacks, an amplification honeypot should emulate amplifier behavior to be appealing to attackers. To minimize harm, amplification honeypots typically apply a rate limit to satisfy amplifier discovery, while avoiding the reflection of meaningful attack traffic to a victim.

2.3 Methodology

We now describe our methodology to systematize, contextualize, and evaluate research about amplification honeypots.

2.3.1 Systematization and Contextualization

Our systematization of knowledge aims for an overview and systematic comparison of amplification honeypot research. This systematization is based solely on previously published work, describing presented methods, data sources, and deployments. Our framework includes the following parts.

Selecting honeypot research. We select six honeypot platforms by conducting a systematic literature review searching venues dedicated to security (*i.e.,* Oakland, EuroS&P, Usenix Sec, CCS, NDSS) and measurement (*i.e.,* IMC, PAM, TMA) research, as well as broader networking venues (*e.g.,* SIGCOMM), covering the last ten years. The six honeypot platforms and configurations discussed in this chapter are seminal for research on amplification attacks.

Comparing honeypot deployments. We compare honeypot deployments by their setup configuration, *i.e.,* number of sensors, duration of deployment, and the geographical as well as topological distribution. Moreover, we describe which protocols are supported by the honeypots.

Comparing attack inference. We introduce precise language for describing heuristics that infer attacks from a sequence of packets captured by honeypots. Then, we show the attack definitions applied by the various honeypot deployments, *i.e.,* what are the exact attack thresholds and how are these conveyed in each publication.

Comparing completeness claims. By considering a realistic attack volume and protocol properties as well as public knowledge about the number of deployed amplifiers, we deduce that attackers can easily impede detection by honeypots. We show how honeypot research still collectively claims nearly complete attack visibility, despite the lack of ground-truth attack data and the possibility that attackers may hide.

2.3.2 Data-driven Evaluation

We extend our SoK by conducting a data-driven evaluation. This is necessary because key methods and assumptions in honeypot research cannot be validated without external observations. Based on results derived by our contextualization (see Subsection 2.3.1), we identify further research questions and explore them. In detail, (*i*) we analyze whether different attack thresholds used in prior work have a significant effect, (*ii*) we verify whether honeypots observe all Internet-wide attacks, and (*iii*) we explore the possibilities to improve thresholds.

Evaluating attack thresholds. We assess the comparability across honeypot projects by describing and analyzing the effects of various flow identifiers and attack thresholds. To this end, we apply both flow identifier types used in honeypot research. We then explore the effects of the complete threshold configuration spectrum w.r.t. temporal (*i.e.,* timeouts) and volumetric

(*i.e.,* packet number) properties. We do this on the dataset obtained by the CCC honeypot platform.

Evaluating attack completeness. The stability of observations (*honeypot convergence*) is used to justify that honeypot sensors capture a representative view of all Internet-wide attacks (*completeness*). To validate this, we first review the convergence metric by an optimal, best-case analysis and then proceed with a randomized approach. Following this, we check whether the (converging) CCC honeypot platform observes a set of baseline attacks against customers of a leading DDoS mitigation provider[2].

Evaluating detection potentials. We evaluate whether attack detection thresholds can be improved. We do so by correlating honeypot, telescope, and our baseline data sets. First, we use the DDoS baseline and try to optimize towards this data set, *i.e.,* we improve the honeypot attack detection (but risk over-training towards this specific baseline). By adopting very permissive thresholds, we infer the upper bound of attack detection. Second, we use telescope baseline data to infer whether attack detection thresholds for honeypots already effectively remove baseline scan events.

2.4 Amplification Honeypot Platforms

We now describe some of the best known honeypot deployments as originally presented in their publications. They implement attack detection mechanisms to identify reflective amplification attacks based on the packets they receive. These detection mechanisms, see Section 2.6, can be applied on any data but were presented alongside the data collection platforms described here.

AmpPot. AmpPot [76] deploys 21 sensors supporting nine protocols. The sensors are primarily deployed in ISP environments with half located in Japan and the others spread globally. These sensors are usually configured with static IP addresses, but a quarter receive dynamic addresses with lease times of up to 51 days. An AmpPot sensor can operate in three modes: (*i*) *emulated* runs a partial, internal implementation of the protocol, (*ii*) *proxy* forwards to a separately deployed service, or (*iii*) *agnostic* amplifies with random data independent of the protocol.

AmpPotMod. AmpPotMod [118] uses a subset of the original AmpPot deployment: eight sensors running in proxy mode (except for SSDP) deployed at ISPs in Japan. The sensors support up to six amplification protocols and use dynamically assigned IP addresses.

CCC. The Cambridge Cybercrime Center (*CCC*) [146] platform is a distributed honeypot platform that supports eight protocols. For NTP and DNS, the sensors proxy to real services. In other cases they respond with a limited, emulated answer. The number of sensors fluctuates over time with a median of 65 active sensors (currently 50). Sensors are spread across 10

[2]It is very likely that even *the* leading DDoS mitigation provider is not able to observe all attacks due to the (*i*) distributed nature of the Internet and (*ii*) local attacks. Still, any honeypot platform claiming completeness (*e.g.,* based on convergence) should *at least* observe the baseline attacks. We further discuss this in Section 2.5.

Table 2.2: Data sources utilized in this chapter to revisit common methods to assess honeypots. All data sources span November 01, 2021–January 31, 2022.

Data Source	Attack Thresholds (Subsection 2.6.2)	Convergence (Section 2.7)	Completeness (Section 2.8)
CCC Honeypot Events	✓	✓	✓
DoS Mitigation Provider			✓
US & EU Telescopes			✓

countries in academic and cloud networks, located in 31 IP prefixes in 8 ASes. 16 sensors are deployed in their own /28 subnet. The remaining sensors are deployed at low-cost cloud providers and in a handful of consumer ISPs.

NewKid. The NewKid platform [52] deploys a single sensor supporting 9 protocols in a university network. The sensor operates in proxy mode for Memcached and DNS, and emulates responses for other services.

HPI. The HPI platform [50] deploys a total of 549 honeypots distributed over five cloud providers and across four continents. The sensors support six protocols (emulated and proxied) in four different modes that signify the protocol correctness and the amplification factor: (*i*) real-small (*ii*) real-large (*iii*) fake-small and (*iv*) fake-large.

It is worth noting that all platforms deploy a form of rate limiting to minimize adverse effects. Table 2.3 summarizes the *setup* properties of the different honeypot platforms.

Impact on other research areas. The groundwork on amplification honeypots was published in three consecutive years, AmpPot [76] in 2015, AmpPot Mod [118] in 2016, and CCC [146] in 2017, followed by HPI [50] in 2021. According to Google Scholar, the oldest honeypot, AmpPot, has been cited the most, reaching nearly three times the citation count of the others. With a few exceptions, all papers are cited in security-related research but have had influence in multiple, related areas. The most impactful citations of AmpPot relate to research on technical aspects of DoS, while AmpPotMod and CCC receive more attention from adjacent areas such as CRIME-related research. Measurement research has more commonly cited AmpPot and CCC compared to AmpPotMod.

2.5 Data Sets for Data-driven Evaluation

We now introduce our data sets, which are summarized in Table 2.2.

2.5.1 Honeypot Data

We use data from the CCC honeypot platform. CCC supplies two types of log formats. First, a list of victims inferred by applying the default CCC thresholds. Second, a list of all event

summaries per sensor. We analyze the second list for testing various thresholds and validate our scripts with the first list by applying the default CCC thresholds and inferring the same victims as CCC did.

We also check whether the CCC platform operated without interruptions. This eliminates a possibly skewed convergence behavior due to external reasons, *i.e.,* a honeypot sensor running only during a fraction of the measurement period would always observe different attacks than a second sensor running at different times.

2.5.2 Telescope Data

Scanning observations vary between telescopes that differ by topological and geographical properties [53]. This is why we use a `/24` telescope from the US and three `/24` telescopes from the EU. In total, 85% of the CCC honeypot sensors are deployed in these regions, which enables a fair comparison.

Our analysis is based on the assumption that telescopes primarily observe scan traffic for UDP. Because network telescopes are fully passive, scanners do not detect open amplifiers in these networks, which could be misused in a subsequent attack event. This means we do not expect spoofed traffic arriving at the telescope. Moreover, attackers sending spoofed queries to a telescope would effectively waste their resources because there is neither reflection nor amplification possible. This makes telescopes a suitable vantage point to identify UDP scanners.

In addition to amplification attacks, other UDP (non-scanning) traffic can be monitored at network telescopes. In 2015, a total of 134 DNS-based amplification attacks have been inferred during a period of 6 months [42]. However, only a handful of these attacks have been verified and most attacks exhibit properties of aggressive scanning rather than attacks, *i.e.,* the number of targeted unique dark addresses equals the number of total packets sent. These observations might be due to the early stage of detection methods of amplification attacks, which, at that time, did not account for fast scanning methods [38].

The deployment of the protocol QUIC [203] recently changed UDP traffic properties at telescopes. Although QUIC runs on top of UDP, it requires a handshake to initiate connections, making it susceptible to state-overflow attacks [110]. This means that we observe DoS backscatter targeting UDP in addition to TCP services. Identifying QUIC backscatter is easy, however, because attacks originate from the default QUIC port and a specific group of content servers. Furthermore, they contain fingerprintable data [110]. Overall, QUIC backscatter does not interfere with our measurements.

Lastly, accidental misconfigurations might lead to UDP traffic at the telescope. We argue that such events are rare and unlikely to reach the ports associated with amplification attacks. However, we cannot completely exclude them.

2.5.3 DDoS Baseline Data Set

We collaborate with the world's largest DDoS mitigation equipment provider with a reported global market share of over 20% in 2020. We receive partially anonymized attack information under a non-disclosure agreement for popular amplification protocols during our main measurement period. In total, we are able to observe all reported attacks for the protocols supported by the CCC honeypots.

The data provided by the mitigation company is based on a DDoS appliance deployed on the direct links between customers and their upstream providers, *i.e.,* they are able to observe all external attacks targeting end hosts in the customer networks. Attack detection is based on observing volumetric peaks and well-known attack vectors to identify anomalous traffic changes. It involves customer feedback, which is important for mitigation (traffic scrubbing), since scrubbing could lead to unwanted packet loss in case of false positives.

Our data set includes start and stop time of an alert, attack type, and flow selection criteria. For each attack event, we obtain the list of protocols misused, destination prefixes receiving traffic as observed by the sensor, but without a detailed breakdown of traffic volumes by target. Although inferring the specific targets and the impact from attack from this list is usually not possible it can be utilized for longitudinal validation. For each attack inferred at the honeypot, we can check whether it is covered by a mitigation provider attack event and one of its prefixes. More specifically, the victim is visible as the source of requests at the honeypot and the destination of attack traffic at the DoS mitigation sensor.

Quality of the baseline. To evaluate the precision of thresholds that are used to detect amplification attacks at honeypots, ground truth data is necessary. Such data has to be created independent of the honeypots since choosing one honeypot as a point of reference for multiple honeypot platforms will lead to ambiguity for two reasons. First, each honeypot platform depends on thresholds. Second, no single configuration can be selected as the better reference point without attack event verification. Unfortunately, there is no public source of ground truth data for DoS-victims and attack events. Such information is often considered private and may inflict unexpected cascading effects, *e.g.,* a victim might experience a loss of customers due to a decreased trust in its systems, or other attackers might be encouraged to launch follow-up attacks on weakened systems. Furthermore, a complete view of DoS attacks is difficult to obtain, because even with large honeypots, attacks often only use a very small subset of reflection-capable systems. So although research-based methods to observe DoS attacks are documented publicly, their inferred list of victims often remains private or limited due to vantage point bias.

Companies, such as our data provider, offering DoS traffic mitigation services and equipment are in a unique position to identify DoS victims. These mitigation providers typically operate on the aggregates of traffic paths and relay points (*i.e.,* routers), observing traffic en route rather than having to reside in an endpoint that may or may not be involved in an attack.

These aggregate observation points have the advantage of scale, with the ability to observe and correlate attack events across an array of covered systems and networks. Mitigation providers typically have aggregate traffic sensors deployed at a variety of customer sites. Anomalous traffic that is detected can be reported, and may eventually trigger automatic mitigation such as blackholing [109] or traffic scrubbing [66]. Although such mechanisms are also based on heuristics in practice, operational data based on such mechanisms produces a confirmed set of victims due to its immediate mitigation actions. In practice, a detected attack (*i*) triggers a report that alerts the customer and optionally (*ii*) activates an automatic countermeasure to protect the target from the attack. False-positives would lead to unhappy and fewer customers, especially because some mitigation services charge by the volume of traffic sanitized. Also, false-negatives would be reported by the customer (since its service still experiences quality degradation because of DoS traffic), which ultimately leads to fine-tuning of thresholds and better detection.

We call our data baseline for two reasons. First, during our measurement period, no customer complained about false positives, so we believe that the detection accuracy is very high. Second, we also believe that this data set provides a representative visibility into attacks because the DDoS mitigation company has a 22% market share, and its customers are internationally and topologically (small, medium, large networks) distributed.

Given that the events included in our baseline data set are attacks, honeypot platforms claiming complete coverage should be able to detect these events (and maybe more).

2.6 Detecting Attacks

Attackers unwittingly use amplification honeypots as reflectors to conduct attacks. This helps honeypot operators to observe and quantify attacks. To distinguish attack packets from scanning and general Internet background radiation (IBR), honeypots group packets into ”flows” using a flow identifier (id). *Attack thresholds* then identify flows that likely belong to an attack.

Flow ids can be created using commonalities among packets such as the combination of source/destination address and source/destination port pairs. Traditional Internet applications minimally use a five-tuple flow id (IP protocol, address pair, port pair) to group flows, but all fields in the IP header, UDP header, and abused protocol could be used. Minimizing the number of flow id fields while correctly classifying all packets in a group maximizes efficiency.

In a reflective attack, the request packets an attacker sends will contain a spoofed source address. The spoofed address becomes the destination (victim) for amplified response packets. This is achieved by handcrafting packets, which requires the attacker to set all fields to protocol-conforming values. Attackers may randomize field values that may vary by operating system or at run-time, such as the IP ID field or UDP source port, in order to complicate packet classification at the honeypots.

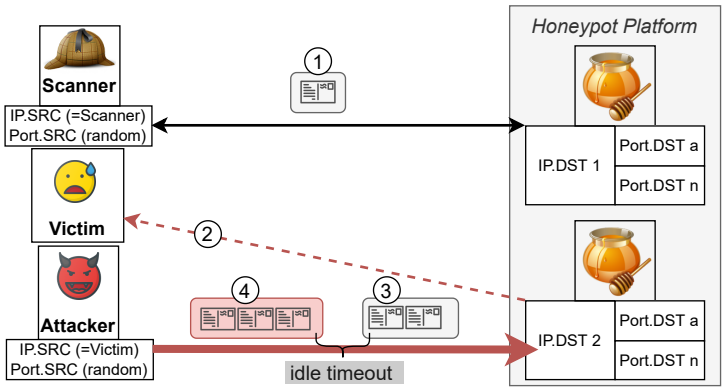


Figure 2.2: Overview of flow identifiers, timeouts, and packet loads to split a train of packets into flows (rounded rectangles) and attacks (*e.g.*, ≥ 3 packets per flow, red rectangle).

Among packets to a honeypot the flow-id of a typical UDP-based amplification attack requires, (*i*) a spoofed source address associated with a victim, (*ii*) a destination IP address of the amplifier (or honeypot), (*iii*) the destination port that maps to the abused protocol on the amplifier, and finally (*iv*) the source port, which can be chosen freely by the attacker. Note, carpet bombing attacks, which target IP prefixes as opposed to a single victim address, may spoof some portion of the most-significant-bits in a source address in order to randomize additional bits in the flow-id.

Other fields, such as the IPv4 ID or TTL can similarly be chosen at random or set to commonly used values to avoid raising suspicion. Research shows that some botnets use recognizable values for the source port, TTL, or DNS values [118]. For example, the ports 80 and 123 are often found paired with NTP (port 123) attacks [30], [76], [118] and make up more than 50% of the attacks together. Protocol specific observations show that source port selection differs among protocols [50]: attacks using CharGen, QOTD, RIP, and SSDP exhibit a hard-coded, stable paired port almost exclusively while NTP and DNS attacks show a larger range of randomized ports (about 50%). Overall, the selected source port in the request packets of an attack may be useful to track a specific pattern belonging to an attack entity, but is otherwise unsuitable as a more generic traffic classifier.

Figure 2.2 puts the flow identifier into context. A honeypot platform is built from multiple sensors that receive packets from a variety of sources such as scanners ①. The goal is to identify packets that are not just information gathering but used to attack victims ② via reflection attacks. Packets in the same flow-id can then be grouped together based on an *idle timeout*, which determines the maximum interval between two packets belonging to the same flow ③ or to a different flow ④. Finally, only flows that contain a minimum *packet load* are considered *attack flows* ④.

We now introduce the various attack definitions from related work. See *Attack Thresholds* in Table 2.3. Note, attack definitions are independent from the deployments described in

Table 2.3: Most recent or commonly used amplification honeypot platforms, their setup, definitions of flows, and attack detection thresholds. For CCC, we show the median number of sensors since deployment.

Honeypot Project	Setup			Flow Identifier						Attack Thresholds	
	Sensors	Distributed		IP Prefix		IP Address		Port		Idle Timeout	Packet Load
	[#]	Geo	Topo	Src	Dst	Src	Dst	Src	Dst	[minutes]	[packets]
AmpPot [76]	21	✓	✓	✘	✘	✓	✘	✘	✓	60	≥ 100
AmpPotMod [118]	8	✘	✓	✘	✘	✓	✘	✘	✓	10	≥ 100
CCC [146]	65	✓	✓	✘	✘	✓	✓	✘	✓	15	≥ 5
NewKid Mono [52]	1	✘	✘	✓	✘	✘	✓	✘	✓	1	≥ 5
NewKid Multi [52]	1	✘	✘	✓	✘	✘	✓	✘	✘	1	≥ 2 ports & ≥ 5
HPI [50]	549	✓	✓	✘	✘	✓	✓	✘	✓	1	≥ 2 honeypots & > 20

Table 2.4: Expected outcome of different attack detection methods, in case of a uniform amplifier utilization and an attack load of 1 Gbit/s lasting 5 minutes.

Attack Configuration							Attack Detected			
UDP Port	Protocol	∼Request Size	Ampl. Factor	# Amplifiers	Reqs/Attack	Reqs/Amplifier	AmpPotMod	CCC	NewKid	HPI
17	QOTD	15 Bytes	140	31k	17.9M	576	✓	✓	✓	✓
19	CharGen	15 Bytes	356	30k	7.0M	234	✓	✓	✓	✓
53	DNS	37 Bytes	41	1.9M	24.7M	13	✓	✓	✓	✘
123	NTP	13 Bytes	557	2.3M	5.2M	2	✘	✘	✘	✘
389	LDAP	52 Bytes	63	8k	11.4M	1430	✓	✓	✓	✓
1900	SSDP	90 Bytes	31	1.9M	13.4M	7	✘	✓	✓	✘

Section 2.4. Data from any deployment can be combined with any attack-detection method. However, we use the names of the original publications to distinguish them.

2.6.1 Current Methods

CCC considers a flow as an attack flow if it contains at least five packets *per sensor* within an idle timeout period of 900 seconds. This is in contrast to AmpPot, which defines higher thresholds: An attack flow must contain at least 100 packets with an idle timeout of 3600 seconds or 600 seconds when observed across *all* sensors. CCC and AmpPot use the source address and the destination port to assign a flow id to multiple packets. CCC also considers the destination address, *i.e.,* the sensor, as an additional restriction to classify packets into an attack flow.

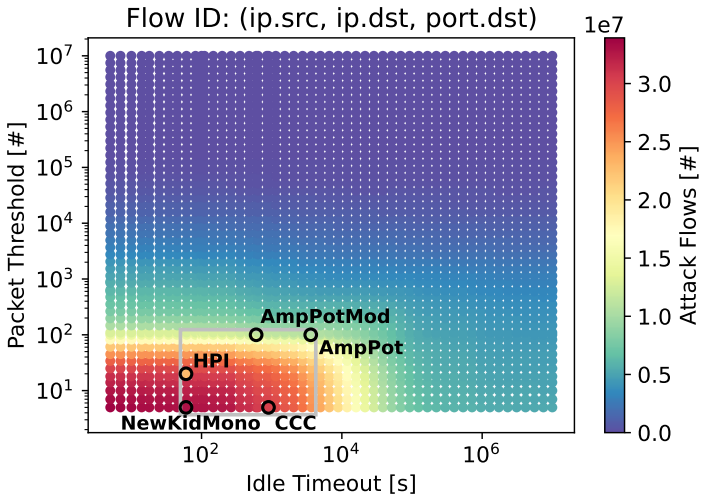
NewKid describes two types of attacks, monoprotocol and multiprotocol attacks. We use the labels NewKid Mono and NewKid Multi to distinguish them. Mono requires five packets in an attack flow with an idle timeout of 60 seconds. The Multi variant extends this definition to include packets that have at least two different destination ports within the idle timeout period. To account for carpet bombing attacks, *i.e.,* attacks hitting multiple addresses in the same IP prefix, the flow id uses the source IP prefix, instead of the address, combined with the destination IP address and, for Mono, the destination port. CCC is also able to infer carpet bombing attacks but only if 16 *individual* attack flows target victims in the same `/24` prefix.

HPI applies an idle timeout of one minute, a packet load of at least 20 packets, and requires activity observed by at least two sensors. Although their flow id is defined per-sensor, they require at least two overlapping flows.

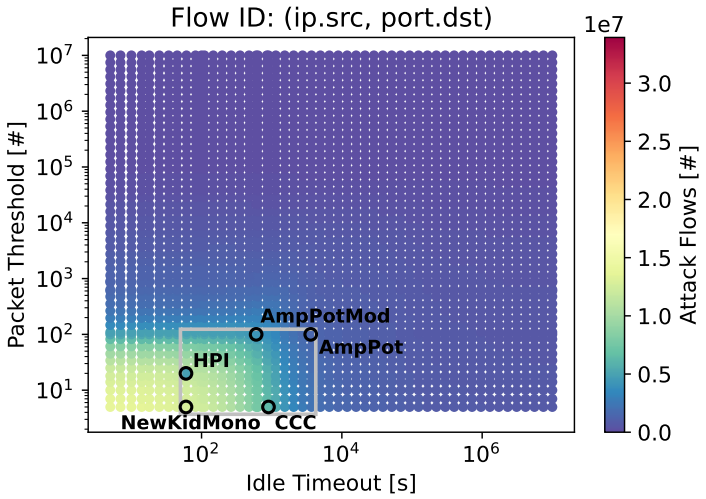
How to (not) present thresholds. We find a recurring pattern that attack thresholds are insufficiently justified. We acknowledge that rigorous thresholds are hard to identify without ground truth. Unfortunately, there is little to no discussion on the effects of the chosen thresholds. Documenting its effects is possible without ground truth and certainly would help the reader in future research.

The AmpPot paper includes a definition paragraph, specifying the minimum flow filter threshold, stating *sources [must send] at least 100 consecutive requests to our honeypots* [76]. The authors claim that this is a *conservative* threshold but do not provide further details on the reasoning or the number of events this configuration excludes. We believe it is based on their analysis of telescope traffic and the behavior of large-scale scanners contacting at least 64 dark addresses on the same port. They find that roughly 94% of the scanners send less than two packets per IP address on average.

In AmpPotMod, the authors reduced the idle timeout to *analyze attack duration with a more fine-grained approach* [118]. It remains unclear how this change affected their results, *e.g.,* the number of detected attacks.



(a) CCC flow identifier: per sensor.



(b) AmpPotMod flow identifier: per platform.

Figure 2.3: Number of attack flows, depending on different definitions of flow identifiers and attack thresholds. Thresholds from honeypot research (Table 2.3) are located in the gray box, HPI attack threshold marks fewer flows as attack.

CCC [146] selected their idle timeout *to loosely correspond with the availability of short lived attacks (under an hour) from booter systems.* However, they do not provide an analysis to validate their choice of threshold.

The NewKid paper illustrates that *thresholds were established empirically* [52] by manually analyzing three weeks of traffic. The authors infer three traffic classes (*slow*, *fast*, *bursty*) and try to classify the first class as scanner and the remaining classes as attacks. The description lacks detail on this inference and the *automatic classification* in particular. It remains unclear how the victim CIDR blocks are selected.

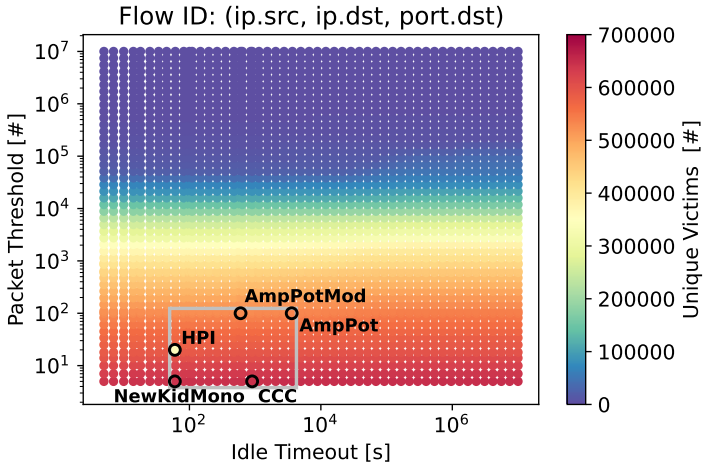
The HPI team states that they *experimentally derived that actors use up to 20 packets from the same source IP address* [50], but no further explanation is given about the experiment setup.

All papers include basic reasoning of the chosen attack thresholds. While the adjustable parameters are similar, the reasoning for different choices of flow id, packet load, and idle timeout remain unclear in practice. We highly encourage future work to use appendices to provide a more detailed analysis. This will enable the community to reproduce data selection processes and inferences.

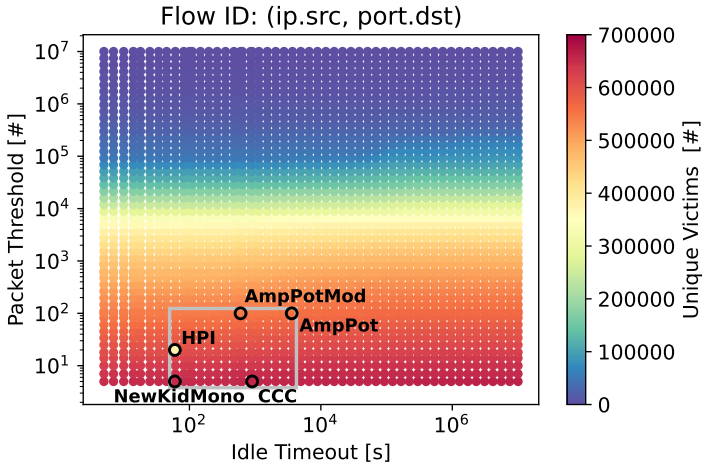
2.6.2 Comparability of Attack Thresholds

We now systematically analyze the effects of various flow identifiers and attack thresholds to assess the comparability across research projects. We distinguish between sensor-based and platform-based flow-identifiers. Although we include all threshold configurations from related work, we will focus on the CCC and AmpPotMod configurations. Their publications have a wide reach and they differ in a key aspect: the CCC flow identifier is applied per-sensor whereas AmpPotMod is applied per-platform. Please note that we do not use ground truth data but rather explore the effects of the configuration spectrum. The dataset contains packets obtained by the CCC honeypot platform.

Counting attack flows misleads. First, we show the number of attack flows for different thresholds, see Figure 2.3. The heat map shows the number of identified flows on the z-axis as a function of the idle timeout in seconds (x-axis) and the packet threshold (y-axis). The maximum x-axis value is around 10^7 which correlates to the complete measurement period of the dataset. A grey square marks the area for the thresholds listed in Table 2.3. The left figure uses the CCC flow identifier, *i.e.,* source address, destination address, and the destination port applied per sensor, whereas the right figure uses the AmpPot flow identifier, *i.e.,* source address and destination port applied across the whole platform. For NewKid, we only show the Mono variant because it was predominantly used in the paper. The value for the HPI thresholds is visually striking, because we additionally include the requirement of at least 2 honeypots sensors for this data point.



(a) CCC flow identifier: per sensor.



(b) AmpPotMod flow identifier: per platform.

Figure 2.4: Number of victims, depending on different definitions of flow identifiers and attack thresholds. Thresholds from honeypot research (Table 2.3) are located in the gray box, HPI attack threshold marks less hosts as victims.

We infer two findings: (*i*) The platform-based flow identifier counts less attack flows because it groups packets across different sensors into the same flow instead of counting the flows per sensor – provided attacks utilize multiple sensors. We find 12.9M attack flows with AmpPotMod thresholds and 30.3M with CCC thresholds when applied to the per-sensor flow ids (Subfigure 2.3(a)), and 4.4M and 6.4M attack flows when applied to the per-platform flow ids, respectively (Subfigure 2.3(b)). (*ii*) Longer idle timeouts only affect the attack flow count up to $\sim 10^4$ seconds (3 hours), but have negligible effect thereafter. At that point short consecutive attacks are likely grouped into a single flow. The idle timeout has a stronger effect on the per-sensor flow identifier because it is less likely to observe packets at the same sensor.

Detected victims uncover high similarity. We now analyze the number of detected victims, see Figure 2.4. The figure uses the same x and y-axis as Figure 2.3 but shows the unique victim count on the z-axis (the maximum is two orders of magnitude lower). Instead of counting attacks or attack flows—which are heavily influenced by the choice of flow identifier: per-sensor vs per-platform—we count the number of victims. Since both approaches are run on the same data, measurements are comparable. Note, that the number of victims should be a lower bound of the attack numbers. The idle timeout still affects results as a long idle timeout might group packets from low volume scanners into attack events, thus potentially generating victim artifacts.

For the per-sensor flow identifier, we find 644k victims using the CCC threshold and 531k victims using the AmpPotMod threshold. For the per-platform flow identifier, we find 654k and 549k victims, respectively. By comparing the respective configurations (CCC flow identifier and CCC thresholds versus AmpPotMod flow identifier and AmpPotMod thresholds) we find only a difference of 15%. Visually, both configurations are present in the same cluster and gradient. Reassuringly, this means that the results of the various honeypot platforms are indeed comparable. An exception to this finding are the HPI thresholds, which require at least two sensors to observe attacks. This leads to a 45% smaller victim set.

2.6.3 Evading Threshold-based Detection

Current studies (see Section 2.4) apply a single threshold configuration that is independent of the misused protocol. The CCC honeypot detects NTP (60%), LDAP (31%), and DNS (4%) as the most popular amplification protocols in 2022. This observation confirms common expectations, which assume attackers choose protocols that allow for high amplification and provide a rich amplifier infrastructure. NTP, for example, does not only provide the highest amplification factor and many amplifiers but also *mega-amplifiers* [30], *i.e.,* hosts that exhibit a significantly larger amplification factor due to their configuration, making this protocol most appealing to attackers.

The use of a protocol-independent threshold is surprising, though, since each protocol exhibits features (*i.e.,* amplification) and deployment (*i.e.,* instances in the wild) that may be leveraged

by attackers in different ways. We now analyze the attacker potentials to impede detection by honeypots. We do not use honeypot measurements but model a realistic attack volume and utilize protocol properties as well as public knowledge about the number of deployed amplifiers—similar to what attackers can do.

We assume a simple attacker model: attackers try to minimize exposure by reducing the load per amplifier while still achieving a desired traffic load. Given a set of amplifiers and a target attack load, an attacker uniformly distributes connection requests among all amplifiers. Based on this model, we infer the number of expected packets per amplifier for an attack load of 1 Gbit/s lasting 5 minutes. This attack scenario is realistic and produces more traffic than the majority of attacks: (*i*) Although new attack traffic peaks are reached yearly, the majority of attacks (98%) do not exceed 1 Gbit/s, even in the year 2021 [177], [218], [251]. (*ii*) A recent honeypot platform observes that 50%–80% of amplification attacks are shorter than 5 minutes [50], depending on the protocol. Triggering high volume attacks by requesting relatively little from a large number of amplifiers is doable given current amplification factors and deployment of amplifiers (see, *e.g.,* NTP or SSDP). We adopt amplification factors from related work [133], the numbers of open amplifiers from publicly accessible scan projects [273], [279], and then apply common attack detection thresholds.

Table 2.4 lists the calculated attack configurations and compares them against the attack thresholds presented in the AmpPotMod, CCC, NewKid, and HPI papers. Depending on the amplification protocol each honeypot sensor would experience different packet loads, ranging from 2 (NTP) to 1430 (LDAP) packets during the attack time. Attacks that require fewer requests per amplifier tend to remain unnoticed by current detection methods. This result highlights that current detection methods may miss smartly tailored attacks and that thresholds can best detect attacks when the packet load per amplifier is high. Overall, this suggests that the honeypot observations are incomplete.

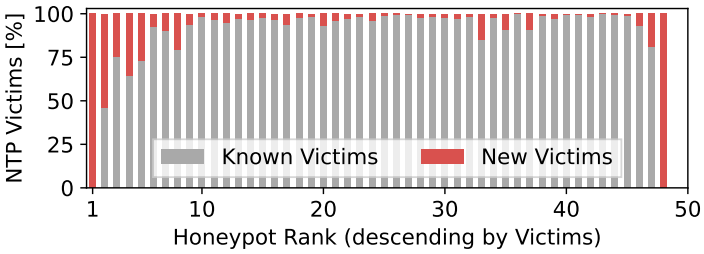
We conclude that honeypot observations cannot be simply explained *in situ* but have to be embedded into the protocol ecosystem and the decision-making that determines amplifier lists used by attackers.

2.7 Honeypot Convergence

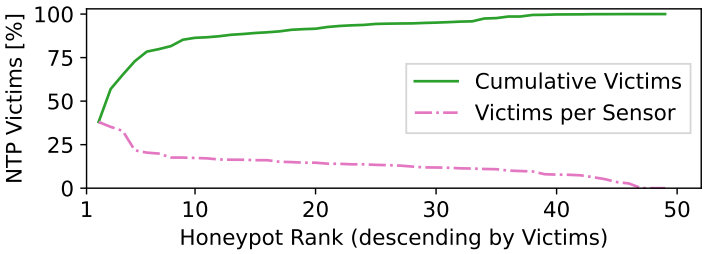
In this section, we revisit accuracy estimations for observations from a distributed honeypot platform. We explore the notion of *honeypot convergence*, a completeness measure of the detections that is influenced by the number of honeypots deployed and their configuration. We evaluate the impact of varying deployment scenarios based on the CCC platform.

2.7.1 Current Methods

Honeypot convergence is based on the assumption that the observed event set stabilizes (*i.e.,* converges) as the set of honeypot probes varies. It is considered a key property of a honeypot plat-



(a) New victims per sensor.



(b) Victims per sensor and cumulative total.

Figure 2.5: Convergence behavior for NTP using a near-optimal selection of honeypot sensors.

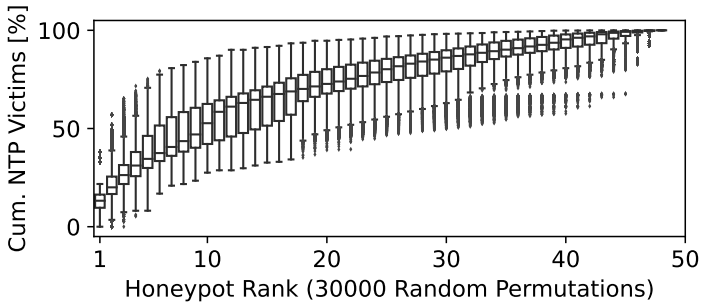
form, because it provides a comparative measure for attacks observed by different honeypot deployments. Convergence *supposedly* occurs when a set of honeypot probes provide a complete view of all attack events.

In the AmpPot paper [76], the authors order all the honeypot probes by name and then compute the running sum of new attacks contributed by each probe in turn. They conclude that 10 AmpPot probes identify $> 90\%$ of all attacks and that additional probes add only very few new attacks.

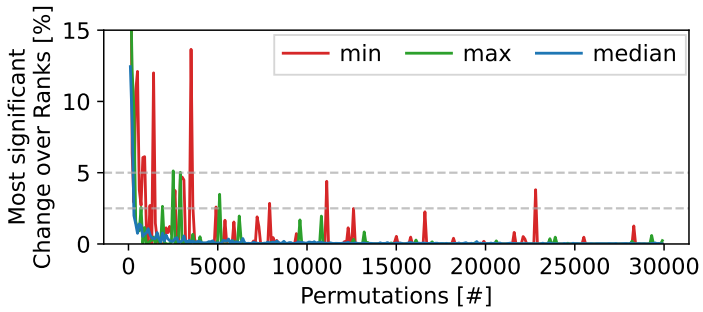
In the CCC paper [146], the authors apply a capture-recapture analysis, a statistical method known from ecology, which derives the number of estimated attacks by random sampling of the honeypot probes. They conclude that the CCC platform captures 85.1%–96.6% of all attacks. Other work derives that already 5 CCC sensors converge and monitor $> 99.5\%$ of the DNS victims [111].

Although the number of sensors is significantly higher ($\sim$150), the authors of the HPI deployment claim to have a complete view also on the basis of convergence behavior.

The stabilization of attack events (*i.e.*, convergence) when adding more probes is a common justification for specific honeypot settings. It remains premature, though, to conclude from convergence that a complete set of attack events has been observed. Convergence also occurs if a large set of attack events never enters the honeypot platform. Recent research observes this for different honeypot deployments, which show very diverging event sets with incomplete pictures of attacks. Two independent studies show small overlaps of only 4% [111] and 8.18% [70]



(a) Convergence: High variances in the results suggest that convergence is less stable than previously assumed.



(b) Relative differences of min, med, and max of detected victims. Even at ≈25k, worst-case results (min) differ by less than 2%.

Figure 2.6: Examining the convergence of NTP over 30k permutations.

between UDP amplification attacks observed at common honeypots and different vantage points (*i.e.,* other honeypot platforms and IXPs), challenging previous assumptions and claims of convergence. Furthermore, analyses based on the large HPI platform show that convergence differs by protocol and that a general approach to high attack visibility (*i.e.,* 99%) is hard to achieve, *e.g.,* RIP measurements require 60 sensors and other protocols ≈150 sensors [50].

Reviewing the implications of honeypot convergence is important because this measure has been used as a fundamental building block for the justification of honeypot results. Given the visibility of a honeypot platform, researchers had no other means but to test for the convergence of their results. We argue, however, that honeypot convergence should be re-interpreted, as it is only a fair measure of the limits of visibility, *i.e.,* a test whether the horizon of the platform has stabilized.

2.7.2 Reproducing Convergence

We use data from the CCC honeypots (see Section 2.6) to illustrate that the strategy of selecting probes has a significant impact on convergence results. Using the default CCC thresholds,

we learn about 1.4M attacks towards 644k victims during our measurement period spanning 3 months. The most common protocols for amplifications are NTP (60%), LDAP (31%), and DNS (4%). We observe continuous scans or attacks for all but one faulty sensor for NTP and LDAP, why we conclude that these services were run throughout the whole measurement period.

We now reproduce the honeypot convergence based on a near-optimal sensor selection, analogous to prior work [76], [111]. We sort the sensors by the number of victims and perform a greedy selection, *i.e.,* we select the sensors with the most unique victims first. Figure 2.5 exhibits the results for NTP, for LDAP we refer to Appendix 2.11.1. The share of new victims, which an individual sensor contributes, decreases quickly until rank 10. For NTP at rank 10, 87% of victims have been already observed and the subsequent sensors do not significantly increase the cumulative count although each sensor observes ∼16.5% of all victims. For LDAP at rank 10, we observe slightly fewer victims (76%). Each additional sensor observes ∼35% of all victims but increases the cumulative share only by 0.5%. In summary, we successfully reproduced the honeypot convergence for the given platform and measurement period.

2.7.3 A Fair Convergence Introspection

This convergence measure, which we just reproduced, follows a probe sampling that prefers sensors with a large number of common victims. As such, it is biased towards fast convergence. We now want to analyze the general convergence behavior and answer the question whether this bias leads to missing relevant data from the result set.

In general, the convergence behavior depends on the number *and the order* of considered sensors. To eliminate order bias, we create 30k random permutations of all CCC sensors and re-inspect convergence for NTP, see Subfigure 2.6(a). This analysis differs from related work [50] by exploring further statistical details instead of only averages. Each box includes the median (bar), up to 1.5× of the interquartile range (whiskers), and all minimal and maximum values (outliers). This plot clearly visualizes the large variances across convergence results, depending on the combination of probes. Considering the best (upper outliers) and worst (lower outliers) case scenarios of 20 sensors (rank 20), we find 39%–95% of NTP victims. Furthermore, the upper outliers resemble very closely the cumulative ratio of victims in Subfigure 2.5(b). These observations lead to two insights. Fist, they confirm our previous observation that probes with higher weight (*i.e.,* more attacks) introduce a bias towards fast convergence. Second, they emphasize that convergence measures should be utilized with great caution when justifying the completeness captured by honeypot deployments.

We still want to justify that we do not compute *all* permutations of currently 50 CCC sensors due to numerical complexity ($50! \approx 3{\cdot}10^{64}$ permutations). Limiting to 30k permutations already shows stable results. To assess the stability, we iteratively create 100 new permutations and add them to the total set of permutations. For each set of permutations, we determine the largest differences of the minima, median, and maxima of detected victims across all ranks.

The results are shown in Subfigure 2.6(b). After an initial phase of significant changes, the median becomes very stable using at least 25k permutations. Occasionally, minima can change up to ∼2%, *cf.* Appendix 2.11.1 for LDAP.

It is noteworthy that the capture-recapture method can be inadequate for estimating an unknown population. Related work finds that (*i*) accuracy depends on capturing a large proportion of the population [84], *i.e.,* the majority of attacks, and (*ii*) it looses accuracy for transient populations [147], *i.e.,* when attackers cease or move between measurement areas due to new amplifier lists. All this makes it very likely that the estimated number of attacks accounts only for a subset of total attacks.

2.7.4 Convergence versus Completeness Metrics

In the previous sections, we have shown that convergence is not a stable metric but (if cautiously applied) can shed light on the horizon of visibility for a honeypot platform deployment. The completeness of the observation (*i.e.,* the detection of all ongoing attacks), however, strongly depends on how an attacker selects the amplifiers. Consider two corner cases and one likely scenario.

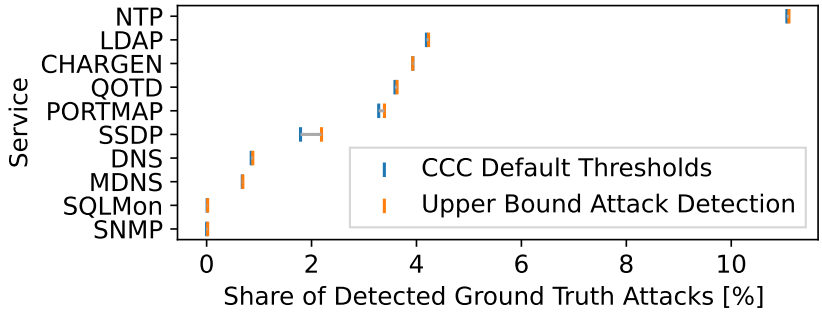
1. An attacker may not select at all but send spoofed requests to arbitrary IP addresses. In this case, the probability of observing the attacker is extremely low for any given honeypot platform.

2. An attacker may—after scanning—use all amplifiers of a given protocol. In this case, a single sensor suffices for detecting the attack.

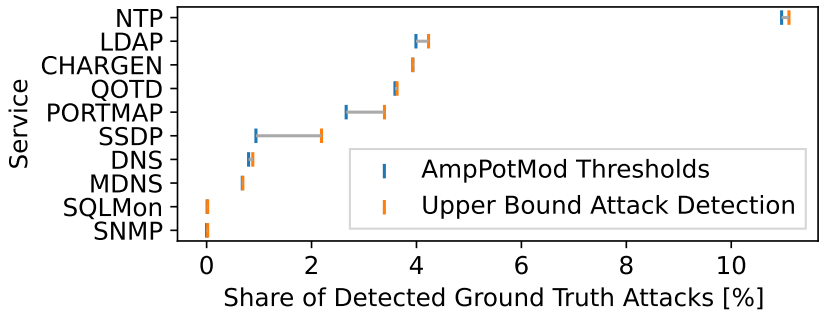
3. An attacker may use a limited subset of amplifiers, *e.g.,* an amplifier hit list. This list may have been collected according to efficiency (*i.e.,* amplification factors), (geographic or topological) locality, or other means. In this case, the probability of detecting the attack strongly correlates with the honeypots conforming to the selection criteria.

Amplifier hit lists may be static, in which case the attack remains invisible if no honeypot is on the list, or dynamic. In the latter case, honeypots may observe scanning and respond accordingly. Honeypots typically expose a low amplification factor due to legal reasons, which makes them less attractive in many attacks.

Finding a good completeness metric. Often, honeypot platforms have a limited diversity in geography or network topology. A valid metric for estimating the completeness of honeypot observations needs to infer global knowledge from local observations, which is the more challenging the smaller and less diverse local observatories are. Preferably, such metric can at least provide a rough estimator of the error inherent to the measurement system. As we have seen in the previous discussions, such an indicator cannot be extracted from the pure measurement set



(a) CCC.



(b) AmpPotMod.

Figure 2.7: Honeypot probes detect at most 11% of the ground truth attacks. There is no room for fine-tuning the thresholds to improve the detection rate, because the probes simply do not observe more events for the victims.

alone. Instead, orthogonal sensors and correlating analyses are needed to capture and quantify the invisible attack data.

An obvious source of control is to compare with alternate measurements such as flow data, network telescopes, or public attack reports. For research that needs to exclusively base on the honeypot platform, we conjecture that additionally observing and analyzing explorative scanning (possibly with varying reply behavior) as well as correlating initial scanning with subsequent attack detection (or not), may open a new angle of view on the completeness of the honeypot attack data.

2.8 Completeness

Using additional data sources, we find that honeypots are unable to observe anywhere near a complete view of real-world attacks, but are quite good at detecting scanning activity.

2.8.1 The Honeypot View is Mostly Incomplete

Similar to honeypots, our DDoS provider data shows that DNS (60%), NTP (23%), and LDAP (8%) are the most popular protocols misused for amplification. Leveraging this real-world baseline data, we can now independently assess whether the honeypots grant a reasonably complete view on attacks. To this end, we detect attacks using the honeypot data and the default CCC or AmpPotMod thresholds. Then, we calculate the share of overlapping attack events in the DDoS provider baseline data for each protocol. The results are visualized in Figure 2.7. We find very limited overlap, *i.e.,* honeypot views on amplification attacks are mostly incomplete. For the best performing protocols, for which we confirmed uninterrupted operations and convergence in the previous sections, we only observe 11% (NTP) and 4% (LDAP) of attacks. This is in stark contrast to current convergence measurements [50], which suggest that we should observe at least 90% of NTP attacks with 50 sensors. Our results, however, comply with recent findings (4% [111] or 8.18% [70]), which examine the overlap between honeypots and IXPs, but based on baseline data. We acknowledge that our baseline data is limited to those networks that share attack alerts with the DoS provider. Nevertheless, we want to stress two important details. (*i*) our data represents a fairly large share of the market (up to 22%) and (*ii*) for a complete coverage of all attacks, honeypots should *at least* observe most if not all of our baseline attacks.

Notably, the relative popularity of DNS differs between honeypots (see paragraph above) and other vantage points such as IXP-based measurements (DNS 43%, NTP 25%, LDAP 20% [70]) and our baseline data (DNS 60%, NTP 23% and LDAP 8%)[3]. We argue that honeypots miss a substantial portion of DNS attacks for two reasons: (*i*) DNS amplifiers have the highest churn rates [80], [81], which makes it necessary for attackers to rescan frequently. Hence, attackers can easily rotate between amplifiers and prefer new amplifiers [111]. (*ii*) Although the DNS ecosystem consists of various amplifiers [112], the driving factor for amplification are queries for names with large zones. This means that the attackers can utilize most amplifiers if they select such a name, which makes the honeypots less attractive or at least less likely to be used. This is supported by the fact that DNS has the slowest convergence [50].

2.8.2 No Potential for Better Attack Thresholds

We ask whether we can fine-tune the thresholds to improve results. For this, we infer the upper bound of attack detection. We use the most permissive thresholds, *i.e.,* every event is classified

[3]In order to avoid confusion, we restate that the set of the three protocols {DNS, NTP, LDAP} represent the most common protocols misused for reflective amplification across the three vantage points. However, the ranking order of these top protocols differs for the honeypots:

CCC Honeypots	NTP (60%)	LDAP (31%)	DNS (4%)
Baseline Mitigation Provider	DNS (60%)	NTP (23%)	LDAP (8%)
IXP [70]	DNS (43%)	NTP (25%)	LDAP (20%)

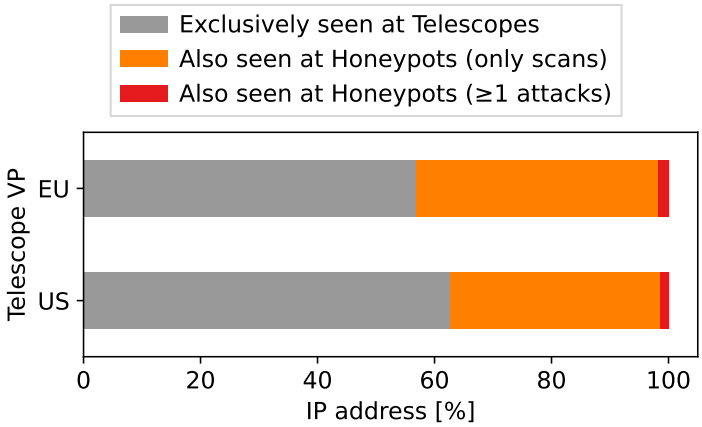


Figure 2.8: Hosts scanning our telescopes and connecting to our honeypot platform. CCC thresholds mainly infer scanning events, indicating successful scan event detection.

as an attack. This potentially leads to many false positives because even scanners sending just one packet to the honeypot platform will be interpreted as an attack.

We visualize the results in Figure 2.7. The grey horizontal lines indicate the potentials for improvements. We find that we cannot significantly fine-tune the thresholds because the honeypots simply do not observe any event for the victims in our baseline data, *i.e.,* there is no packet that relates to any of the IP addresses under attack.

This limited potential suggests that optimizing the thresholds would lead to overfitting with respect to our baseline data set. Also, such thresholds would only be optimal for a particular point in time and probably lose the acquired precision in the long term.

2.8.3 Misclassification of Scans

We now utilize network telescopes to assess whether attack detection thresholds for honeypots successfully eliminate scan events.

Telescopes and honeypots observe the same scanners. At our telescopes, we identify all scanners contacting service ports supported by the CCC honeypots. During our main measurement period, we find 27k unique scanner addresses in the US and 16k in the EU. We now check whether these addresses have been observed at the CCC platform. The CCC honeypots observe 37.4% of the US scanners and 43.1% of the EU scanners. Not all scanners are observed since not every scanner performs a complete address space scan, *e.g.,* it is part of a botnet or pool which splits the address space, or operates very locally [53].

Telescopes and honeypots agree on scanners. We now apply the default CCC thresholds, inspect the events caused by scanners, and compare the results by region in Figure 2.8. Strikingly, 36% of US scanners only triggers scan events at the honeypots. Likewise, the honeypots infer attacks only for 1.4% of the scanners. At the EU, we observe similar trends with 41.3% of the sources performing scans only and 1.8% triggering attacks. We repeat this analysis using the

AmpPotMod thresholds and find comparable results (not visualized). However, AmpPotMod thresholds detect slightly fewer attacks. The share of addresses performing attacks decreased for both vantage points, the US (1.4% → 1.1%) and EU (1.8% → 1.4%).

Please note that the detected attacks are not necessarily misclassifications. Upon receiving a response, a scanner might start testing the capabilities of a honeypot, which triggers an attack event. But such behavior is rather unlikely because scanners try to remain under the radar in order to avoid being blacklisted and to help discover as many victims as possible. Overall, we find that both threshold configurations are successful in exposing scan events as such.

2.9 Network Access, Economic Considerations

2.9.1 Network Types and Service Proximity

Honeypots can be deployed in any type of network with public reachability. Similar to the various threshold configurations, the effect of different network access types is little understood. For the large amplification honeypot platforms, we typically see sensors placed on eyeball, hosting, and academic networks. Still, we miss discussion on how the observations differ across network types.

Quantitative and qualitative differences have been shown for non-amplifying honeypots placed in mobile network service providers, darknets, and academic networks [287], *e.g.,* only a few topological Internet-domains have started to place dedicated focus on attacking mobile networks. For example, malware and scanners have been shown to limit their operations geographically and topologically [53]. Such differences observed across network types must also be anticipated for amplification honeypots.

Since many open services disappear because of IP churn [81] and not because they were taken down, it is beneficial to periodically rescan the network to update the service-to-address-mappings. According to [69], scan traffic can be reduced by 25-90% while missing only 1-10% of the population. This means that attackers utilizing such optimization will more likely discover and misuse honeypots that are in proximity to other amplifiers.

Cloud providers share their physical infrastructure through the use of virtualization. Outages and the mitigation of (unrelated) attacks) on shared infrastructure may affect honeypot measurements and thereby attack detection. Therefore, researchers need to pay close attention to the fate-sharing risks and factors of an otherwise well-functioning honeypot system.

2.9.2 Economic Considerations

Attackers misusing amplifiers are often operating in pursuit of economic goals. For instance, fee-based booter (or stresser) operators sell DoS attacks as-a-service and have been linked to the misuse of open amplifiers [79], [137]. Booter operators run websites where any individual

can purchase attacks [136]. Although some DDoS-as-a-service websites have been shown to utilize the same set of amplifiers, for most operating websites the overlap is minimal [137].

Researchers have used booter services to attack their own infrastructure and found attacks utilize on average 346 amplifiers from 27 autonomous systems [71]. Honeypots observed only ∼40% of DNS self-targeted attacks [79]. Overall, booter services are responsible for a significant number of amplification attacks, *e.g.,* 26% of DNS and 13% of NTP attacks were linked to a specific set of booters [79]. This means that observations by honeypot sensors can be extremely biased if they are used by a specific booter. Furthermore, take downs of booter websites can reduce the number of observed attacks [26], [71] and negatively bias the attack landscape perceived by a honeypot system.

Unfortunately, little can be done to influence the selection process of attackers. Obviously providing potent amplifiers helps, however, this opposes ethical measurements which deploy rate limiting. Therefore, special care has to be taken while analyzing significant peaks and drops, *e.g.,* for number of attacks for a specific protocol. Variations in attack detection may rather be caused by a specific booter omitting the honeypots from active use rather than a reflection of aggregate attack event trends.

2.10 Discussion

Why our results differ. Our results on the completeness of honeypot observations clearly differ from past research, indicating that honeypot systems miss a substantial share of all Internet-wide attacks. We identify two major reasons for the differences: First, honeypot observations, especially for early deployments, show a very fast convergence, which was misinterpreted as an indicator for completeness. Convergence, however, can only serve as an indicator for cost-efficiency of a particular deployment. Second, the access to orthogonal vantage points, *e.g.,* commercial on-path mitigation appliances, is rare and regulated by NDAs. By closely cooperating with a DDoS mitigation provider, we designed a method that evaluates the completeness of honeypot observations but still respects data privacy.

Following our systematic approach, we believe that our results exhibit *a more trustworthy* view on the amplification ecosystem. This is because (*i*) we do not select a singular configuration but explore complete threshold spaces and analyze convergence after random permutations of the sensor order, and (*ii*) our completeness results are bolstered by a curated DDoS attack list from a major mitigation provider.

Our limitations. Our results are based on a dataset that was gathered recently and covers a specific time period. Not all datasets that were used in prior publications were at our disposal; hence we could not evaluate some of the prior research against our baseline data. Even with the data constraints, however, we were able to use the configurations of various publications to compare detection properties of multiple honeypot thresholds. Furthermore, we argue for our finding that honeypots capture only a limited part of the global attack landscape since other

Table 2.5: Convergence does not implicate but can co-occur with completeness, depending on the attacker behavior.

Conv.	*Compl.*	Scenarios	
		Honeypot Observations	Reflector Selection
✘	✘	Each sensor captured different attacks.	Some attackers did not use the honeypots.
✓	✘	Multiple sensors captured the same.	Some attackers did not use the honeypots.
✘	✓	Each sensor captured different attacks.	All attackers used the honeypots.
✓	✓	Multiple sensors captured the same.	All attackers used the honeypots.

work [70], [79], [111] has raised similar concerns, while using complementary vantage points and time periods.

Convergence vs. completeness. Although convergence does not indicate completeness, both properties can occur at the same time. In Table 2.5, we depict examples under which conditions these properties occur. Attackers who are able to detect honeypot sensors as their targets, *e.g.,* due to rate-limiting at honeypots, can decide to never use them as amplifiers. This impedes completeness. At the same time, attackers that repeatedly use the same honeypot sensors for different attacks foster convergence since they add additional weight to the frequency of occurrence. This illustrates that deploying more honeypots sensors, even with a diverse geographical and topological distribution, does not necessarily lead to more reliable results. Instead, a thorough understanding of the attacker decision making is essential.

Honeypots are useful. It is discouraging to detect only up to 11% of attacks, in particular when facing the costs of deploying (renting cloud servers, buying dedicated hardware *etc.*) and maintaining amplification honeypots. Even though honeypots lack a complete view on the attack landscape, knowing this imperfection removes unwanted interpretation bias. We argue in favor of honeypot results as an important component of a larger complex ecosystem even if they are imperfect. We believe accepting this will help researchers to better interpret the observed phenomena and to understand *their fragment* of attacks. Since researchers are restricted to ethical measurements and hence rate limit honeypots, attackers will always be able to elude the trap.

Recommendations for attack definitions. A comprehensible, precise attack definition is essential for honeypot research. We find only textual definitions of attack thresholds in related work. Although these can be sufficient, they are ambiguous and open to interpretation. This is

why we recommend precise wording, preferably taken from common sources. A good candidate for this is found in the IPFIX specification [261], from which we adopted the usage of *idle timeout* and *flow*.

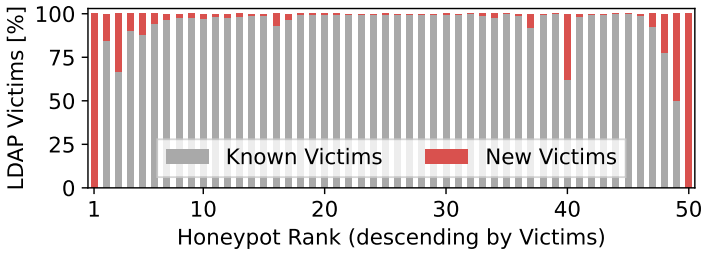
Overall, at least three definitions are required: (*i*) What identifies a victim? This could be a single source IP address, an IP prefix, an autonomous system, a name *etc.* (*ii*) What is an attack flow? One should clarify which flow keys are observed for flow inference and which thresholds are applied. (*iii*) What is an attack (event)? This is especially important for system-wide flow identifiers, when distinct attack flows towards the same victim are observed from different vantage points.

Directions for the future. Our results indicate the importance of extensive baseline data and ground truth. However, our community should not depend on it. Non-proprietary, auxiliary vantage points such as telescopes and correlating observations can also help to assess or improve the precision of measurements. We see such heterogeneous deployments in active use by commercial parties, *e.g.,* GreyNoise. Simply adding more honeypot sensors does not necessarily solve measurement challenges such as the honeypot convergence, which is among other potential obstacles caused by the decision making process of the attacker.

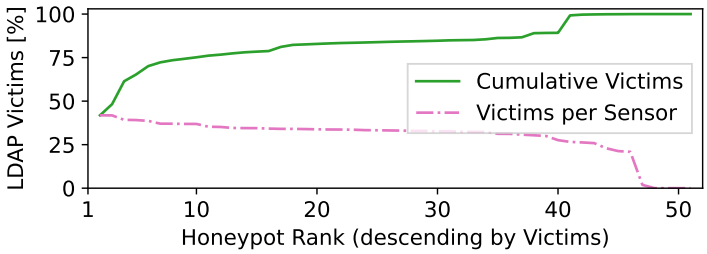
A fundamental problem for honeypot research is that aggressive scans exhibit traffic patterns similar to reflective amplification attacks. Conversely, low-volume attacks misusing relatively few amplifiers can resemble patterns of cautious scanners. Prior work was conducted based on the assumption that these phenomena do not intersect, but they do. This intersection can be illuminated by considering a careful definition and explanation of thresholds w.r.t. the observed data and the current amplification ecosystem.

The ever-changing ecosystem is the reason why we refuse to recommend a single best threshold configuration in this chapter. It is likely that any such recommendation will soon be obsolete as attacks and methods evolve. Additionally, our results indicate that even perfect attack classifications will not be able to detect all attacks. There is room for clarification on the impact of thresholds, and the correlation of minor events to make classification of various measurements easier. However, opportunistic classifications into obvious scans and obvious attacks are valuable.

With these considerations in mind, we go beyond just a call for comparable metrics. Given the same dataset, we need a way to compare the effects of different thresholds. We also encourage authors to present detailed analysis on their choice of attack thresholds. Finally, since the observation range of honeypots is directly related to being targeted by attackers, we argue that a future research agenda should include methods to replicate the creation of amplifier hit lists. Mimicking this part would complement our tool set and improve informed honeypot deployment.



(a) New victims per sensor.



(b) Victims per sensor and cumulative total.

Figure 2.9: Convergence behavior for LDAP using a near-optimal selection of honyepot sensors.

2.11 Additional Analysis

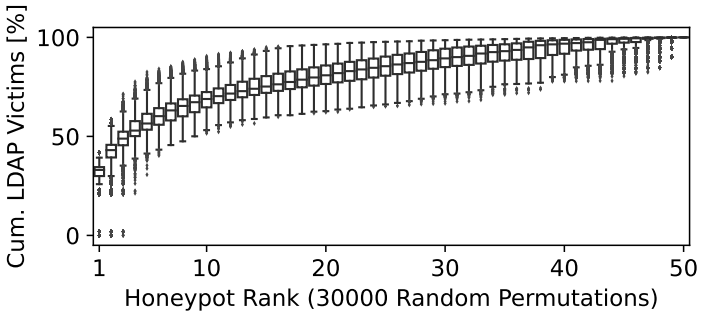
2.11.1 Examining Convergence for LDAP

In Figure 2.9 and Figure 2.10, we present detailed results of the convergence measure for LDAP. These results confirm that our convergence observations for NTP presented in Subsection 2.7.2 and Subsection 2.7.3 also held for LDAP.

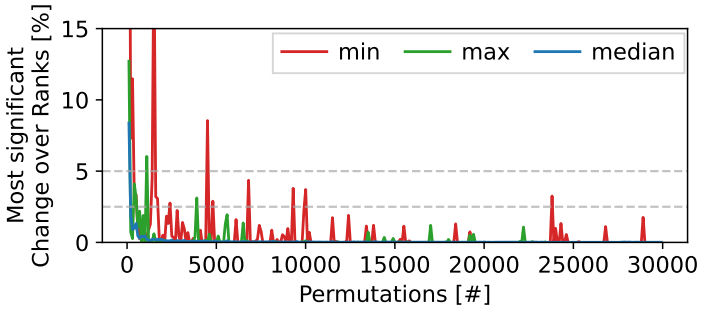
2.12 Conclusion and Outlook

In this chapter, we revisited methods to measure and infer reflective amplification attacks based on honeypots. We applied a data-driven approach that allowed us to challenge long-held assumptions. Using data from a large-scale honeypot, multiple network telescopes, and extensive baseline data from a leading DDoS mitigation provider, we were able to reproduce, confirm, or disprove common measures of attack detection, honeypot convergence, and attack completeness.

Contrary to popular belief, we found that (*i*) honeypot convergence has limited significance because it is a statistically unstable metric and (*ii*) observations by honeypots are incomplete, honeypots miss large fractions of ground truth attacks. We explored the complete spectrum of attack detection thresholds and embedded the thresholds of related work in our system.



(a) Convergences. High variances in the results suggest that convergence is less stable than previously assumed.



(b) Relative differences of min, med, and max of detected victims. Even at ≈25k, worst-case results (min) differ by less than 2%.

Figure 2.10: Examining the convergence of LDAP over 30k permutations.

Related work, although using different thresholds, largely produces comparable results but common thresholds cover only a very narrow part of the parameter space. We highlighted the various features that should be considered by researchers when deploying honeypots and analyzing data. These include setup properties, flow identifiers, and attack thresholds.

Our results underscore three open challenges. First, a well-defined definition of an attack, which accounts for traffic patterns observed by honeypots. Second, a reliable metric to assess the completeness of honeypot observations. Such metric should provide an error margin and not depend on external ground truth data. Third, to increase completeness, well-defined features that guide honeypot deployment. These might include deployments in heterogeneous network types, better protocol emulation, or sophisticated rate limiting methods. Most importantly, our community should gain a better understanding of the mechanics behind the creation of amplifier hit lists. Being able to reproduce the set of amplifiers used by attackers will allow researchers to tailor amplification honeypots in terms of location and behavior such that they will be targeted and capture a sufficiently complete view.

Chapter 3

The Far Side of DNS Amplification: Tracing the DDoS Attack Ecosystem from the Internet Core

Abstract

In this chapter, we shed new light on the DNS amplification ecosystem, by studying complementary data sources, bolstered by orthogonal methodologies. First, we introduce a passive attack detection method for the Internet core, *i.e.,* at Internet eXchange Points (IXPs). Surprisingly, IXPs and honeypots observe mostly disjoint sets of attacks: 96% of IXP-inferred attacks were invisible to a sizable honeypot platform. Second, we assess the effectiveness of observed DNS attacks by studying IXP traces jointly with diverse data from independent measurement infrastructures. We find that attackers efficiently detect new reflectors and purposefully rotate between them. At the same time, we reveal that attackers are a small step away from bringing about significantly higher amplification factors (14×). Third, we identify and fingerprint a major attack entity by studying patterns in attack traces. We show that this entity dominates the DNS amplification ecosystem by carrying out 59% of the attacks, and provide an in-depth analysis of its behavior over time. Finally, our results reveal that operators of various `.gov` names do not adhere to DNSSEC key rollover best practices, which exacerbates amplification potential. We can verifiably connect this operational behavior to misuses and attacker decision-making.

3.1 Introduction

Denial of Service (DoS) attacks pose a major, omnipresent threat to the stability of the Internet. About one-third of the active `/24` networks on the Internet received DoS attacks over a two-year period [64], and 90% of attacks mitigated at a large IXP involved reflection attacks [109]. To bring about reflection, attackers spoof source IP addresses to send request packets that supposedly originate from an intended victim, and abuse the infrastructure that replies to

these requests (*e.g.,* open DNS resolvers). Amplification is successful if the responses are larger than the requests.

The DNS is a core Internet component. It primarily operates over the transport-layer protocol UDP. Due to its stateless nature, UDP is particularly susceptible to spoofing, and at least 14 protocols that work on top of UDP allow for reflection attacks [133]. The Network Time Protocol (NTP) and DNS are (currently) the most-abused protocols [64], [70], [109].

Notably, amplification attacks are not limited to UDP. Poor implementations of network stacks allow attackers to use TCP as well [82]. A recent DNS amplification attack exploits inefficient resolver implementations and works regardless of the underlying transport-layer protocol [2]—DNS amplification remains one of the most popular attack vectors, despite recent changes such as DNS-over-TLS [199] and DNS-over-HTTPS [197].

Expert measurement methods are essential to observe global attack activities. Having a thorough understanding of attack dynamics and the abused infrastructure is crucial to effectively mitigate DNS-based attacks and to reduce the opportunity for infrastructure abuse. Several efforts exist to monitor amplification attacks on a global scale. Primarily, the monitoring infrastructures are implemented with the help of honeypots [76], [118], [146]. In such works, careful assumptions are made about the share of global attacks that honeypots account for [76], [146] because the amplification ecosystem consists of a large number of amplifiers [273] with high churn rates [81]. Moreover, sophisticated attackers learn about the location of honeypots and exclude them [260].

In this chapter, we extend the understanding of the DNS amplification ecosystem by jointly analyzing results from four complementary measurements, including the Internet edge and core. First, we introduce a method to infer DNS amplification attacks at Internet eXchange Points (IXPs). We exploit the central position of the IXP to comprehend abused infrastructure dynamics and explore opportunities to fingerprint attack origins. Second, we use a large, distributed honeypot platform to infer whether the IXP and honeypots observe the same set of attacks, and to investigate if attackers appear to exclude honeypots from attacks. Figure 3.1 visualizes our extended perspective on inter-domain DNS amplification attacks. Note that we anticipate attackers to abuse infrastructure that responds to DNS queries, which includes DNS forwarders and recursive resolvers. Third, we compare our observations with data from Internet-wide open resolver scans, allowing us to assess the extent of existing views on abusable infrastructure. Last, we consider comprehensive DNS measurement data to gain insights into the type of DNS infrastructure abused (*i.e.,* open resolver versus authoritative nameserver) as well as the amplification potential of attacks.

In detail, we address three key research questions:

Question 1 (Section 3.5). *Does an IXP-centric view contribute additional insights into DNS amplification attacks?*

As we will show, passive observations of DNS-based reflection and amplification attacks at an IXP can identify misused query names and abused infrastructure beyond honeypot-based

inferences. Surprisingly, with an overlap of only ∼ 4%, IXPs and honeypots detect mostly disjoint sets of attacks. In total, we find 24k new attacks over the course of 3 months, which were not observed by the honeypots.

Question 2 (Section 3.6). *Can we fingerprint outstanding attackers within the DNS amplification ecosystem?*

We fingerprint a larger attacking entity by correlating the use of `.gov` names and static DNS transaction ID behavior. The entity in question is demonstrably dominant and responsible for 59% of inferred attacks. Our data suggests two topological changes (*i.e.,* relocation) of the attacking infrastructure within one year, indicated by shifts in network layer observables. We observe that the entity frequently changes abused amplifiers. Moreover, we recognize patterns in misused query names that strongly suggest attempts by the entity to improve the overall amplification factors.

Question 3 (Section 3.7). *How efficient is the current exploitation of the DNS, meaning: (i) how are the amplifiers misused; and (ii) can the amplification factor still be improved?*

We are able to pinpoint the abuse of at least 10 to 1000 amplifiers in most events. Our results show that attackers mainly misuse legitimate `.gov` names in spoofed DNS queries, which is likely, because names under the `.gov` zone are DNSSEC signed. Bilateral clustering also shows that only 2% of attacks use static amplifier lists. 95% of the amplifiers for which we observe abuse are also found by a large-scale platform that scans for abusable infrastructure, which suggests that attackers use mostly well-known, publicly documented amplifiers. Nevertheless, we reveal that 2% of amplifiers are abused before they show up in public scan data, suggesting that attackers also employ alternative methods to find amplifiers.

Overall, our observations show that attackers exploit amplifiers effectively, and the turnover makes fine-grained source-IP filtering much harder. In spoofed requests, attackers also misuse query names that lead to significant amplification factors. After inspecting 440 million domain names in DNS measurement data, we detect only 9000 names with larger amplification potential. At the same time, our estimation of DNS response sizes for these names reveals that they could cause up to 14× more amplification. This shows that attackers do not fully exploit the DNS-based attack vector.

In the remainder of this chapter, we present background and related work in Section 3.2. We outline four viewpoints from the complementary measurements in Section 3.3, and introduce our DNS attack detection method for an IXP in Section 3.4. We then proceed to answer our research questions in Section 3.5–Section 3.7, summarize discussions in Section 3.8, and conclude in Section 3.10.

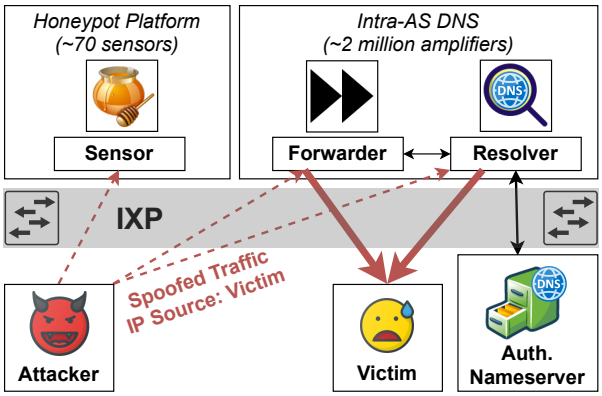


Figure 3.1: Vantage points and stakeholders of distributed, inter-domain DNS amplification attacks.

3.2 Background and Related Work

Reflective Amplification Attacks and Honeypots. Reflection and amplification attacks [266] are traditionally observed with honeypots [249], which apply straightforward thresholds to infer attack activity and to discern mere scanning for reflectors [76], [118], [146]. The advantage of using honeypots is that all incoming requests are likely part of attacks or scans since legitimate DNS services do not send DNS queries to those sensors. Honeypots, however, cannot infer the extent to which other infrastructures are involved (*e.g.,* public DNS resolvers) and are therefore limited in the assessment of general attack properties such as intensity. An additional challenge arises because the number of attacks visible to honeypots appears to converge quickly with only a few sensors deployed. Deploying more sensors does not necessarily increase the breadth of observation. This effect was shown with fewer than 10 sensors [76]. Thomas *et al.* [146] use a capture-recapture approach to estimate a 85%–97% visibility into UDP reflection attacks.

The research community has so far shown a tendency towards detection techniques for edge networks [96], [133]. We instead centre on IXP-based detection at the Internet core. Only NTP-based attacks have been studied at IXPs [71] by explicitly launching attacks via an attacking infrastructure. We focus on attacks in the wild and on DNS-based reflection, which requires a comprehensive detection mechanism. We also consider four Internet-scale, complementary data sources to investigate attack visibility and attacker behaviour. Our approach allows us to refute the common assumption that (sizable) honeypot infrastructures offer a near complete view on DNS-based reflection attack activity.

Recent prior work [70] started to compare attacks seen at an IXP and a honeypot using a flow- and volume-based DoS classification. The authors found little overlap (8.18%) between

both vantage points. Following up on this, we present the first in-depth comparison between various DDoS ecosystem viewpoints, precisely targeting DNS.

DNS Amplifier Ecosystem. A DNS infrastructure that responds to *all* incoming requests is prone to be abused for reflection. This includes resolvers, forwarders, and authoritative nameservers [6], [93], [112]. DNS is the second most-common amplification vector, although its amplification potential is ∼10× smaller compared to NTP [6], [30] and it has the highest churn in reflectors among protocols susceptible to reflection. Kührer *et al.* [80], [81] show that this is mainly caused by open resolvers in access networks, *e.g.,* home routers, where dynamic address allocation leads to the quick disappearance of about 50% of identified amplifiers, when indexed by IP address.

Anecdotal evidence suggests that attackers abandon reflectors once response rate limiting (RRL) is detected [291]. Due to ample availability of DNS reflectors (2.1M in 2021 [112]), RRL can be counteracted [133] by scaling. High reflector churn as well as RRL force attackers to maintain and frequently update sizable amplifier lists. This aligns with the observation that DNS exhibits the highest daily rate of unique scanners [146]. Exploiting our IXP-centric view, we follow the abuse of amplifiers over a three-month measurement period, allowing us to unveil how efficiently attackers deal with churn. Honeypot-based studies have to date not been able to do so.

Forged DNS Queries and Names. The query name and type in DNS queries affect the amplification factor. Historically, the most common queries included unpremeditated as well as crafted domain names, which were set up and used for amplification attacks immediately after registration [76]. `ANY` is an evident query type, yet querying for specific records can equally lead to large responses [41], [76]. DNSSEC is a DNS extension that enables verification of DNS content but at the same time significantly increases the potential for amplification due to larger response sizes [6], [131]. Consequently, benign `.gov` names, which are subject to a DNSSEC mandate [145], started being misused in amplification attacks [146].

We shed light on how attackers select names and study effective amplification in attack traffic at the Internet core. We also analyze large-scale DNS measurement data to estimate the amplification potential of other names, allowing us to reveal that while attackers are prudent in selecting names, other choices would lead to higher amplification.

Origins of DNS-Based Attacks. As reflection and amplification attacks involve IP spoofing, attack attribution is challenging [184], [45], [85], [108]. In the case of NTP and its moderate amplifier churn, considering the set of abused NTP servers has shown utility towards attributing attacks to a DDoS-for-hire service [71]. However, other research shows that overlap in underlying infrastructure can exist, in addition to other obstacles to fingerprinting [137]. Not all honeypot sensors are necessarily used by attackers at the same time and attackers can choose to abuse a subset of available reflectors in subsequent attacks. Nevertheless, clustering methods such as KNN allowed researchers to fingerprint a few major attacking entities and attribute

attacks to them [68], [79]. Another commonly used feature is the IP Time-To-Live (TTL) field, which was used to narrow down attack origins [9], [78].

Despite challenges in fingerprinting attackers, we successfully use network and application layer data to fingerprint a major attack entity, responsible for over half of the attacks detected at the IXP. We follow this entity for over 9 months.

3.3 Complementary Data Sources

We involve diverse and largely independent data from four data sources, bolstered by orthogonal methods. We next provide an overview of our main data to further comprehend the DNS amplification ecosystem, Our starting point is data from an Internet eXchange Point (IXP) for a three-month measurement period (2019-06-01 – 2019-08-31).

3.3.1 Traces from a Large, Regional IXP

IXPs are a key component of the Internet to interconnect Autonomous Systems (AS) without introducing high costs. Observing traffic at a popular IXP provides a similar vantage point to that of large transit providers [3]. We use traffic captures from a large, regional IXP in Europe. Our IXP connects over a hundred member networks and observes traffic peaks of 600 Gbps. We now detail how we identify and sanitize DNS data in IXP traffic, before using the data for attack detection.

Identifying DNS Traffic at IXPs. Our traces involve `1:16k` packet sampling and packets are truncated after 128 bytes, which can be challenging with respect to analysis of higher-layer (*e.g.,* application-level) protocols. On the upside of things, DNS usually operates with single UDP packets, hence packet sampling has no adverse effect as we do not need to observe complete flows. Moreover, the first 128 bytes of packets are sufficient to analyze DNS query packets. On the downside, in terms of analyzing DNS answers, response data is usually only partially visible (about 2 resource records per packet on average), since each DNS response contains request as well as answer data. Even though large UDP packets might be truncated and we cannot see the full answer data, we are still able to infer response packet sizes from the UDP length field, which precedes the DNS header.

Please note that we focus on DNS over UDP because TCP-based amplification attacks do not exploit features of DNS but only inefficient implementations of transport-layer sockets [82]. Also, even though stubs and forwarders use more recent DNS variants (DNS over TCP/TLS/HTTP) to contact resolvers, a recursive resolver usually still uses UDP to reach authoritative nameservers. TCP attacks use unencrypted traffic [2]. During our measurement period, only 1.25% of unencrypted DNS packets are based on TCP as a transport layer.

We use Tshark's DNS packet filter and dissector for protocol identification and empirically verify that truncated DNS packets are identified correctly. In the case of a UDP packet that

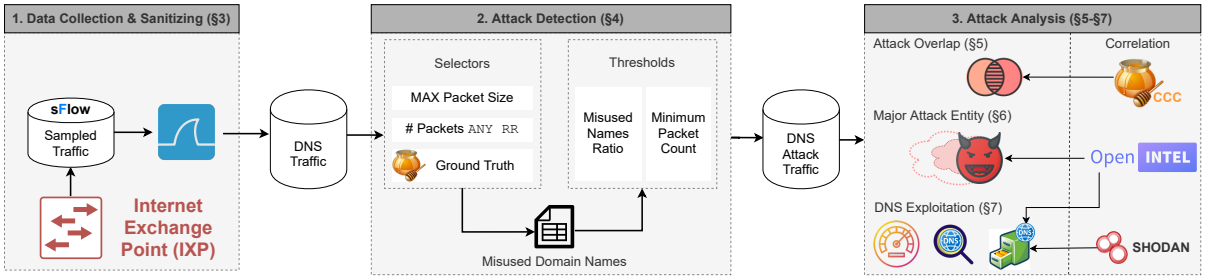


Figure 3.2: Overview of our data sources and attack inference steps.

leads to IP fragmentation, only the first fragment is identified as it contains the DNS header. This effectively avoids double counting of fragmented DNS answers. Overall, we find 33 million sampled DNS packets from June to September 2019, which correspond to a total of 528 billion DNS packets.

Sanitizing and Annotating DNS traffic. We only consider packets that include: (*i*) IP and UDP headers; and (*ii*) well-formed values for IP addresses, UDP length, DNS query types and names, *i.e.,* values allowed and standardized by the respective RFCs. In the process, we disregard 3% of previously identified DNS traffic. In the resulting data set, we observe slightly more requests than responses per day (60% are requests). Daily aggregate packet counts follow a weekly pattern with small changes during the weekends. The most and second most common DNS query types are `A` (57%) and `AAAA` (13%) records. Using public routing data [264] and IXP member information we map the origin AS for 99% of packets, and the peering hop AS for 96%. For each query and answer packet, we also note the client and server IP addresses.

3.3.2 Additional Data Sources

Honeypot Data. We use data from the *Cambridge Cybercrime Center (CCC)* honeypots [146], which are distributed and capture reflection attacks at the Internet edge. This honeypot infrastructure has various features: (*i*) it provides topological diversity by using 80 active sensors that are distributed across 62 IP prefixes and 15 ASes; and (*ii*) it emulates open DNS resolvers, which are responsive to reflection attempts, while not harming the Internet. We learn 31k DNS reflection attacks from the CCC data during the same 3 months.

It is worth noting that we carefully verified that the CCC platform is able to make similar observations compared to related honeypot platforms (for details see Subsection 3.9.1).

Large-scale, Active DNS Measurements. To investigate to what extent attackers might achieve amplification, we involve a longitudinal data source of daily DNS measurements that accounts for names that are not necessarily misused in amplification attacks (yet). We use data provided by the OpenINTEL project, which actively measures about 65% of the global DNS namespace, using well over 1200 zonefiles as a starting point [130]. OpenINTEL queries for a

Table 3.1: Our various data sources backed by complementing methods to analyze DNS amplification attacks.

Data Source	Type	Viewpoint
IXP	Traffic	Transit, Internet core
CCC Honeypot	Traffic	Amplifier, edge network
OpenINTEL	Scans	DNS TLD zone walking
Shodan	Scans	Complete IP address space

set of common resource record types, which allows us to map amplifier IP addresses to DNS infrastructure and to estimate response sizes (*i.e.,* amplification factors).

Internet-wide Scans. To verify whether an end host provided DNS services in the past, we use data from the Shodan search engine [273]. These data include daily scans of the complete IPv4 address space to discover Internet services per IP address.

We summarize our data sources in Table 3.1. Data sources of the category *scans* are based on active measurement methodologies, whereas *traffic* is brought about by passive observations. The complementary viewpoints allow us an in-depth understanding of effects observed for the DNS amplification ecosystem, as we show in the following sections.

3.4 Inferring DNS Amplification Attacks at an IXP

We first introduce our methods to infer misused DNS names and DNS amplification attacks in IXP traffic traces. We then briefly report about using these methods for live monitoring. Figure 3.2 shows an overview of our processing steps.

3.4.1 Identifying Misused Names

In DNS reflection attacks, queries for the right combinations of domain names and resource record types can trigger large responses and hence lead to sizable amplification. For this reason, attackers are likely to use effective names recurringly. Based on this assumption we find a list of suspicious DNS query names.

We develop the list of names using three so-called selectors. Two of our selectors consider features in the IXP data. The third selector involves the CCC honeypot data. The CCC data accounts for a substantial number of reflection attacks for which we may observe attack traffic at the IXP.

Selector 1: Max Packet Size. Our first selector considers the maximum (response) packet size of each and every query name observed at the IXP. Note that the response size per name may vary over time and also depends on the query type. We rank query names such that the first selector can pick, *e.g.,* the top-ten names in terms of max packet size. Large DNS

Table 3.2: Distribution of attacks and attack traffic for misused names. .gov names that dominate amplified DNS traffic.

TLD	.gov	.za	.cc	.pl	.cz	.com	.org	.se	.eu	.be	root(.)	.br	.ru
# Names	17	1	1	1	1	2	2	1	1	1	1	1	2
% Packets	74.92	1.32	3.92	1.1	1.17	1.31	0.99	0.54	0.38	6.23	6.73	1.38	0.005
# Attacks	22758	3969	3863	3732	3712	3388	3316	2663	2385	1551	1120	184	2
Max. Size [B]	8069	5155	4408	5954	5881	10270	6090	5535	4096	8199	4098	3893	–

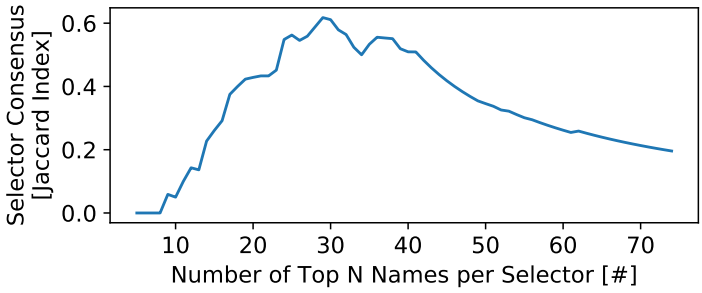


Figure 3.3: Our selectors detect the same names with a different ordering up to a set size of 29 names. These names are most likely to be misused in attacks.

responses may lead to IP fragmentation. Even in the presence of fragmentation, the UDP header, however, allows us to determine the size of the DNS response (see Subsection 3.3.1). The largest response of more than 10k bytes was triggered by an `RRSIG` query, the remaining top-ten largest responses are triggered by `ANY` queries.

Selector 2: Number of `ANY` Packets. The `ANY` query type is a convenient way to bring about DNS amplification, provided that `ANY` queries are not restricted by the authoritative nameserver of the chosen query name. This is why our second selector considers names that most appear in `ANY` query packets. The ten top-ranked names according to Selector 2 are used almost exclusively for `ANY` queries. Considering `A`, `AAAA`, and `ANY` packets, the share of type `ANY` packets is higher than >99.99% for all names but for the root (`.`) name (97%).

Selector 3: Query Names Used Against Reflected DDoS Victims. For our third selector we start by extracting all DNS attack victim IP addresses and timestamps from the CCC sensor data. Next, we search for the IXP DNS traffic associated with the attacks. Selector 3 then chooses the most common names used in the traffic in question. We find DNS attack traffic for 16% of all CCC DNS attack events (≈ 4.4k victim IP addresses). We identify two reasons for invisible CCC attack traffic at the IXP: (*i*) The traffic is not routed via the IXP, and (*ii*) the traffic is routed via the IXP but the packets are not sampled (given our `1:16k` sampling rate). The ground truth attack traffic consists almost exlusively of `ANY` packets (99%) and we observe only 482 unique names with this selector.

We consider the IXP DNS traffic associated with victim IP addresses at the time of an attack as *ground truth*. We later use this ground truth to validate detection thresholds. It is worth noting that the CCC data also provides query names, however, we decide to not use them in favour of selecting names that are actually visible from the perspective of the IXP in ground truth attacks.

Number of Names per Selector. The number of names chosen per selector is configurable. To determine the highest similarity, we calculate the Jaccard index for the three sets of names using increasing set sizes. We observe a high consensus for 29 names per selector (see Figure 3.3),

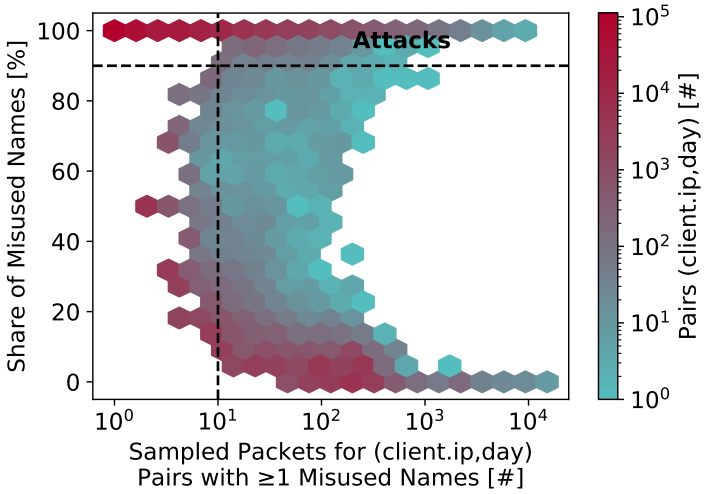


Figure 3.4: Share of misused names compared to overall traffic. Many clients exchange DNS traffic with only misused names, which aids attack detection.

which shows that the first 29 names chosen by each selector are almost the same, but with a different ordering. Note that selecting the point of highest consensus is a conservative measure for two reasons. First, this reduces the number of distinct names but chooses names for which up to three selectors agree. And second, the selector results follow a long-tail distribution with the knee points before the consensus point, which means that selecting more names would lead to adding *insignificant* names. All things considered, we set the size to detect misuse at 29 names per selector.

Finally, we merge the three selector sets of names to create our final list of names. The union combined with the high consensus point allows for a conservative name selection while still keeping significant names detectable only by a single selector. Our final list contains 34 names. For 32 of these names (94%), we detect attack traffic (see Subsection 3.4.2), which demonstrates the effectiveness of the selectors in identifying misused names. Table 3.2 shows properties of the considered names, most of which are part of the `.gov` zone. 21 names are mutually detected by all 3 selectors. The intersection of Selector 1 and 3 contains three names, and the intersection of Selector 2 and 3 contains five names. We find two exclusive names with Selector 1. Overall, the IXP and the first two selectors are sufficient to create our list. The CCC data does not add any names compared to the unions and intersections of all three sets. Using the honeypot-based selector is a good verification of the first two selectors, though, since it is based on ground truth attack traffic.

3.4.2 Attack Detection with Misused Names

Using the previously introduced list of misused names, we now further analyze DNS packets that contain queries or answer data for these names. This will allow us (*i*) to define two thresholds for the detection of attacks at the IXP, and (*ii*) to group related packets into attacks.

Threshold 1: Traffic Share of Misused Names. We calculate the daily "traffic share" of suspicious packets for client IP addresses (*i.e.,* supposed DNS query originators). A high share can indicate attack activity. Please note that the client IP address denotes the source IP address of DNS requests and the destination IP address of DNS responses. The share of suspicious packets is calculated for each unique *(client.ip,day)* pair for which at least 1 suspicious packet was observed. This excludes unrelated DNS activity, *i.e.,* clients which exchange traffic for only benign names on a given day. Then, we visualize the share of misused names for each *(client.ip,day)* in Figure 3.4. This reveals that with an increasing packet count a bimodal distribution becomes more pronounced, *i.e.,* even though clients exchange large numbers of DNS traffic, the related traffic consists of only misused names or almost none. The low shares occur due to the fact that one of the misused names is the root (`.`) name, which is also a very common name for legitimate DNS traffic. This distinctive distribution allows for the introduction of thresholds to detect attacks. With our first threshold, we define that a client is under attack if the share of misused names exceeds 90% on a given day. This finds extreme cases of suspicious traffic shares but still allows for a small error margin, *i.e.,* we might observe other names due to legitimate DNS traffic of the client. Note that for clients with a low traffic volume (*e.g.,* 1 sampled packet), this single threshold is not enough since it most-likely leads to many false positives. We therefore set a minimum packet count threshold at the beginning of the bi-modal distribution (details see Threshold 2). With respect to the minimum packet count threshold, the traffic share threshold of 90% accounts for the smallest possible error, *i.e.,* exactly 1 sampled legitimate packet.

We argue that the high traffic share of misused names is a strong indicator of attack traffic. To illustrate this argument consider ten sampled packets, our sampling rate of `1:16k`, and a misused name share of 90%. This would correspond to 144k packets with only misused names. No client should reasonably exchange so much DNS traffic for legitimate reasons, especially in the presence of DNS caches.

Threshold 2: Minimum Packet Threshold. We now explore the effects of a minimum packet threshold at the IXP, in particular we analyze the trade-off between the detection of all attacks (*visibility*) and reducing false positives. To this end, we use our ground truth attack events that we found at the IXP with the help of CCC sensor data. We count the number of packets for these attack events and plot the fraction of visible events w.r.t. minimal packet count, see Figure 3.5. To provide a reference point, we also include the visibility of DNS traffic for all *(client.ip,day)* pairs. Overall, this plot demonstrates which share of DNS traffic remains visible at the IXP if a minimum packet threshold is applied. We find that 22% of visible

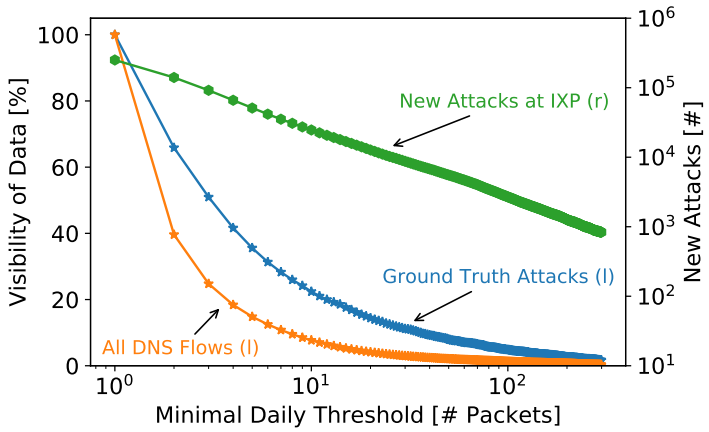


Figure 3.5: Visibility of all DNS flows and ground truth attack flows depending on the number of packets considered. Number of detected DNS attacks at the IXP based on the thresholds are shown on the right y-axis.

ground truth attacks exhibit at least 10 sampled packets, *i.e.,* they remain detectable while applying a minimum packet threshold of 10 packets. Note that for all *(client.ip,day)* pairs, the visibility for 10 packets is, as expected, much lower (8%), since regular DNS flows only consist of significantly fewer packets. Looking at the total number of additionally detected attacks at the IXP (secondary y-axis), the threshold of 10 packets at minimum strongly limits the number of detected attacks. We argue that this significantly reduces false positives (or at least vague cases) but still allows us to find over 24k new, significant attack events at the IXP. Again, these attacks were not observed by the honeypots, hence provide an opportunity for new insights.

Validation. A sound attack detection mechanism should be able to detect attacks for which we know to be visible in the sampled IXP traces. Given this notion, we now investigate the detection rate for visible CCC attacks, based on the defined thresholds and a varying number of names for our selectors. This allows us to verify whether precision of our attack detection would increase by adding more misused names. Figure 3.6 shows that the detection rate converges at 99% with 29 names per selector for our threshold configuration. This clearly illustrates that we do not need to fine tune our detection method further. Also, this result is coherent with the selector consensus, which, again, suggests that adding more misused names does not have a beneficial effect.

First Glimpse into Detected Attacks. At the IXP, we found 25.7k attacks to 19k unique client IP addresses, which includes 24.6k new attacks (as previously mentioned). The detected attacks are dominated by traffic created by misusing `.gov` names, see Table 3.2. The attack durations match the observations of security reports [218], [250] with many short-lived attacks (25% shorter than 7 minutes 50% shorter than 33 minutes). One third (36%) of the total

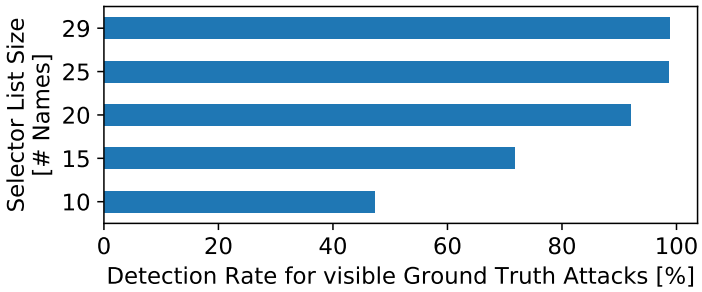


Figure 3.6: Attack detection rate based on selector list sizes and 2 thresholds. We reach 99% for visible ground truth attacks and see a convergence around 29 names.

attack traffic is sent towards victims in ISP networks, which is the largest victim group after content networks (24%).

We see no signs of NXNS attacks [2]. Those rely on responses including NS referrals with many NS names (> 30) but no glue records. In our data, 70% of the responses include at most 1 NS entry and 90% at most 10 NS entries. Recently detected attack vectors (*i.e.,* `SRV`, `URI`), which also offer a 10× amplification factor [101], are also not used, yet.

3.4.3 Live Monitoring

We deployed our method at the IXP to verify online detection capabilities in realistic settings. Our prototype consists of two building blocks: (*i*) A module that identifies potentially misused names in near real-time. (*ii*) A module that continuously analyzes changes compared to the previous day. Without advanced performance optimizations, we are able to identify misused names within a maximum delay of 5 minutes, on commodity hardware.

We utilize our deployment to assess victim and name fluctuations. Overall, we see quite stable numbers of unique victims and also very stable lists of misused names. On average, we observe 631 unique victim /24-prefixes (492 /16-prefixes and 121 /8-prefixes) per day. The name lists have a mean Jaccard index of 0.96 in comparison to the respective previous day. This suggests that daily updates for misused names are not necessary; we keep them to identify changes quickly.

3.5 Comparing IXP and Honeypot Data

We now present basic properties from attacks inferred at the IXP and compare the observations to honeypots. We find that the IXP and the honeypot sensors observe a vastly disjoint set of attacks. Both vantage points share only 1.1k attack events, which corresponds to 4.2% of all events at the IXP and to 3.5% of 31k attack events at the honeypots. This is a surprising result, given that prior work [76], [146] assumed that a distributed honeypot, such as ours, can

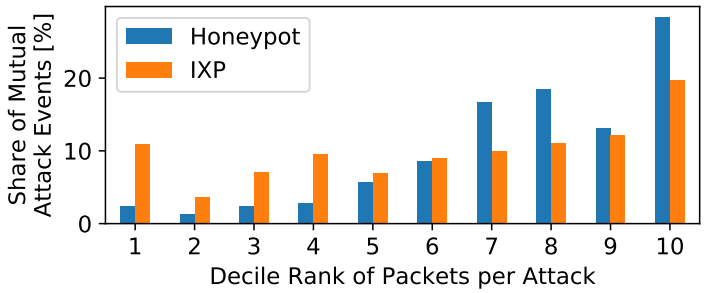


Figure 3.7: Attacks detected by the IXP and honeypots (1098) differ in relative attack intensity score: Mutual attacks are rather strong honeypot attacks, but medium-sized IXP attacks.

capture a large percentage of global reflection attacks. We consider an IXP vantage point to be an opportunity to observe DNS amplification attacks which have been so far invisible to the research community, and potentially provide new insights, *e.g.,* for attacks that deliberately exclude honeypot platforms.

While the overlap is small, we now check, for comparative purposes, whether the honeypots and the IXP agree on the observations for mutual attacks. To this end, we calculate a relative attack intensity score for all attacks that have been identified by each type of vantage point. We do this by sorting all attacks by the total packet count and calculating the deciles. Then, we rank each attack with the decile score of 1 to 10. We plot the relative distribution of intensities for mutual attacks in Figure 3.7. Overall, honeypots are rather sensitive vantage points: the mutual attacks are mostly strong honeypot attacks (with a mean intensity of 7.7) and medium-sized IXP attacks (with a mean intensity of 6.3). We argue that this is due to packet sampling and our thresholds, which make smaller attacks invisible at an IXP. Hence, honeypots are good vantage points to detect small-sized attacks, *if* they are abused by the attacker as reflector. IXPs, on the other hand, show that a substantial number of large attacks occur, which were not observed by the honeypots. They are likely to see even more small attacks but this would require significantly smaller sampling, which is uncommon in practice. We will leave this for future work.

3.6 Tracing a Major Attack Entity

In this section, we reveal a major attacking entity, which is responsible for 59% of all attacks at the IXP. To this end, we identify recurring patterns based on the selection of domain names, the creation of DNS requests, and the selection of amplifiers. Our method does not depend on our specific vantage point but can be generalized to other inter-domain data sets.

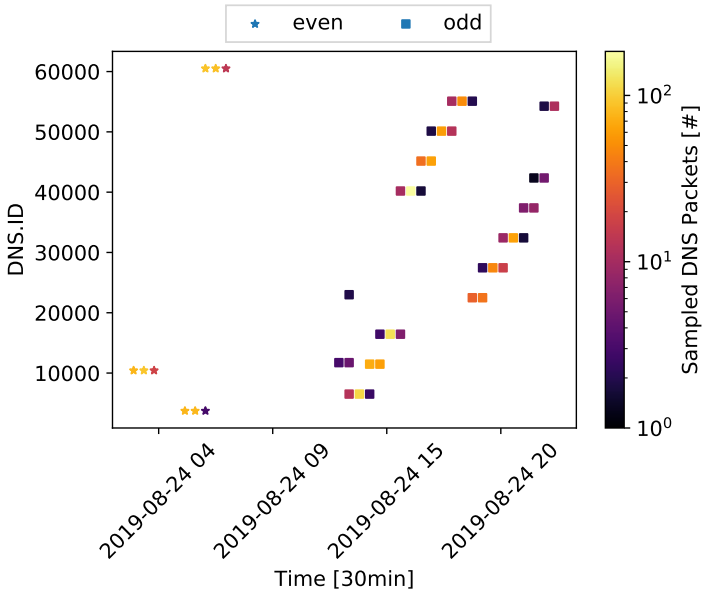


Figure 3.8: Number of attack packets and value of DNS IDs for a single victim. For each attack event, DNS IDs tend to be odd or even per attack phase which also allows for fingerprinting the attack entity.

We link multiple independent events to an attack entity. We explicitly use the abstract term *entity* as we do not refer to a specific botnet, booter website *etc.* but to the essence that maintains an infrastructure to select names and amplifiers to launch attacks.

3.6.1 Fingerprinting Using Domain Names

We conduct a time series analysis of the misuse of names. Our results reveal a clear transition between names for selected `.gov` names (see Subfigure 3.9(a)), which contribute 59% of the overall attack traffic. Names appear to be chosen in lexicographical order, except for few weeks in which two names were used concurrently.

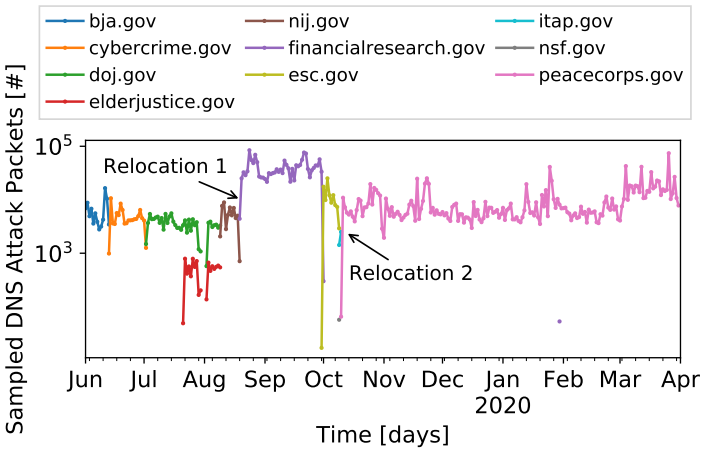
The transition pattern between names strongly suggests that a specific entity is involved in the attacks. Independent misuse of the same domain would not lead to clear, abrupt transitions. Interestingly, we observe an increase in attack traffic at the IXP following changes in misused name. This hints at a driving factor behind the transitions in lexicographical order. To further investigate this observation, we analyze the expected response sizes offered by names, as this is crucial for amplification attacks. Subfigure 3.9(b) depicts the `ANY` response sizes of each name inferred from the OpenINTEL data set, which provides us with historical DNS data. The dashed line indicates the recommended maximum payload size (4096 bytes) of EDNS [181], the extension mechanism in DNS to carry, *e.g.,* DNSSEC data. **We observe that the expected response sizes change while names are actively misused in attacks, and also that**

transitions to other names follow drops in sizes. Further analysis of the OpenINTEL data set reveals that the plateaus in response sizes—which last two weeks—relate to DNSSEC key rollovers. When a new zone signing key (ZSK) is introduced, an increase in response size can be expected, as multiple `DNSKEY` records are present at the same time. ZSK rollovers can be completely automated in software, which explains the regular patterns. RFC 6781 [215] recommends two rollover schemes, pre-publish and double-signature. Pre-publish introduces only the new key in stand-by mode, *i.e.,* the key is not yet used to sign RRsets. This allows resolvers to learn about the new key before it is actively used. This scheme, however, is prone to race-conditions and misconfigurations [25] which impair the validation process. To overcome the challenges of pre-publishing, the double-signature scheme has been introduced. Double-signature allows two active ZSKs and generates two (redundant) `RRSIG` records signatures. The old ZSK can then be retired at any given time. On the downside, this scheme doubles the number of signatures in a zone. Although both rollover schemes are proposed in RFC 6781 [215], pre-publish has been established as a de-facto standard. It was used `4x` more often than double-signature in 2016 [25], [254], and `8x` more often in 2020 [254]. Also, it is recommended by various DNS software vendors [254].

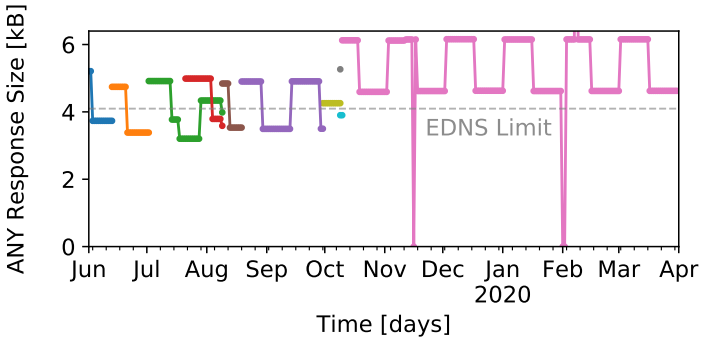
We only observe double-signature schemes for the misused `.gov` names. This leads us to conclude that **operators of these `.gov` names, many of which are US federal government domain names, do not only not adhere to best practices, which exacerbates amplification, but also that these decisions introduce misuse by others.** A recent (Q2 2021) sample of DNSSEC records for `nsf.gov` and `doj.gov` shows that the rollover practices did not change.

Even though we observe transitions after a (variable) number of days when the expected size is below the recommended EDNS limit (see the valleys in Figure 3.9), we cannot reasonably infer the decision making process behind. Either the attack entity completely understands DNSSEC mechanics or simply observes < 4096 byte responses and then (manually) transitions to the next name.

By analyzing the packet sizes in the sampled IXP data, we can confirm that the attack entity achieves effective amplification factors. In contrast to Subfigure 3.9(b), which exhibits the potential maximum `ANY` response sizes, Figure 3.10 shows the relative frequency of the actual response sizes observed at the IXP and extracted from the UDP headers, grouped by each name. Please note that we consider all DNS query types for the misused names. In the attack traffic, however, we only observe the type `ANY` for these names. Most names exhibit a bi- or tri-modal distribution. The observed clusters of response sizes near the theoretical limit highlight that the attack entity succeeds in finding names (and related authoritative nameservers) as well as amplifiers that still allow `ANY` requests. Closer investigation reveals that smaller response sizes appear rather at the end of a name's life cycle. This bolsters our result that the entity observes the current effective amplification factor and updates misused names upon a decline.



(a) Attack traffic volume of misused names based on IXP data.



(b) Estimated `ANY` response sizes of currently misused names based on OpenIntel data set. Plateaus indicate DNSSEC key rollovers. Cutoff of a single anomaly (value of 12.5kB) for better readability.

Figure 3.9: Time series of synchronized names misused by major attack entity.

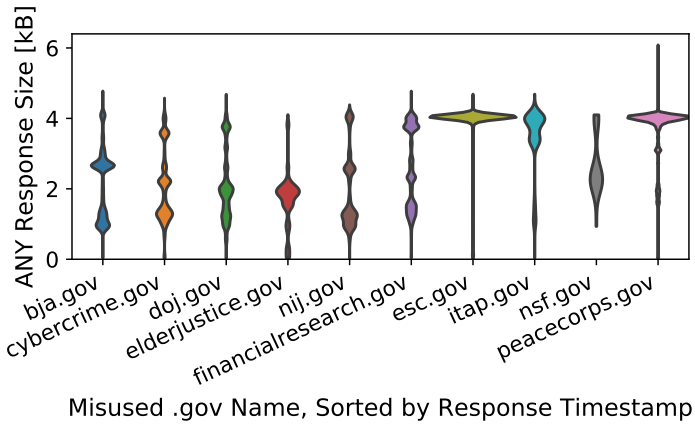


Figure 3.10: Violin plot of the observed DNS response sizes at the IXP for the major attack entity.

Additional Fingerprinting Features. To verify that we can link multiple events to a single entity, we seek other features that may indicate uncommon consistency over time. To this end, we perform an entropy analysis of packet header fields that usually should show high randomness. If they do not, we suppose the deployment of the same attack tool. Pre-built headers and usage of raw sockets may lead to such consistent behavior, for example.

For each attack event, we check whether the number of distinct values of a specific feature grows linearly with the total packet count. We investigate header fields such as `IP.ID`, `UDP.SRCPORT`, and `DNS.ID`. As network and transport layer features change after reflection, we consider only DNS queries (*i.e.,* packets that are sent before amplification) here.

Unfortunately, all features in the network and transport layer headers exhibit a linear growth and hence a high entropy. Fortunately, we detect a low randomness for the DNS transaction ID, a feature in the application header. The number of IDs in use is usually 1-2 orders of magnitude smaller than the total packet count, see Figure 3.11. The low entropy gives good reasons to manually inspect the DNS IDs. We found that 91% of attack events have only a (seemingly random) selection of odd or even IDs. With respect to the minimal number of sampled packets containing misused names (9), the probability for this observation with random DNS IDs is $2 \cdot (1/2)^9 = 0.4\%$. Also, we rule out measurement artifacts such as a synchronization between traffic and sampling, since sampling selects `1 out of 16k` and not `every 16kth` packet. Hence, we argue that we found an arithmetic structure and not only a random phenomenon. For the remaining 9% of attack events we observe two phases with odd and even IDs, respectively, and a distinct shift. Indeed, the overall selection of IDs for the attack events with synchronized names follows a two-day rhythm, alternating between odd and even DNS transaction IDs every 48 hours, independent of other features. We showcase attack phases and related transaction

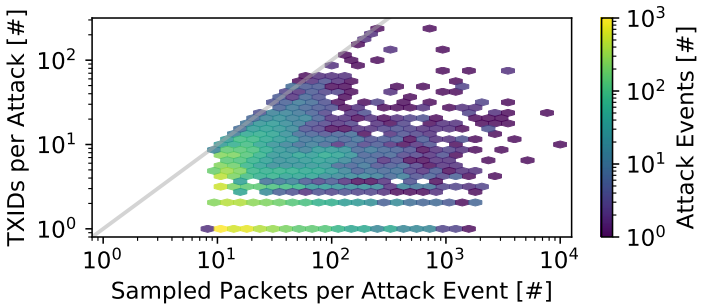


Figure 3.11: Entropy check: # of unique DNS transaction IDs and packets for all packets from attack entity. A limited number of DNS IDs indicates pre-built queries.

IDs for a single victim in Figure 3.8. The selection of IDs is probably seeded with timestamps and not linked to the properties of the victim.

In summary, we are able to fingerprint a major attack entity based on two properties: (*i*) its selection of names and (*ii*) the implementation details of the attack tool (selection of DNS IDs). Both features are part of the application header, which means that we can link attack traffic to this entity even after the reflection occurred. Similar to our observations in Section 3.5, the fingerprint of this attack entity is only visible for $\leq 0.6\%$ of the attacks detected by the honeypot, *i.e.,* the attack entity is only clearly visible at the IXP.

3.6.2 Attacked Victims, Misused Amplifiers

We now describe the executed attack events as well as reconfigurations and relocations controlled by the attacker. The attacks we associate with this entity are distributed. We observe almost as many victim destination IP addresses as covering victim prefixes per day, see Figure 3.12. In this plot, we highlight the transitions between misused names with vertical lines. The number of victims remains stable until the transition to the last name occurs. Then, the number of victims increases by almost an order of magnitude. The increase also correlates with the total number of packets (compare Subfigure 3.9(a)).

We check whether the attacker reconfigures only the list of misused names or also the list of misused amplifiers. To this end, we count the daily number of new and known amplifiers, *i.e.,* amplifiers that have been already misused at least once, see Figure 3.13. This plot introduces two findings. First, although the number of total attacks increases, the number of misused amplifiers remains stable. This suggests that the entity misuses only a specific set of amplifiers per day. Random subsets, however, are selected per attack event, which we will show in detail in Section 3.7. Second, periods with significantly more new amplifiers usually follow name transitions, indicating that names as well as the amplifier list were updated at the same

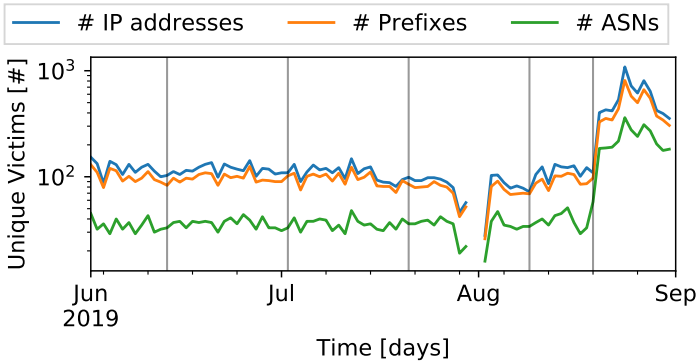


Figure 3.12: Number of unique victims identified by IP address, prefix, and AS numbers. Vertical lines highlight transitions of misused names.

time. Nevertheless, new amplifiers appear almost daily, revealing a more continuous update behaviour, which is necessary due to IP churn of DNS hosts.

In order to understand the increased number of attacks, we continue investigating other features. We find that starting with the peak mid of August, the DNS request-response ratio for this entity shifts dramatically. Before, we observe almost purely amplified DNS traffic, *i.e.,* DNS responses. The absolute numbers of responses remain stable, however, we see a stark increase in requests. Now, $\sim$ 85% of attack traffic consists of requests. Moreover, 99.8% of the requests originate from the same ingress AS and exhibit the same IP TTL of 250. Seeing such a concentration from a single ingress AS indicates a centralized attack infrastructure, because botnets are usually distributed across multiple networks. Such infrastructures are usually the hidden back-end of booter websites. Unfortunately, the customer cone of this ingress AS contains more than 16k ASes, so we are not able to fully trace back the infrastructure. We do not find topological changes at the IXP that would justify the shift in attack traffic properties, *i.e.,* we do not find any new members and the ingress network has been already a member during the whole measurement period. Also, it is unlikely that this shift is caused by unrelated routing updates as the paths to all amplifiers would have to change simultaneously for such a homogeneous effect. Instead we argue that since the shift occurred concurrently with a name transition, this effect is triggered by the attack entity itself, namely due to a relocation into the ingress cone of the IXP member. We define relocation as the topological transition of a centralized attack infrastructure into another network. We later observe a second relocation in mid October.

To sum up, by revisiting the main measurement period with a fingerprint in hand, we were able to identify reconfigurations of names and amplifiers, and a relocation of the attacking infrastructure. We were able to do this on the basis of network and application layer information visible at an IXP.

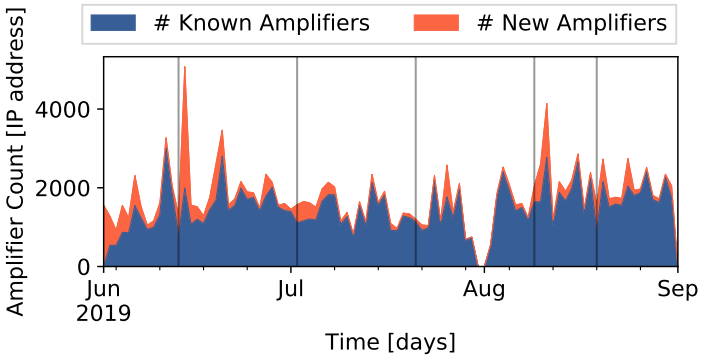
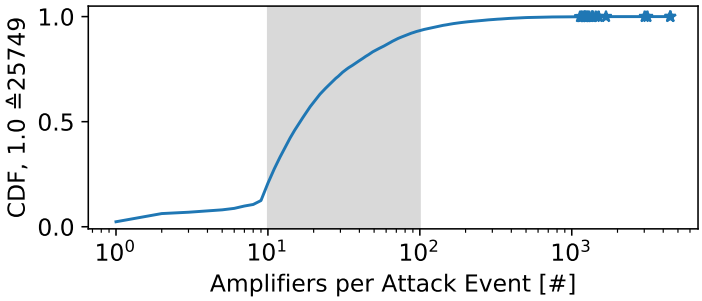
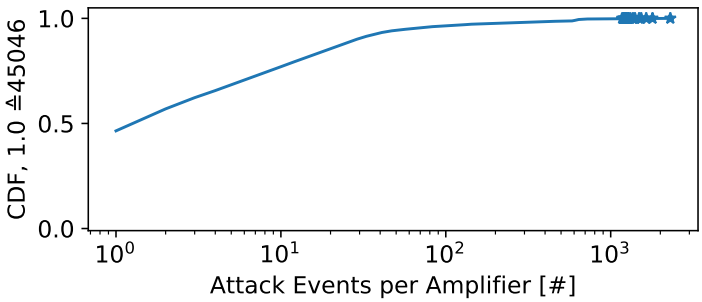


Figure 3.13: Known and new amplifiers used by the major attack entity. Bursts of new amplifiers correlate loosely with name transitions (vertical lines).



(a) Number of amplifiers per attack: Most victims are attacked by 10 to 100 amplifiers.



(b) Number of attacks per amplifier: 50% of the amplifiers participate in more than one attack, even 23% of amplifiers in more than ten attacks.

Figure 3.14: Distributions of amplifier involvement in attacks. Last 20 data points are highlighted.

3.7 Unveiling DNS Attack Practice

In our last analysis, we use our inter-domain IXP perspective to disclose the abused infrastructure. Also, we analyze how (all) attackers perform in terms of amplification efficiency.

3.7.1 Amplification Ecosystem

We start by investigating whether attackers continue to abuse the same reflectors across attacks. Repeatedly using a stable infrastructure may make attackers (*i*) fingerprintable and (*ii*) susceptible to frequent re-addressing of edge resolvers.

How many amplifiers are used in attacks? And in how many attacks do particular amplifiers appear? The OpenINTEL data accounts for a large number of authoritative nameservers active during our main measurement period: approximately 4.2 million `NS` names that together map to well over a million IP addresses. We use these data to associate amplifier IP addresses observed at the IXP with authoritative nameservers, where applicable.

We find that only 908 authoritative nameservers are abused in attacks—about 2% of all amplifiers observed at the IXP. By exclusion, we conclude that the vast majority of abused DNS amplifiers are open resolvers or forwarders. We discuss further a classification of forwarders and resolvers in Subsection 3.9.3. This does not come as a surprise, because authoritative servers should not recursively resolve DNS queries, which makes them less attractive reflectors. Root-query-based attacks, however, utilize 4× more authoritative nameservers, which can be linked to attacks misusing misconfigured root hint-files [210], [165, Chapter 4]. We observe that 80% of attack events use between 10 and 100 amplifiers (numbers not extrapolated by sampling rate), *cf.* Subfigure 3.14(a). Also, Subfigure 3.14(b) shows 23% of amplifiers that participate in more than ten attacks. Such recurrent use of amplifiers may allow for fingerprinting attackers.

Do attack entities work with stable lists of amplifiers? After observing recurrent amplifiers, we now investigate whether attackers use relatively static amplifier sets. As the DNS is subject to high amplifier churn [81] from home gateways with 24h IP address lease times [80], we expect sets to exist for short time spans, only.

We approach our analysis by quantifying the (dis)similarity of two attacks from measuring the Jaccard distance over its respective sets of amplifiers. A group of similar attacks (*i.e.,* cluster) with a low Jaccard distance among each other indicates a fixed list. We use a bilateral clustering method by using two well-known algorithms: T-Distributed Stochastic Neighbor Embedding (T-SNE) [92] and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [40]. We compute both algorithms independently to exclude a biased result from a single clustering method.

T-SNE allows us to visualize high-dimensional data on a two-dimensional plane. Similar attacks are moved towards each other and dissimilar attacks are move apart. We observe very stable results for different perplexity parameters. The clustering results are visualized in Figure 3.15, each gray scatter point represents a single attack event. T-SNE indicates a strong

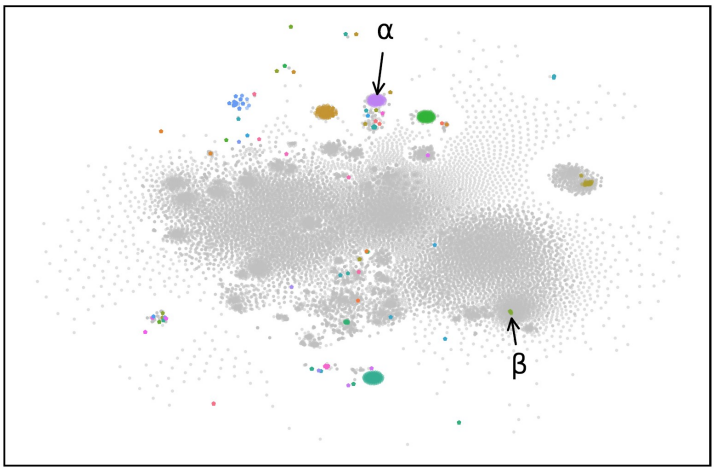


Figure 3.15: T-SNE visualization of attack events based on Jaccard distance over the amplifier sets. DBSCAN clusters marked with colors (gray being not classifiable). Both clustering algorithms agree on the dissimilarity of attack events.

dissimilarity between most events, with some noticeable clusters. DBSCAN groups nearby neighbors into clusters and marks non-classifiable outliers within low-density regions. We next combine both clustering results in the single Figure 3.15. DBSCAN cluster IDs are encoded with colors, the non-classifiable outliers in gray.

We see 67 clusters while ∼92% of attack events remain outliers. We inspect clusters of at least 5 attacks and 5 amplifiers to find stable sets. Here, the most static amplifier set (α) was used for 177 attacks during 40 days without any change. The largest set (β) uses ∼527 amplifiers per attack while always introducing a small, steady change. We can attribute in total only 2% of attack events to fixed sets. Attackers seem to steadily use a random combination of known and new amplifiers. This reinforces our previous findings that attackers leverage the amplification ecosystem and that source-based filtering is infeasible to mitigate DNS amplification attacks.

Do attack entities recruit new amplifiers? Since our results suggest that attackers steadily vary their amplifier sets, we question which amplifiers are used over time. To this end, we use the Shodans historic lookup, which allows to retrieve its complete scan history for a given IP address. Shodan omits transparent DNS forwarders. It lists currently around 2 million recursive DNS resolvers, which all can be abused for reflection. Next, we perform a historic IP address lookup for all 45k amplifiers observed at the IXP.

We find that 95% of these amplifiers are reported by Shodan to serve recursive DNS at some point in time. This finding grants two insights: (*i*) it confirms independent observations that although most amplifiers are known to the community, we fail to remove these amplifiers ultimately [112], [120]; (*ii*) attackers do not use private but mostly publicly indexed amplifiers.

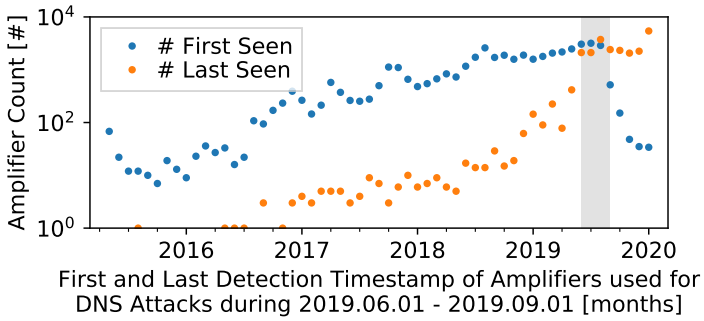


Figure 3.16: Number of Shodans first and last interaction with reflectors that were observed by us during the attacks at the IXP. Timerange of attacks highlighted in gray.

Scan results can differ even for Internet-wide measurements, *e.g.,* due to the origin of the scanner [154]. However, both Shodan and attackers observe a very similar set of DNS-amplifiers.

To examine the age of an amplifier that was abused during our measurement period, we determine its first and last successful detection by Shodan in Figure 3.16. A significant number of amplifiers was first seen during six months preceding the attack, *i.e.,* attackers mostly use amplifiers that are not older than six months. Also, many amplifiers are observed for the last time during or right after our main measurement period indicating that (*i*) operators change the inadvertently open state of their resolvers; or (*ii*) the amplifiers churned because of a dynamic IP address. Notably around 850 reflectors (2%) appeared in attacks before discovery by Shodan. This suggests that some attackers run their own scanning engines with a higher scan frequency or accuracy than Shodan.

At this point, our methodology allows to passively identify DNS amplifiers as they are abused, even before other measurement efforts succeed. Overall, we observe substantial DNS amplifier churn at the IXP but discern no downside for attackers. Note that we observe actual amplifier abuse at the IXP, not the churn in amplifier reachability (which scans can reveal). Although the total number of abused amplifiers remains stable between attacks, we see on average only 45% of abused amplifiers in subsequent $(\text{day}_i, \text{day}_{i+1})$ pairs. Comparing the first and last day of our three-month measurement period, only 20% of amplifiers still make an appearance. This observation suggests that attackers effectively detect—and purposefully rotate—new DNS amplifiers.

3.7.2 Potential Amplification Factors

Do attackers select names that maximize amplification? We investigate whether attackers inquire names for maximizing amplification, or whether there is an unused threat potential. Using the OpenINTEL data, we estimate the response sizes of `ANY` queries of 440 million domain names and plot the CDF, see Figure 3.17. Please note that we calculate the response sizes

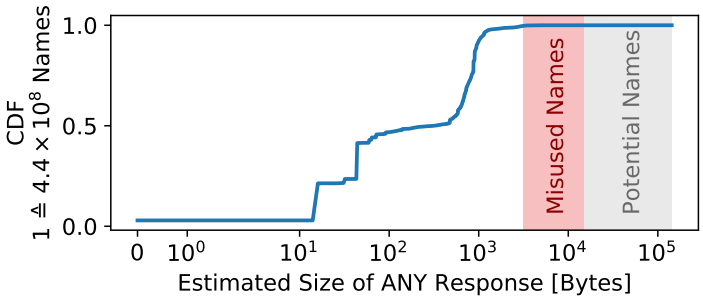


Figure 3.17: Estimated `ANY` response sizes for names measured by OpenINTEL. We highlight the range for currently misused names (red), and show the range of potential names (only 9048 distinct names) to increase the amplification factor (gray).

based on the cumulative resource record sizes stored in the DNS and ignore common software or protocol limits (4096 bytes for EDNS and 65,536 bytes for UDP).

The names previously observed in misuse exhibit a response size highlighted in the red area. Overall, only 9048 domains show a higher amplification factor than the highest ranked, misused name—about 0.002% of all names (gray area in Figure 3.17). This suggests that attackers attempt to cherry-pick names for high amplification factors without being optimal. Our estimated largest response size is 142,855 bytes, whereas the largest we actually observed is 14× smaller.

Can we expect larger attacks in the future? Frighteningly, we find that ∼92,000 names (0.02%) in the OpenINTEL data set may lead to a response size larger than 4096 bytes. Even though EDNS [181] recommends to not send larger replies, our measurements reveal that the DNS infrastructure frequently does so in practice (see Subsection 3.6.1).

Visible DNS attack events contribute a substantial amount of DNS attack traffic to the Internet core. Notably, the overall attack traffic at the IXP accounts for 5% of the total DNS packets and 40% of the total DNS traffic volume. This trend becomes even more apparent when only `ANY` traffic is considered: 68% of `ANY` packets and 78% `ANY` bytes are part of attacks. The situation will grow worse when attackers begin to use the names with a higher amplification factor.

3.8 Discussion

Is the observation of the major attack entity a bias of our vantage point? Despite being a central element of today's Internet, most IXPs still operate locally to interconnect networks. To verify that our observations are not a local phenomena based on our large, regional IXP, we assume that popular names are likely to be cached in the DNS. To quantify world-wide usage of names, we apply a modified cache snooping analysis. In a nutshell, we

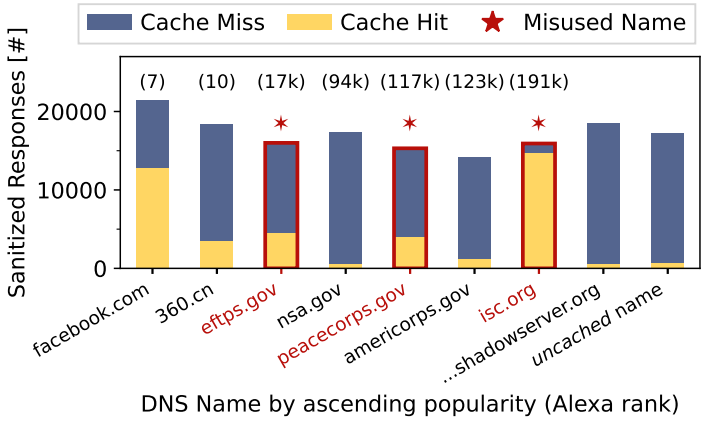


Figure 3.18: DNS cache hits for a set of arbitrary names and misused names. Names with a low popularity in the Alexa ranking but high cache hit rates indicate world-wide usage of the names for other reasons.

resolved misused names as well as a set of arbitrary names via all public resolvers and compare whether the names were cached or not (details see Subsection 3.9.3). We correlate the cache hits and misses with popularity of the names in the Alexa ranking. For reference purposes we created a name only for this study (*i.e.,* a name that was not cached before).

Figure 3.18 shows that misused names (highlighted in red and with a $\star$) have a similar cache hit ratio as very popular western and eastern names, even though our misused names exhibit much lower popularity based on the Alexa ranking. The results indicate that the misused names are resolved frequently but not because of common (Web) services. Hence, we argue that the IXP and our methodology give insights into behavior of global scale.

Our results are further substantiated by a recent study published by an anti-DDoS provider [219]. Labovitz [219] confirms that one of the misused names identified by us (`peacecorps.gov`) has been utilized by the booter *SynStresser* to perform attacks. Also, some of our attack events correlate with publicly documented attacks [265].

Would authoritative name servers provide a complete picture? No. 98% of open DNS amplifiers are forwarders and not recursive resolvers [112]. This means that the majority of amplifiers do not communicate with authoritative servers. Also, recursive resolvers will contact an authoritative server only when the name is not locally cached. Cached responses, however, are common because they make DNS scalable. TTLs may range between 1 hour up to days [106]. We observed the impact of caching in the bi- and tri-modal distributions of attack traffic at the IXP (see Section 3.6). Furthermore, our data corpus includes individual resolvers that serve up to 20k DNS amplifiers, which illustrates that caching is more likely and, thus,

requests less visible at the authoritative servers. Hence, neither the misuse of a name nor the complete attacking infrastructure might be visible to an authoritative server.

What can operators do to improve the situation? Operators can help by configuring their authoritative nameservers or recursive resolvers to (*i*) block `ANY` requests completely, (*ii*) respond to `ANY` requests only via TCP or with a minimal subset [164], (*iii*) deploy rate limiting. Similar recommendations have been proposed for years [131], unfortunately DNS amplifiers still exist. Our observations suggest that those countermeasures are still helpful because an attack is based on a relatively stable set of queries. In case advanced query patterns [2] are issued in the future, which our method would detect, the deployment of filters that focus on names or observations across multiple resolvers are options. As we found that some few resolvers serve a significant amount of amplifiers (*i.e.,* forwarders), educating those first will have larger impact.

To the best of our knowledge, this is the first work that shows the adverse impact of DNSSEC key rollovers in the context of amplification attacks (see Subsection 3.6.1). Double-signature rollovers temporarily create a second, superfluous set of signatures, which makes these names more attractive to attackers. Operators should pay attention to misuse during rollovers for their zones. Overall, we recommend pre-publish rollovers which currently are best practice [254].

Advantages of IXPs? IXPs are considered to be central vantage points [24]. We introduced methods to leverage IXPs to shed new light on DNS amplification attacks. We found that honeypot platforms see less compared to what was assumed before, extending recently observed trends [70]. To achieve a similar coverage compared to large, regional IXPs, honeypots require broader distribution. What concerns us most is that honeypots are easy to detect [107], [260], [155], either because they deploy (for good reasons) rate limiting [76], [146] or they expose other features such as delays that enable fingerprinting. Prior work clearly indicated that malware adapts and hides [57]. In contrast to honeypots, IXPs are native part of the Internet infrastructure. They do not need to deploy detection schemes that expose to an attacker. They allow for monitoring of Internet traffic where networks intertwine, which also simplifies operational maintenance of a monitoring system.

3.9 Additional Analysis

3.9.1 Validation of the CCC Honeypot Platform

To verify that the CCC honeypot used in this chapter makes similar observations compared to previous honeypot studies, we compare various attack thresholds and analyze the convergence of our honeypot platform. CCC infers attacks using a threshold of 5 requests per sensor with no gap of more than 900 seconds before stop replying to requests. This is in contrast to other honeypots that set higher thresholds (*i.e.,* 100 packets and no gap of more than 3600 seconds [76] or 600 seconds [118]). Therefore, CCC applies a more sensitive attack detection,

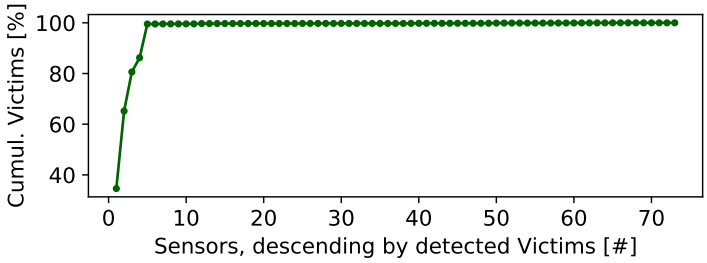


Figure 3.19: Honeypot convergence for the CCC platform. We observe a similar behavior compared to related projects.

which becomes apparent by a slightly higher number of reported DNS attacks for similar time ranges [76], [146].

A major property of honeypot platforms is the convergence of visible attacks by deploying a small number of sensors [76]. We reproduce the convergence analysis of prior work [76] for our data gathered at the CCC platform and make very similar observations. 99.5% of victims are already visible with only 5 sensors, see Figure 3.19. However, we require 50 sensors to cover 99.9% of victims due to a long-tail distribution, It is worth noting that the CCC platform is assumed to capture most DNS attacks (between 85.1% and 96.6%) on the basis of a capture-recapture statistical technique [146].

Overall, these results suggest that our honeypot platform behaves similar to related projects. The sensitive thresholds and convergence behavior suggest the observation of all DNS attacks, which we refute in Section 3.5.

3.9.2 Spoofed Traffic at IXPs

To narrow down the origin of attack traffic, we can only consider DNS requests because these packets originate from the attacking infrastructure. For attack events, we find a substantial asymmetry of requests and responses. 24% of attack events consists of only requests, 65% consist of only responses. Although legitimate Internet traffic is also subject to asymmetric paths [51], it is a common indicator that this extreme asymmetry is caused by IP address spoofing.

Even though the DNS requests contain spoofed source IP addresses, corresponding MAC addresses are unaltered. This allows us to identify the ingress network at the IXP. For each ingress AS, we retrieve the AS cone size using the CAIDA AS Rank [171]. Overall, we observe spoofed DNS traffic from only 18 ingress ASes, see Figure 3.20. 82% of spoofed traffic comes from a single ingress AS, and 12% from a second AS. Unfortunately, these requests are coming from networks with very large customer cones, which impedes attribution [45], [108]. We find only one stub AS introducing attack traffic, which is the only case that allows us to attribute the attack origin. Please note that this limitation is not specific to our IXP [45], [108].

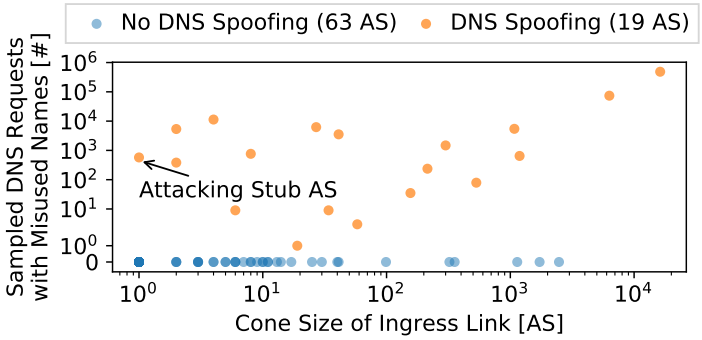


Figure 3.20: AS Cone size of Ingress Links for spoofed traffic. Most spoofed traffic originates from very large cones which makes tracing back almost impossible.

3.9.3 Cache Snooping to Check Name Popularity

To verify whether a name is frequently resolved globally, we use a modified cache snooping (CS) analysis. CS exploits the fact that popular names remain in DNS caches. Cached responses are identified by TTLs which are smaller than the default TTL defined by the authoritative nameservers. CS has been used to scrutinize the caching behavior at large public DNS resolvers [125]. Today, there is relatively little guidance backed by research about how to set TTLs, so operators usually reuse the same TTLs (*e.g.,* 5 minutes, 1 hour, 1 day) [106], which makes this analysis easier.

Phase 1: Identifying DNS Resolvers. We perform a scan of the complete public IPv4 address space and search for DNS amplifiers. Simply initiating DNS queries to all potential amplifiers and checking whether the DNS TTLs comply with default TTLs does not yield accurate results. This is due to the common DNS deployment, in which a DNS forwarder uses a recursive resolver and thus inherits current TTLs from this resolver. Hence, we first need to exclude forwarders from CS.

To identify DNS resolvers, we operate our own name and authoritative DNS server that responds with an `A` record set to the IP address of the resolver that directly queries our authoritative nameserver. By comparing both the IP address of the `A` record and target with the source IP address of the respond, we can distinguish resolvers (addresses match) and forwarders (addresses differ), details see [112]. This method allows for fast scanning with pre-built DNS queries and also limits the traffic at our authoritative nameserver since forwarders using the same resolver will return a cached entry. The relation between forwarders and resolvers has been measured before, but the previous methodologies [6], [73], [139] embed the IP address

of each target into the subdomain. This embedding requires the analysis of queries at the authoritative nameserver, which impedes reproducibility.

Phase 2: Assessing Name Popularity. After isolating the resolvers, we now can initiate a CS `ANY` scan to find uncommon cache activities. The reasoning here is that misused names are uncommonly often present in caches although being not popular, as measured by *e.g.,* the Alexa Rank. We find recently misused names with the help of our long-term monitoring tool. We sanitize responses by removing (*i*) answers with erroneous flags and codes (*e.g.,* rcode `REFUSED`), (*ii*) responses from obvious DNS manipulators, (*i.e.,* sources that change the TTLs or `A` records), (*iii*) duplicate responses from a single source. Then, we classify a response as a cache miss if all answer resource records contain a default TTL, a cache hit otherwise.

We focused our analysis on similarly popular `.gov` names. Please note that `americorps.gov` also has a larger (+25%) maximum TTL than `peacecorps.gov`, so it would be expected to produce more cache hits.

We utilize two anchor names to verify the correctness of this measurement. First, we reuse a name from Shadowserver which has well-documented, daily scanning times and TTLs. We initiate our scans *after* the daily expiration time to showcase correct cache evictions. Second, right before our CS scan, we create a completely new name, which should produce cache misses only. Still, the anchor names reveal a small amount of cache hits. We consider these cache hits to be the error rate of our measurements. We assume mutual resolver caches and DNS optimizers responsible for these errors.

3.10 Conclusion and Outlook

We studied the DNS amplification ecosystem from the Internet core, in combination with complementary data sources.

Our attack detection method for public peering points has enabled us to unveil distributed inter-domain attacks. Our results show that the DNS attack vector is more popular than previously captured by (even distributed) honeypots, a common vantage point in the context of reflection and amplification attacks. We were successful in tracking a prominent attack entity and identifying concrete attack patterns. Our study reveals that attackers are able to detect new abusable amplifiers quickly and reasonably change which infrastructure they abuse. At the same time, we find that attackers could achieve higher amplification by choosing (query) names more prudently. especially in the case of attacks utilizing spoofing and highly variable amplifier sets.

Our study also reveals that operators of various US federal government domain names break from recommended DNSSEC key rollover practices, which does not only exacerbate the amplification potential of various `.gov` names, but which our results can also tie to amplification attacks and attacker decision-making. For future work we plan to extend our methods to cover

a larger number of protocols and explore the fine-tuning of our thresholds to identify more subtle attacks.

3.11 Ethical Considerations

Our research may raise the following ethical concerns.

Privacy Invasion through Deep Packet Inspection. Our IXP vantage point provides a view into application-layer payloads. These data are particularly sensitive as they can contain personal information, or reveal the interests of users (*e.g.,* visited websites). However, we do not use the data to identify or study users. We also present only aggregated views, eliminating the possibility for third-parties to infer privacy-sensitive information. Finally, we focus on attack traffic, which consists of misused query names that do not disclose the interests of particular users.

Educating Attackers. This chapter presents misused query names in clear view, effectively showing the attackers suitable names for amplification. We argue that these names are already extensively misused in attacks, hence publishing them will not reveal new information. At the same time, we identified over 9000 names that can offer higher amplification than what we witness in practice. We will not divulge these names.

Alerting the Major Attack Entity. Releasing a DNS signature as detailed as presented in Section 3.6 could warn the attack entity responsible for more than half of the attacks. We argue, however, that publishing this information can do more good than harm as it will assist mitigation efforts by the research community.

Part II

Efficacy of Attack Mitigation

Chapter 4

Down the Black Hole: Dismantling Operational Practices of BGP Blackholing at IXPs

Abstract

Large Distributed Denial-of-Service (DDoS) attacks pose a major threat not only to end systems but also to the Internet infrastructure as a whole. Remote Triggered Black Hole filtering (RTBH) has been established as a tool to mitigate inter-domain DDoS attacks by discarding unwanted traffic early in the network, e.g., at Internet eXchange Points (IXPs). As of today, little is known about the kind and effectiveness of its use, and about the need for more fine-grained filtering.

In this chapter, we present the first in-depth statistical analysis of all RTBH events at a large European IXP by correlating measurements of the data and the control plane for a period of 104 days. We identify a surprising practise that significantly deviates from the expected mitigation use patterns. First, we show that only one third of all 34k visible RTBH events correlate with indicators of DDoS attacks. Second, we witness over 2000 blackhole events announced for prefixes not of servers but of clients situated in DSL networks. Third, we find that blackholing on average causes dropping of only 50% of the unwanted traffic and is hence a much less reliable tool for mitigating DDoS attacks than expected. Our analysis gives also rise to first estimates of the collateral damage caused by RTBH-based DDoS mitigation.

4.1 Introduction

The Border Gateway Protocol (BGP) is used to exchange IP prefix reachability information between Autonomous Systems (ASes) to form the global Internet. Yet, one BGP application has the opposite effect in practice: Signaling Remotely Triggered Black Hole filtering (RTBH) through BGP requests a neighboring AS to discard traffic destined towards an owned IP prefix. The most prominent and well-established use case for RTBH filtering is the mitigation of volumetric Distributed Denial-of-Service (DDoS) attacks. Recent attacks peak beyond multiple

Tbps (Terabit per second) [246]. DDoS attacks build upon simple to exploit IP address spoofing [189], [191] in combination with amplification characteristics of network protocols such as NTP, DNS, or cLDAP [30], [64]. These attacks deplete network bandwidth to suppress legitimate traffic towards a destination IP. In consequence, a network or web service is not reachable anymore. Still, DDoS attacks do not only cause damage at the attacked system itself, but can also overwhelm the infrastructure of intermediate or upstream networks [222]. Such collateral damage often impairs common customers badly.

Intermediate ASes mitigate the collateral damage of DDoS traffic passing through their infrastructure by signaling RTBHs to their neighbors that specifically cover the target address of the DDoS attack. Thereby, volumetric attack traffic is dropped before it reaches the final destination and alleviate the damage to the network infrastructure under attack. Internet exchange points (IXP) are particularly well suited for this kind of prevention, since they provide a convergence point where hundreds of ASes meet and exchange inter-domain traffic [3], [33].

RTBH filtering is a light-weight and easy to use tool. It is widely deployed and can be highly effective, that is why RTBH is a well established reactive DDoS mitigation technique today [49]. On the downside, RTBH is a coarse granular mechanism that drops all traffic to a specific prefix, and does not provide information about the attack traffic while it is ongoing. Therefore, advanced alternatives such as ACL filters [189], BGP FlowSpec [54], [235], [267], and Advanced Blackholing [34] have been introduced. Yet, RTBH continues to play a significant role in DDoS mitigation.

Understanding RTBH's operational intricacies and use cases as well as its traffic patterns and efficacy are crucial for understanding the effectiveness and success of RTBH and for the evaluation of its alternatives. Furthermore, an in-depth investigation can help to uncover issues with the established ways of using RTBH on the Internet. However, understanding the operational practices as well as the corresponding traffic patterns at large scales is limited in both academia and industry. A number of publications on RTBH traffic patterns describe representative investigations of single events [33], [34], but no large-scale, statistical analysis with in-depth empirical evidence exists.

We start with exploring the operational practices of RTBH at a large IXP and separate RTBH by their inferred use case. Thereby, DDoS attack mitigation RTBHs can be separated from other use cases and investigated in detail. We analyze the traffic patterns of DDoS RTBHs and gain thorough insights how these are connected to operational practices. To our surprise, we find use patterns and deployment of RTBH in the wild that differ widely from common expectations.

Our contributions are as follows:

1. A description and characterization of RTBH use cases based on the literature as well as industry expert interviews

2. We collect and analyze unsampled RTBH data over a period of three months and classify RTBH events by their use cases

3. We uncover the statistical efficacy of RTBH traffic dropping over a large data set of 590 million sampled flows

4. We present a detailed correlation analysis between DDoS attacks on the data plane and RTBH signaling on the control plane

5. We provide insights into the detrimental effects of dropping attack traffic completely and quantify the beneficial traffic

This chapter is organized as follows. We present the understanding of RTBH use cases and literature in Section 4.2. Section 4.3 introduces our control and data plane data set. We investigate which features of RTBH and to what extent they are used in Section 4.4. Empirical evidence on the traffic characteristics of DDoS mitigation RTBHs is presented in Section 4.5 followed by an investigation on collateral damage of RTBH filtering in Section 4.6. We discuss our findings on the background of discussions with industry experts in Section 4.7 before drawing a conclusion in Section 4.8.

4.2 Background and Use Cases

RTBH filtering is thought to be originally conceived to mitigate DDoS attacks on the Internet. Its low operational overhead to signal blacklisting makes it attractive for other use cases as well. In this section, we introduce RTBH as a tool for protecting infrastructure from DDoS attacks. Furthermore, we identify the use of RTBH in the context of prefix squatting protection and content blocking as well. Finally, we describe the expected blackhole characteristics for every use case based.

4.2.1 RTBH Primer

Remotely Triggered Blackholing uses BGP to signal blackholes, in contrast to other blackholing approaches such as access control lists. To start (or stop) a blackhole at IXPs, a member sends a BGP announcement (or withdrawal) to the IXP route server. The route server distributes the blackhole route to all or a subset of IXP peers, including a specific next hop IP address (*i.e.,* the blackhole). It is worth noting that any peer applies local BGP policies on the received blackhole route to decide whether to accept or filter the received blackhole route, as the peer does for any other route. Based on this decision, subsequent data that matches the blackhole route will be forwarded to the IXP infrastructure and dropped (see Figure 4.1).

A known drawback of RTBH is the collateral damage due to the rather coarse granularity of destination IP prefixes [54]. RTBH drops *all* traffic towards the prefix under protection,

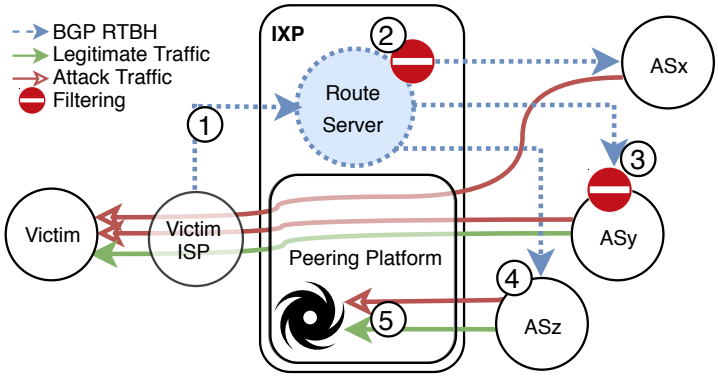


Figure 4.1: Remotely Triggered Blackholing (RTBH) at IXPs: ❶ RTBH announcement via BGP, ❷ Propagation filter, ❸ BGP policy rejects RTBH route, ❹ BGP policy accepts RTBH route, ❺ Packet drop.

Table 4.1: Literature-based expected characteristics of RTBHs by use case.

Use Case	Trigger	Prefix Length	Reaction Latency	Duration	Traffic	Target
Infrastructure Protection	Automatic Detection and Triggering	/32	Secs-Mins	Mins-Hours	Attack	Server
Prefix Squatting Protection	Manual	$\leq$ /24	n/a	Months	Scanning	None
Content Blocking	Manual	/32	n/a	Weeks-Months	Normal	Server

i.e., legitimate traffic as well as attack traffic, because it cannot distinguish services on the transport layer.

4.2.2 Infrastructure Protection

RTBH was designed to prevent forwarding of unwanted traffic [217], [285], *e.g.,* (*i*) attack traffic (DoS), (*ii*) incoming scan traffic [37], or (*iii*) Internet background radiation [119]. For the latter two, the traffic volume is comparatively small and operational best practices such as firewalls and static ACL filters [189] are adequate solutions. In contrast, today's terabit-level DDoS attacks are a serious threat to the operation of Web services [216], [246], and even challenge the Internet backbone infrastructure [222]. To alleviate the negative impact on the Internet infrastructure, RTBH is used as a cheap and convenient technique to filter unwanted traffic at intermediate network nodes [49]. Such a central location to blackhole unwanted traffic

are IXPs [33]. Traffic of hundreds of ASes can be dropped or filtered on the IXP switching platform [34], making IXPs a good vantage point for this kind of studies.

For the usage of RTBH at an IXP, we expect a significant rise of inter-domain traffic volume, seen by a member. In reaction, this member will send most likely a `/32` RTBH. Note, RTBHs are announced and withdrawn constantly by the victims to gather attack status information—if the traffic is discarded no telemetry data is available [49]. Thus, our assumption is to observe a temporally correlated anomalous traffic peak directly before the first RTBH during an attack event. Since 75% of DDoS attacks are volumetric attacks [250], we expect to see a change in the port distribution, i.e., more traffic from amplification candidate protocols often used in DDoS attacks such as DNS, NTP, or memcached. The average duration of DDoS attacks was 218 minutes by the end of the year 2018 [218]. We also assume that servers are a frequent target of DoS attacks, but also attacks to clients have been observed before [30]. Attacking business-critical, often used servers allows the attacker to exert pressure and blackmail the victims for financial gain. Profit margins of a DDoS attack can reach up to 95% [180]. Consequently, we should be able to observe legitimate, regular traffic patterns and compare them with attack traffic, which allows quantifying collateral damage of RTBH as a DDoS mitigation approach.

Based on measurements with an Internet telescope, related work shows that [65] RTBH is usually triggered automatically after a short reaction time. We expect to see this behavior also at our vantage point. RTBH triggered by DoS mitigation mechanism should be rather short, optimally only for the duration of the attack.

In summary, we anticipate the following order of actions for this use case: First, the attack event takes place in the form of a DDoS attack. This distributed attack utilizes multiple attack vectors and attacks either state (e.g. TCP Syn attack) or capacity (UDP-Amplification) of its victim. The attack starts with the increase of unwanted traffic and ends with its disappearance. Second, two parties might react to the attack event. Either the victim itself announces a RTBH or one of its upstream providers, whose links are a collateral damage of the attack. Since all traffic is dropped, the victim is blinded about the progression of the attack. Hence RTBHs will be withdrawn to test for attack traffic and then re-announced. Not only bogus traffic is dropped but also legitimate flows, which is the collateral damage of the mitigation mechanism. We expect to see different traffic properties for the legitimate and attack traffic.

4.2.3 Prefix Squatting Protection

The increasing scarcity of freely available IPv4 address space and its importance not only for legitimate businesses but also for spammers alike increases the pressure on unused IPv4 address space. Prefix hijacking is a well known phenomenon where IP prefixes are taken over by third parties on the Internet, either erroneously or with malicious intent [100], [141]. Mitigation techniques such as RPKI exist, but are still not sufficiently deployed to completely prevent prefix hijacking [127]. IP prefix squatting is a variant of prefix hijacking, where third parties take over

address space that is assigned to another AS but not announced from this legitimate origin [153]. These prefixes are easier to hijack because there is no competitive announcement [16], [223], [258], [141].

One common mitigation technique for prefix squatting is to announce the assigned address space. To ensure the address space is not used at the same time, the same prefix is announced as an RTBH.

Prefix squatting is used in practice, *e.g.,* to send email spam from valid address space and to prevent backtracking [16], [100], or for internal infrastructure addressing in case of address shortage [213]. Considering the severe negative consequences of prefix squatting and the low effort to mitigate, we expect to see applications of this use case in the wild. In fact, we find very few incidents that may refer to RTBH to protect against prefix squatting.

4.2.4 Content Blocking

Applying RTBH to block clients from accessing content occurs rarely but is possible. Giotsas *et al.* [49] found that attackers (*e.g.,* port scanners, vulnerability scanners) and not victims have been blocked by network operators to prevent access to server content.

Another motivation for the deployment of BGP blackholing is censorship. RTBH can be used to block traffic towards an IP address hosting undesirable content. Compared to access control lists, RTBH reduces operational burdens as it simplifies the maintenance of blacklists [195]. Instead of configuring ACLs on every router separately, a single router maintains the master file and signals the blackhole routes to the peers via BGP. This is specifically beneficial in scenarios that require frequent and rapid changes. We consulted several network operators whether this case has been observed in real-world. Even though the answer was negative, we include this use case for completeness.

In both scenarios, RTBH is characterized by midterm, stable RTBHs routes, triggered by few BGP updates. In particular, *blocked traffic does not reflect typical DDoS traffic patterns.*

4.2.5 Expected Characteristics

All three RTBH use cases (infrastructure protection, squatting protection, and content blocking) are expected to exhibit different characteristics in terms of BGP signaling and data traffic. Both content blocking and squatting protection are expected to show long-term and stable RTBH routes without attack traffic. In terms of prefix lengths, however, they should differ. Content blocking is expected to use very specific prefixes, *e.g.,* `/32` to filter the addresses of content hosts. Squatting protection, on the other hand, is expected to predominantly cover $\leq$ `/24` prefixes. RTBH usage for infrastructure protection is expected to show very specific prefixes, similar to content block, but should exhibit DDoS attack traffic during or shortly before the RTBH. We summarize our observations in Table 4.1. It is worth noting that the classification is not strictly exclusive but indicates common tendencies.

4.3 Data Corpus

This analysis is based on data from a large European IXP that offers remotely triggered blackholing as a service to its members. Our data sets contain three months of passive control as well as data plane measurements. Since more than 95% of the traffic and more than 98% of RTBH events at this IXP are from IPv4, we focus this work on IPv4, only.

We now present our data sets in more detail including the different data sources, potential challenges when aligning different sources, and a brief overview about blackholing activity at our vantage point.

4.3.1 Control and Data Plane Data Sources

We measure data on the control plane (*i.e.,* BGP) to identify remotely triggered blackholing. For a better understanding of the RTBH impact on the blackholed IP prefixes, we capture traces on the data plane (*i.e.,* IPFIX). All measurements are consistently taken from September 26, 2018 until January 11, 2019. We had to exclude few hours and December 6 due to infrastructure maintenance.

Control Plane. An AS initiates (or terminates) RTBH at the IXP by sending BGP update messages with a specific BGP community [212] to the public IXP route server, which distributes this information further to either all of its peers or to a subset. We collect these messages and gain the following information: (*i*) when the blackholing should start and stop, (*ii*) which AS triggered RTBH, (*iii*) which ASes should send data to the blackhole, and (*iv*) the origin AS of the RTBH prefix. The time resolution of the collected, RTBH-related BGP messages relies on the NTP protocol for synchronization and, therefore, is expected to be accurate at a level of 10ms [19].

Note that RTBHs established in bilateral (private) peering is out of scope of this work.

Data Plane. We collect IPFIX packet samples (1 out of 10,000 packets) of incoming traffic from peers at all member-facing ports of all network devices at the edge of the IXP switching fabric. On average, we sample 70,000 packets-per-seconds. From the collected packets, we extract the packet sizes, source and destination MAC addresses, destination IP addresses, source and destination transport ports. Based on this data, we can attribute 590 million packets as originated from or addressed to any of the blackholed IP prefixes. To identify the ASes that exchange the packets at the IXP, we map source and destination MAC addresses of the sampled packets to the router interface addresses of the ASes connected to the IXP switching fabric. This collection includes 47,000 IPFIX flows received from internal system of the IXP as a source or destination device, *i.e.,* 0.01% of the total number of flows. The internal traffic is removed from the data set before further processing. Thereby, we have full, 1 in 10000 sampled visibility

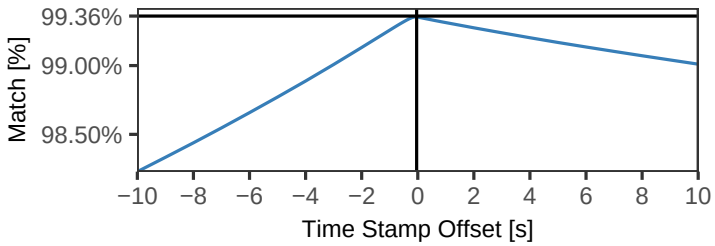


Figure 4.2: Maximum likelyhood estimate for time offset between control and data plane sources.

of all member traffic coming into the IXP switching fabric. This data set is used for analyzing both forwarded traffic and dropped, blackholed traffic.

Identifying Dropped Traffic. The dropping of blackholed traffic at this IXP is implemented with the help of a unique (blackhole) MAC address that does not forward data. By announcing a special next hop via BGP, which in turn maps to this MAC address, we redirect packets to the blackhole. Consequently, the data is dropped, and we can mark any sampled packet with destination to the blackhole MAC as dropped traffic.

Using the sampled data of dropped packets in correlation with our control plane measurements, we calculate the amount of dropped traffic triggered via the route server. We find that on average, 95% of the dropped bytes are controlled by RTBH signaled via the route server and therefore represent the majority of the observed traffic. The remaining 5% belong to traffic that was dropped because of other RTBH sources.

Accuracy of Timestamps. All measurement devices synchronize their system time using NTP in the local subnet, which allow for a time series analysis between both data sets. Deviations, however, are still possible and need careful verification. To quantify errors, we measure which share of the sampled packets was dropped because of blackhole announcements visible in the recorded BGP data and which share was not dropped. Based on the timestamps from the control and data plane, we apply a maximum likelihood approach to estimate the time offset between both data sets.

During the measurement period, $\approx 50\,\mathrm{M}$ packets addressed to RTBHs were dropped by the blackholing service. The offset between the control and the data plane is depicted in Figure 4.2. The maximum overlap is 99.36% for an offset of $-0.04\,s$, showing that both data sources are sufficiently consistent in time.

Note that control plane data is needed for the subsequent analysis as we are also interested in events where BGP announcements signal RTBH but the receiving ASes still forwards data (*i.e.,* does not select the announced RTBH prefix as best route).

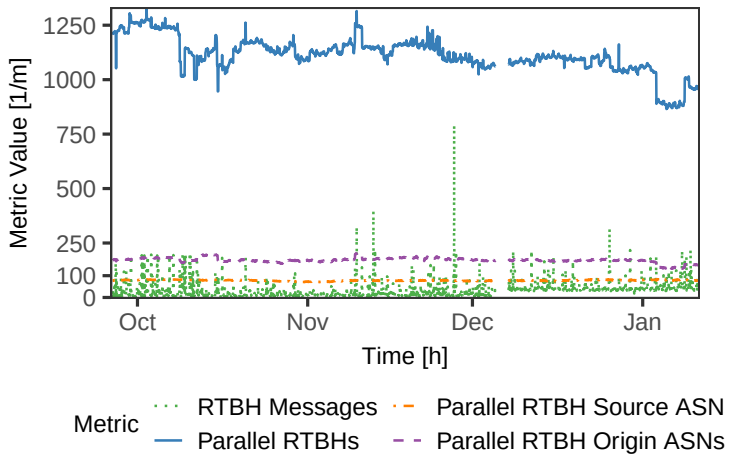


Figure 4.3: Number of active parallel RTBH over time.

4.3.2 RTBH Load

Figure 4.3 provides an overview on the load of the RTBH signaling. During the measurement period, 830 member ASes have been connected on average to the IXP peering platform. 78 of these peers announced 1,107 RTBHs for 170 origin ASes at any given minute in the observation period. At most 1,400 RTBH prefixes were active during the same minute, which is less than two prefixes per connected peer. The number of RTBH-related BGP messages stays below 500 messages with a few spikes of up to 600 and one spike up to 793 messages per minute or less than 14 messages per second.

These numbers illustrate nicely that RTBH adds negligible overhead on the control plane in terms of memory and processing. This resource efficiency might explain the popularity of using RTBH to protect the Internet infrastructure.

4.4 Acceptance of RTBH Features

The efficacy of RTBH-based filtering relies on both receiving related BGP announcements from the route server *and* accepting the received routes as best paths. In this section, we answer the two questions: Do network operators try to reduce the negative impact of RTBH? Do network operators accept RTBH announcements to filter traffic?

4.4.1 Using Targeted Blackhole Routes

The RTBH service at our vantage point allows network operators to instruct the route server to selectively announce RTBHs to specific ASes on the peering platform, which reduces collateral damage. Using BGP communities, the victim AS can select to which peers its RTBH announce-

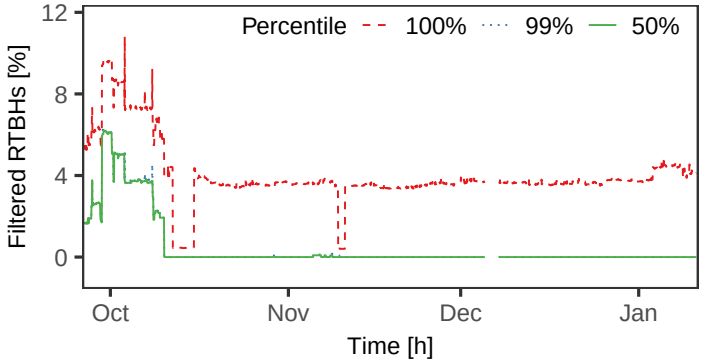


Figure 4.4: Percentage of all announced blackholes at a given time that are filtered and are not visible to 100/99/50 percentiles of peers on the peering platform. 99% and 50% quantiles overlap such that only the 50% quantiles are visible in large parts of the figure.

ment will be forwarded by the route server. Thereby, unfiltered communication continues with unaffected neighbors.

It should be preferential for an operator to affect only the ASes transporting malicious traffic by RTBH. We investigate this hypothesis by analyzing the BGP communities which are collected in our control plane data set. Thereby, we are able to obtain the specific view of every BGP peer on the set of blackholed prefixes at every point in time throughout the measurement period.

Figure 4.4 shows the percentage of all announced blackholes at a given point in time that are filtered to not be visible at all peers. The quantiles indicate which part of the announced blackholes are filtered and, therefore, not visible to all (100%), 99%, and the median (50%) of the connected peers. Significant deviations of parallel RTBHs are visible during some weeks at the beginning of October 2018. At this time, the median of the peers saw up to 6.2% fewer RTBHs than the route server and a single peer even 10.8% fewer. After mid October, however, the median and 99% percentiles of the peers dropped down to at most 0.2% fewer RTBHs compared to the full visibility at the route server. The peer with the fewest received RTBHs saw only a minus of up to 4.9% parallel RTBHs. Based on these findings we conclude that selective filtering and announcements are the exception and commonly not used to reduce the collateral damage for targets of DDoS attacks.

4.4.2 Accepting Blackhole Routes

Any BGP peer that does not accept a blackhole route from the route server will continue to forward the traffic that was intended to be filtered. Acceptance of this route is beyond the control of the triggering AS, but subject to local BGP policies of the receiving peer. Using the RTBH visibility information derived in Subsection 4.4.1, we calculate the fraction of data that a router transmits even though it received a blackhole announcement.

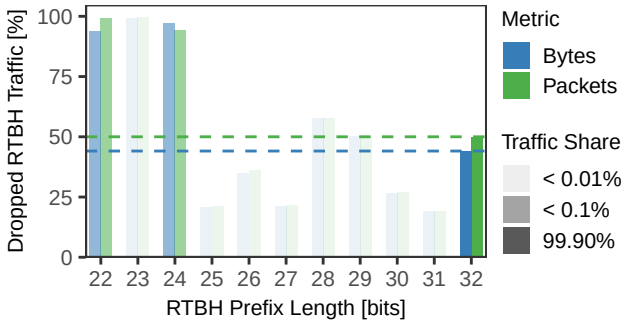


Figure 4.5: Observed shares of dropped traffic by RTBH prefix lengths; dashed lines denote averages. The traffic shares are visualized as opacities of the bars.

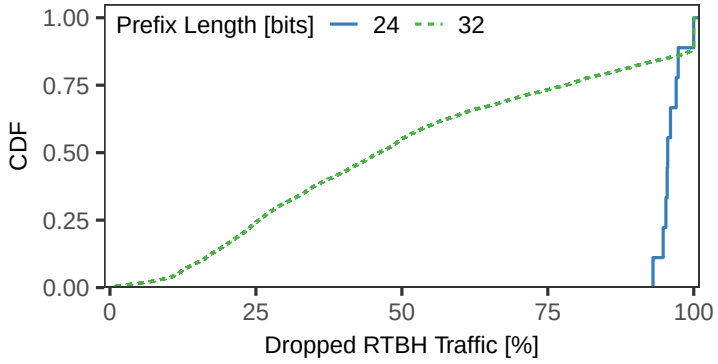


Figure 4.6: Distribution of dropped RTBH traffic shares for selected prefix lengths.

Figure 4.5 depicts the amount of traffic dropped for all active blackhole prefixes relative to the overall amount of traffic for those prefixes during the blackhole distinguished by prefix length. The opacities of the colors visualize the RTBH traffic share of the respective prefix lengths compared to the overall blackhole traffic. For example, 99.9% (highest opacity) of the overall RTBH traffic was sent to `/32` prefixes and $< 0.01\%$ (lowest opacity) of the traffic to `/25` or `/26` prefixes. The dashed lines show the average drop rates of RTBH announcements considering all RTBH prefix lengths.

It is clearly visible that the vast majority of traffic for blackholing corresponds to `/32` prefixes. To our surprise, however, only 50% of the packets (or 44% of bytes) are filtered, *i.e.*, more than half of the traffic continues flowing to the victims. In contrast, blackhole routes to less specific prefixes (`/22`, `/23`, and `/24`) are accepted as best paths in 93% – 99% of the cases. Those prefix lengths are common in BGP announcements in general [274]. Considering that more specific prefixes (`/25` - `/31`) exhibit a behavior similar to `/32` in terms of the dropped rate, we assume incorrectly configured BGP policies because accepting (RTBH) prefixes longer than /24 bits requires to change the common BGP configuration and to whitelist such announcements.

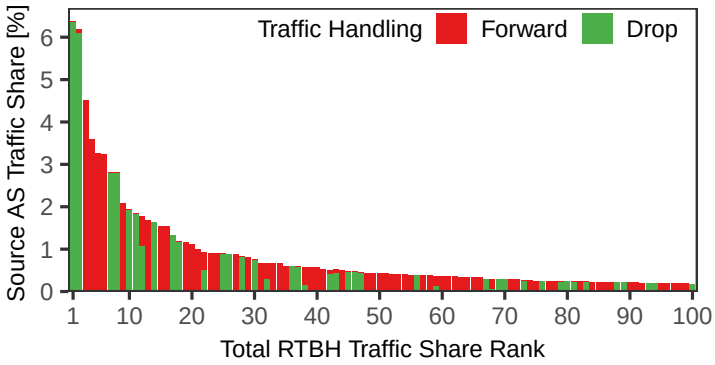


Figure 4.7: Reaction of top 100 source ASes by traffic share to /32 RTBHs.

To better understand the varying acceptance for different prefix lengths, we investigate the behavior of `/24` and `/32` prefixes in more detail. Figure 4.6 shows the CDF of the observed drop rate for these two prefix lengths. The drop rate of `/24` RTBH prefixes varies between 82% and 100% with a median of 97%, making `/24` blackholes a fairly predictable configuration to successfully mitigate unwanted traffic.

For `/32` prefixes, the blackhole traffic drop share ranges between almost zero and 100%, with 30% for the first quartile, 53% for the median, and 88% for the third quartile. This wide distribution results in a high uncertainty regarding the expected effectiveness when announcing an `/32` RTBH. In the median case, the unwanted traffic will be reduced to at least half, but in some cases an RTBH announcement will cause no data reduction at all. Triggering RTBH for single hosts (`/32` prefixes) is often very appropriate, but may lead to a rather unpredictable reaction in reducing unwanted traffic.

To characterize the AS peers further that ignore `/32` announcements and cause low drop rates, we investigate the top 100 source ASes that contribute most of the traffic volume to `/32` blackholes. Figure 4.7 shows the relative amount of dropped and forwarded traffic by these ASes that all together account for over 85% of the total traffic to RTBHs, many of which are heavy hitters in RTBH scenarios. Only 32 of these ASes drop more than 99% of the traffic to RTBHs. 55 of the top source ASes forward only less than 1% of the traffic to the blackhole route. Interestingly, 13 ASes exhibit an inconsistent behavior as they send significant parts of the traffic to the blackhole route and forward other parts of the traffic to the victim AS.

For a deeper dive, Figure 4.8 groups the top 100 ASes by their AS types and scopes based on PeeringDB data. Most ASes that do not (or partially) accept blackhole routes are network service providers (NSPs), which comes as a surprise. We expected these companies to be well-prepared for complex BGP configuration tasks. One reason for the contrary may be that global NSPs deploy alternate measures of DDoS mitigation, outside the public peering ecosystem.

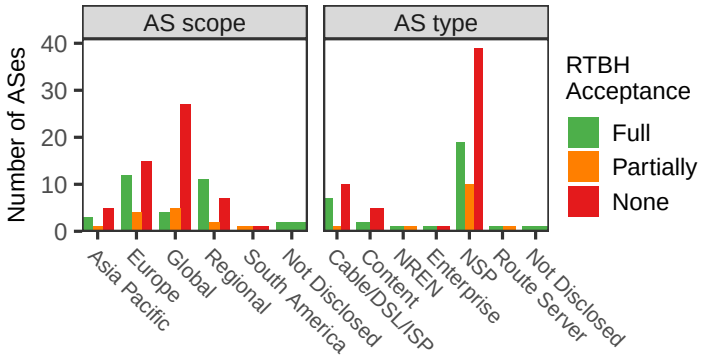


Figure 4.8: The PeeringDB organization types of the top 100 source ASes by traffic share sent to /32 RTBHs.

4.5 Evidence of DDoS Attacks

The default use case of RTBH is considered DDoS protection. In this section, we explore this common assumption by correlating events at the data and control plane. This analysis requires a careful modeling of common DDoS and mitigation patterns to differentiate the signals at the control and the data plane.

4.5.1 Preparatory Steps

Blackholes for infrastructure protection are announced and withdrawn repeatedly to check whether the attack event is still ongoing (see Figure 4.9). Theoretically, RTBHs block all traffic and hence also indicators about the attack status. In practice, as we have shown in the Section 4.4, some traffic still arrives. However, due to the high variance in actual drop rates, this remaining traffic is a highly unreliable source of status information. That is why we still see frequent re-announcement patterns.

We use this on-off pattern to identify RTBH announcements that target at the same *attack event* and merge them into a single *RTBH event*. Each RTBH event reflects the mitigation process after the attack was detected. Small gaps between multiple RTBH announcements that belong to the same attack event are likely to show attack traffic as well. To prevent the misclassification of traffic, we include traffic during these gaps into RTBH events. The challenge is to find an appropriate time threshold Δ between consecutive RTBH announcements, which distinguishes RTBH announcements that belong to the same or another RTBH event.

For each blackhole update bh_i of a single RTBH event, the following applies with respect to the observed timing between BGP withdrawals and announcements:

$$|\, bh_i[\text{withdraw}] - bh_{i+1}[\text{announce}] \,| \leq \Delta$$

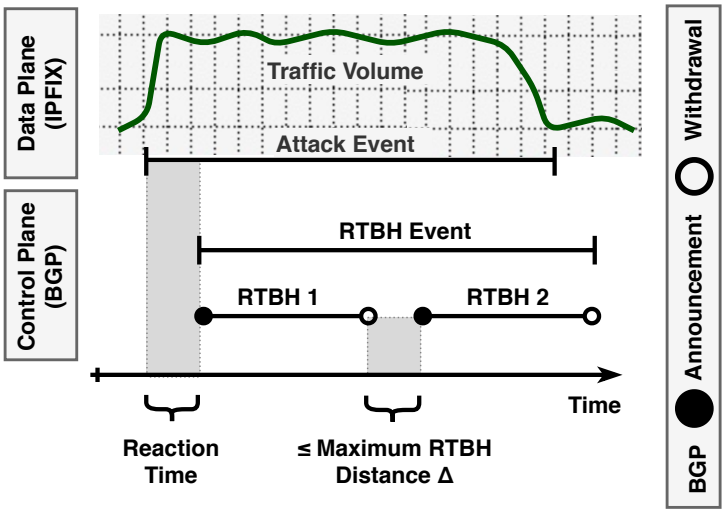


Figure 4.9: Attack and RTBH events: A sequence of re-announced RTBHs.

Now, we need to find an appropriate merge threshold Δ. For this, we consecutively increase Δ and inspect the amount of blackhole events, relatively to the overall number of RTBH announcements (see Figure 4.10).

The last significant effect is visible up to about $\Delta = 10$ minutes. Furthermore, a 10 minute Δ is consistent with the delay found between the detection of DDoS traffic and the triggering of a blackhole [65]. Therefore, even if the blackhole originator mistakenly disables a blackhole while an attack event is still ongoing, a newly triggered blackhole would be part of the correct, preceding blackholing event. For this Δ, 400k blackhole announcements are grouped into only 34k RTBH events, which is a reduction to 8.5%. We highlighted the lower bound $\Delta = \infty$ (red dashed line), for which the number of RTBH events equals the number of unique blackholed prefixes.

Fixing the merge interval to the reasonable threshold of $\Delta = 10$ minutes, we can now use the aggregated RTBH events to examine the traffic before and during RTBH events.

4.5.2 Visibility of Pre-RTBH Events

Assuming that most RTBH events are triggered by volumetric DDoS attacks, we search for traffic anomalies during the 72 hours before the first RTBH announcement. We refer to this time range as the pre-RTBH event. If a pre-RTBH event contains an anomaly, the corresponding RTBH event is said to have a preceding anomaly.

First, we identify all pre-RTBH events that include at least one sampled packet and thus may give additional insights into the traffic behavior. We aggregate into five minutes slots. The results are shown in Figure 4.11. Surprisingly, traffic appears for only 18k of the total 34k pre-RTBH events. This means that 46% of all pre-RTBH events did not exhibit a sufficient

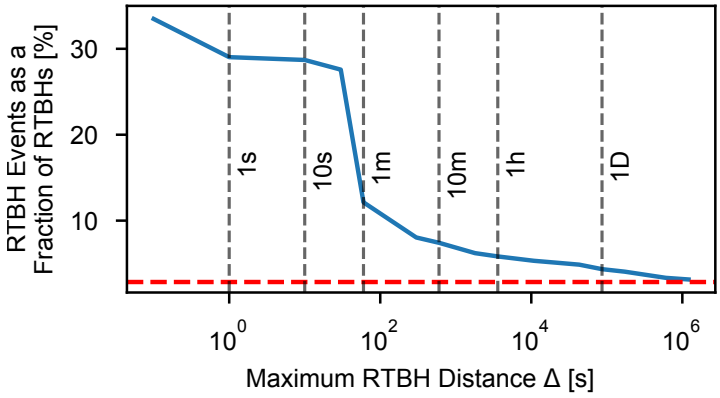


Figure 4.10: Fraction of blackholing events in all RTBH announcements.

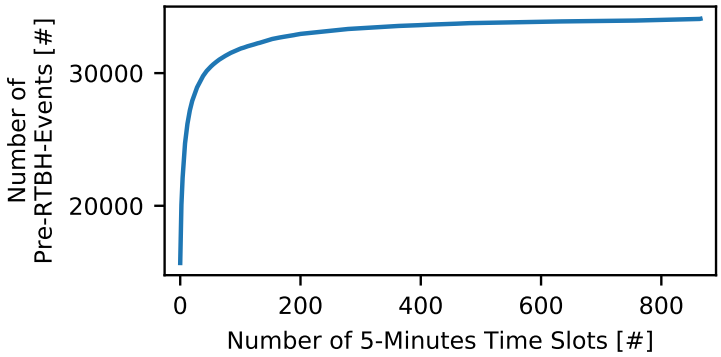


Figure 4.11: Number of time slots contributing traffic samples within 72 hours before RTBH started.

amount of packets to be sampled, even though our vantage point is one of the largest IXPs. For these cases, data plane monitoring cannot explain the root cause of the RTBH events. 13k of pre-RTBH events exhibit data for at most 24 time slots during a total of 2 hours (see Figure 4.11). This indicates very sparse data. Manual inspection shows that those pre-RTBH events represent incidents where unusually high traffic peaks are visible shortly before the first RTBH announcement. This motivates further investigation, which we continue in the next section.

4.5.3 Classification of Pre-RTBH Events

We want to automatically describe and classify the traffic behavior for the pre-RTBH events. For this, we observe five traffic features: (*i*) number of packets, (*ii*) number of flows, (*iii*) number of unique source IP addresses, (*iv*) number of unique destination ports, and (*v*) number of non-TCP flows.

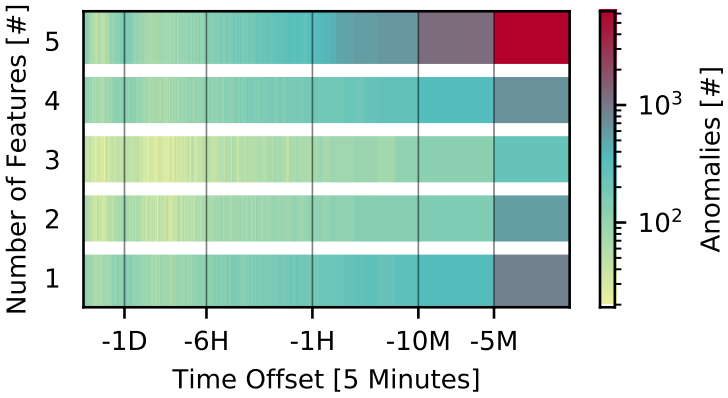


Figure 4.12: Level and Time Offset of Traffic Anomalies during pre-RTBH events.

As a straightforward indicator, we use Exponentially Weighted Moving Average (EWMA), a simple sliding window mechanism to detect unusual traffic peaks. For each detection, we consider a 24 hours window, which shifts every five minutes and spans 288 time slots. The most recent values have the highest weight, the oldest the smallest weight. We use the same notation as common data analysis tools [256]. The decay parameter α and the weight w are calculated as follows:

$$\begin{aligned} \alpha &= 2/(s+1), \text{with } s = 288 \\ w_i &= (1-\alpha)^i \end{aligned}$$

Then, the weighted moving average is defined as

$$y_t = \frac{\Sigma_{i=0}^{t} w_i x_{t-i}}{\Sigma_{i=0}^{t} w_i}$$

Please note that we require a full window for an anomaly detection. This means that no anomaly can be found during the first 24 hours. We perform an EWMA anomaly detection independently for each feature. Values are tagged as anomalous when they exceed the moving average by $2.5 \cdot \text{SD}$ (standard deviation). Then, we count the number of features that have an anomalous traffic peak for each time slot. We refer to this as the anomaly level.

Figure 4.12 shows the distribution of all anomalies by level and time offset relatively to the RTBH event start. There is a clear trend for the time-lag between anomalies and RTBH events in that most anomalies occur up to ten minutes before the first RTBH announcement. This short reaction time indicates automatic DDoS mitigation tools. Usually, all five features show anomalous behavior shortly before the blackhole. We also find multiple cases in which an anomaly was found only for one of the five features. This emphasizes the importance of a multi-sided traffic analysis to detect individual anomalies.

Based on these results, we now are able to classify pre-RTBH events into three classes: Pre-RTBH events (*i*) without sampled traffic, (*ii*) with sampled traffic, but no anomaly before the

Table 4.2: Class Distribution of Pre-RTBH events.

Pre-RTBH Event Class		% Events
Data	Anomaly $\leq$ 10 min	
✘	–	46%
✓	✘	27%
✓	✓	27%

RTBH event, and (*iii*) with sampled traffic and at least one anomaly before the RTBH event. The first class has been quantified in the previous section.

We find 9k pre-RTBH events (27%) with an anomaly up to 10 minutes before the initial RTBH announcement. Also, we find only 11k pre-RTBH events (33%) with an anomaly up to 1 hour before the initial RTBH announcement. This means that only one third of all RTBH events are triggered by volumetric traffic changes. This finding deviates significantly from the original intention of RTBH as a tool of DDoS mitigation. We summarize these results in Table 4.2.

Relevance of Anomalies.

Our approach allows to identify volumetric DDoS anomalies, whether it is a spoofed or unspoofed, direct or reflected attack. Despite being able to detect sudden peaks, we are not able detect long-tailed DDoS attacks such as Slowloris. However, this type of attack does not produce large traffic volumes and is not expected in the context of RTBHs. The median DDoS attack size in mid 2018 was 1287 Mbps [221]. Dividing by a MTU of 1500 Bytes, this corresponds up to 100k packets per second on the IXPs switch fabric. Due to the large number of packets even for median sized attacks, we expect to observe anomalies even though our data is sampled.

We are well aware of the fact that we miss ground truth data to perform a validation of our methodology. Note that a correlation with public news about DDoS attacks is not necessarily helpful due to the possible use of other mitigation tools. For example, during our measurement period Imperva reported one of the largest attacks ever observed [275] but their mitigation portfolio contains only scrubbing center redirections and DNS diversions. Nevertheless, we tried to verify the largest attacks observed. Although many companies are reluctant to publicly disclose information on attacks and hence admit reachability issues, we were successful in some cases. E.g, an online shop, which has the fifth largest RTBH event by attack volume, confirms the attack and time with a public announcement [262]. We saw an active RTBH event (with a preceding anomaly) for more than 7 hours.

Being a best-effort analysis, we were very cautious and performed multiple consistency checks and manual inspections to ensure the correct detection of traffic spikes. Fortunately, a clear trend becomes apparent, which justifies a simple detection methodology. Either we do not

observe any traffic changes at all or very significant bursts. To further substantiate this observation, we now analyze the relevance of the anomalies compared to the average traffic behavior 72 hours before the RTBH event begins. Since most anomalies occur ≤ 5 minutes before an RTBH event starts, we focus on this last time slot.

For every traffic feature, we calculate its relative rise during the last five minutes prior to the RTBH event, which we refer to as *Anomaly Amplification Factor*. This factor is depicted in Figure 4.13. The last time slots often do not contain traffic, either because the whole pre-RTBH event does not contain any data at all, or only during other time slots. Nevertheless, if packets were sampled during the last five minutes, a multiple of up to 800 can be observed. In 15% of the cases this slot shows the maximum value of the entire time range.

These results indicate very strong changes in traffic patterns that occur with the anomalous behavior. It does not require any fine-tuning. Instead, we tested extreme configurations such as thresholds of $10 \cdot \mathrm{SD}$ (instead of 2.5) with very stable results.

4.5.4 Classification of RTBH Events

We now inspect the traffic during the RTBH events. Even though we sample packets at an Internet exchange point of very large data volume, the sampling does not necessarily capture packets during an RTBH event. To gain insight into this general measurement challenge, we first classify the events according to traffic visible on the data plane. Then, we correlate the visible traffic with commonly misused services.

Overall, the sampling captured packets for only 29% of all RTBH events, albeit we applied a high sampling rate of 1 out of 10,000 packets. More than half of the RTBH events that feature captured data have also a preceding anomaly within 10 minutes. These incidents account for 18% of all RTBH events. Interestingly, one third of the RTBH events with a preceding anomaly have no traffic during the RTBH event. We explain this by (*i*) very short-lived DDoS attacks and (*ii*) other mitigation points on the Internet that drop the attack traffic before reaching our vantage point (*e.g.,* scrubbing [65]).

We now analyze the network service misused to generate attack traffic. We expect to observe attack traffic during RTBH events with a preceding anomaly. This is why we identify the protocol distribution for each RTBH event for which a (preceding) anomaly was detected *and* the monitoring system sampled traffic. We find that UDP is the most prevalent transport protocol in this context: (*i*) 99.5% UDP, (*ii*) 0.3% TCP, (*iii*) 0.1% ICMP, (*iv*) 0.1% other. This protocol distribution differs significantly from the normal traffic mix at IXPs [3], [34].

For the dominant UDP traffic, we check whether the RTBH events relate to attacks based on common UDP amplification protocols. To prevent biased results due to outliers, the RTBH traffic analysis is conducted on a per event basis. Note that the analysis relies on transport ports because the application payload is not available for privacy reasons. We find that the majority of packets can be assigned to one or two amplification protocols during RTBH activity

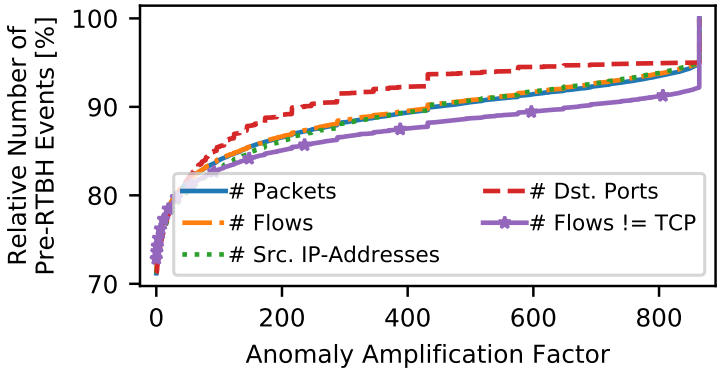


Figure 4.13: Last time slot compared to the mean of the respective pre-RTBH event.

Table 4.3: Different UDP amplification protocols* per RTBH event that shows data and preceding anomaly.

Different protocols* [#]	0	1	2	3	4	5
Events [%]	6	40	45	8.3	0.6	0.1

*Considering the following known amplification protocols/UDP ports:
`QOTD/17`, `CharGEN/19`, `DNS/53`, `TFTP/69`, `NTP/123`, `NetBIOS/138`
`SNMPv2/161`, `LDAP/389`, `RIPv1/520`, `SSDP/1900`, `Game/3659`
`Game/3478`, `SIP/5060`, `BitTorrent/6881`, `Memcache/11211`
Game/27005, Game/28960, Fragmentation/–.

(cf., Table 4.3). The most common amplifying vectors per event are cLDAP, NTP, and DNS, all of them significantly misused for amplification attacks [76]. Note, that we investigate on the filter complexity which is required to blacklist DDoS traffic precisely. That is why we focus on the protocol mix, not the amount of packets or bytes transferred. The overall trends in terms of blackholed packets per protocol have been described in related work [34].

4.5.5 Potentials of Fine-Grained Filtering

Since most events show traffic patterns of well-known attacks, we investigate the impact of fine-grained filtering to prevent collateral damage. For each RTBH event with an anomaly and available traffic, we emulate the filtering of UDP amplification packets. Figure 4.14 shows the relative amount of RTBH events where filtering of a specific ratio of amplification packets is possible. Fortunately, 90% of the RTBH events could be supported completely by dropping common UDP amplification traffic based on an a priori known port list. Such fine-grained filtering would prevent collateral damage in a lightweight fashion. The remaining 10% require

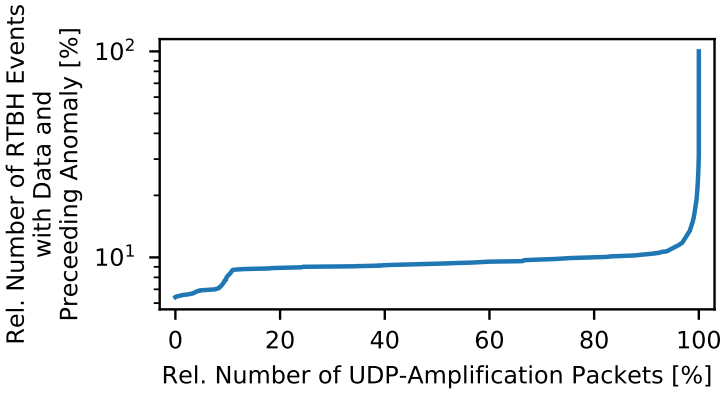


Figure 4.14: Relative amount of dropped packets per event if filtered by known UDP amplification traces.

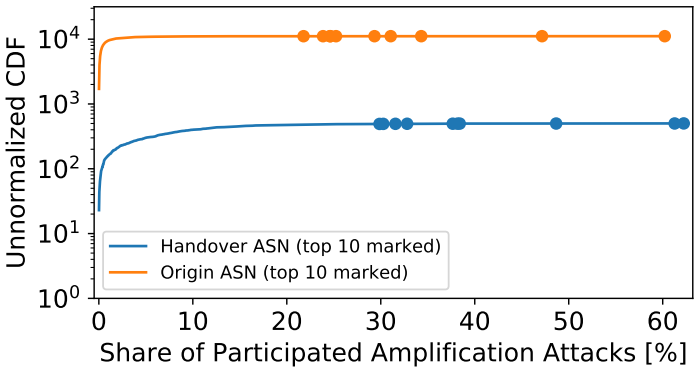


Figure 4.15: Share of UDP amplification attacks in which ASes participated. Top 10 Handover and Origin AS are highlighted.

more investigation and are more difficult to mitigate. We observe attacks on random ports, increasing port numbers, and the use of multiple transport layer protocols.

So far, we did not investigate the sources of attack traffic on an event-basis. Since we observe pre-dominantly UDP reflection attacks, most source-IPs are not spoofed as they are sent from reflectors to victims. This allows us to determine the origin AS of the attack traffic, i.e., the AS hosting the amplifier. Moreover, we are able to determine the handover AS, i.e., the ingress AS at the IXP switch fabric. As this mapping is based on the MAC-address, it is also not susceptible to spoofing.

Figure 4.15 shows the CDF for the share of UDP amplification attacks in which the handover and origin ASes have participated. Overall, we have observed 501 handover AS (55% of all IXP members) and 11124 origin AS (17% of all advertised AS) participating in attacks events. The majority of handover AS do not participate in more than 10% of events and most origin AS

do not participate in more than 3% of events. However, we also find very frequent AS. We highlighted the 10 last discrete steps in the CDF, which mark the top 10 AS in each category. The same AS is the top-ranked origin AS (60% of events) and handover AS (62% of events). Although participating in so many events, this origin AS is only responsible for 6% of the total attack traffic. On average, we observe 1086 amplifiers during an attack and traffic from 30 handover AS or 73 origin AS. Our results indicate a highly distributed usage of amplifiers, which makes fine-grained blacklisting based on the attack source very difficult.

4.6 Investigating Collateral Damage of RTBH

We have analyzed the blackholed traffic without inspecting the legitimate traffic. We will now try to identify legitimate traffic based on reoccurring traffic patterns outside of the RTBH events. Such information could be used to implement whitelists during an attack, and to approximate the collateral damage during RTBH events.

4.6.1 Port Distribution per Host

Since we observed traffic anomalies before RTBH events (see Subsection 4.5.3), we prepend a 10 minutes reaction time to each of these RTBH activities. Traffic during this reaction time is not classified as legitimate. We select hosts (identified by an IP address) with incoming and outgoing traffic on at least 20 different days, which is a conservative lower bound of samples to identify legitimate traffic. Only 30% of blackholed IP addresses meet this criteria. To verify our assumption that servers are a common DDoS victim and hence tend to be blackholed, we first need to distinguish server hosts from client hosts. Therefore, we inspect four features:

1. # of unique source ports in incoming traffic
2. # of unique source ports in outgoing traffic
3. # of unique destination ports in incoming traffic
4. # of unique destination ports in outgoing traffic

We expect the following behavior based on a common client-server scenario. A server should receive traffic on few dedicated listening ports. In contrast, clients use random source ports to initiate communication. The server will thus receive traffic from many different ports and reply to these many ports from its stable ports.

We use a RadViz projection [55] to visualize our results, see Figure 4.16. RadViz visualizes multi-variate data by projecting an N-dimensional data set into a 2D space. Features are represented by anchor points equally spaced around the perimeter of a unit circle. Each data point is attached to all anchors by a spring, the stiffness of which is proportional to the numerical value of that feature. The values are normalized by the maximum number of values each feature

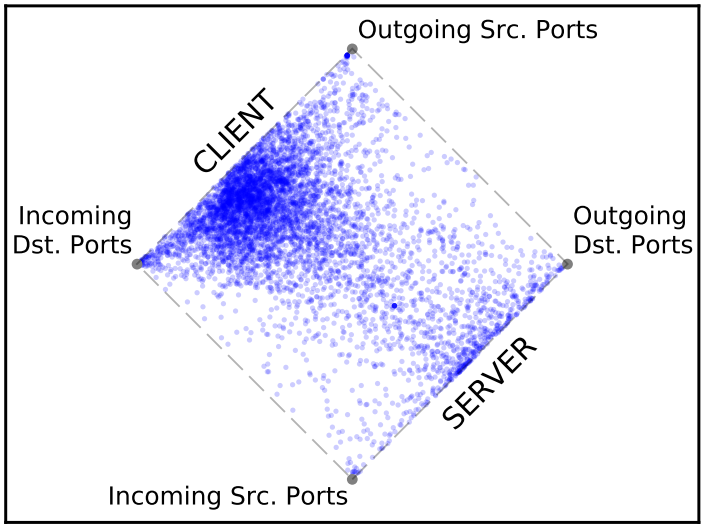


Figure 4.16: Port Distribution of IP addresses outside of pre-RTBH events.

can attain. Data points are closer to the anchors for which they have higher values than for the others.

In our case, each data point represents a host, the features represent the ports, and the normalization factor is derived by the maximum port number (*i.e.,* 1/65535). Client hosts will be pulled by an anchor that represents high diversity in the number of unique destination ports in incoming traffic (or high diversity in the number of source ports in outgoing traffic). On the other side, server hosts that send traffic to clients will be pulled by an anchor that represents high diversity for source ports in incoming traffic (or high diversity for destination ports in outgoing traffic).

We observe more IP addresses that show traffic patterns of clients (see Figure 4.16). To our surprise these nodes are protected by RTBH.

4.6.2 Detecting Stable Traffic Patterns

To get a better understanding of the previous observation, we refine our results by inspecting the incoming traffic in more detail. This analysis is particularly challenging as client traffic is highly variable, which makes the detection of normal traffic patterns difficult.

For each destination IP address, we determine the number of days with incoming traffic and for each day the most utilized destination port, which we call *top port*. Note that we differentiate between protocols, so each port is identified by a protocol-port-tuple, *e.g.,* (TCP, 80). Based on this, we compute the port variation, which is the ratio between the number of top ports and days with traffic. Consequently, a port variation of 1 means that we have observed a different top port on each day. A port variation close to 0 indicates very stable top

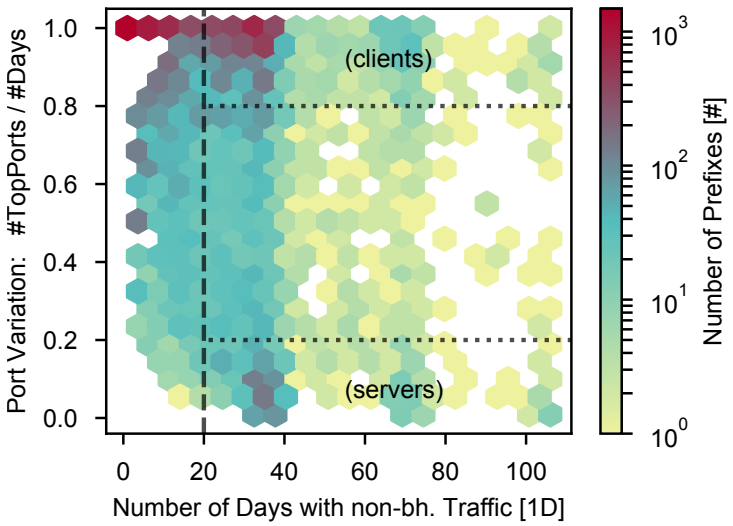


Figure 4.17: Top port variation and classification of IP addresses for traffic outside of RTBH events.

Table 4.4: ASN types for detected server and client IP addresses based on Peering DB.

Type	Clients	Server
# Hosts	4057	1036
Content	2%	**34%**
Cable/DSL/ISP	**60%**	14%
NSP	14%	13%
Enterprise	1%	1%
Unknown	23%	38%

ports, which resembles the behavior of frequently used servers with well-known applications. We show our results in Figure 4.17. We use the port variation to classify hosts as clients or servers. Again, we require at least 20 days of captured packets. We find over 4,000 clients and 1,000 stable servers.

To gain confidence in our results, we map each client and server IP address to its origin AS. Then, we retrieve the AS type from Peering DB [257], see Table 4.4. The most common AS type for clients is "Cable/DSL/ISP" (60%), for servers it is "Content" (34%). This means we found over 2,000 hosts with traffic patterns resembling clients that are actually located in ISP networks and have been targeted by DDoS attacks. DDoS attacks on clients have been reported before [30], [280], [146]. These attacks occur mainly due to disputes in online gaming and to manipulate e-sport matches [280]. Nevertheless, we are surprised how pronounced this shows up in our data set, in particular in comparison to the identified number of traditional servers.

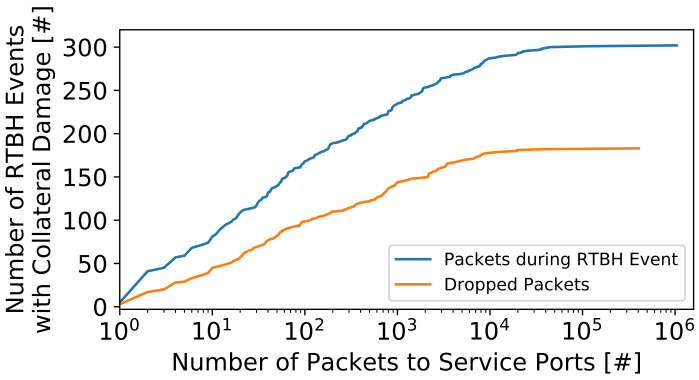


Figure 4.18: Collateral damage during RTBH events for servers. We differentiate by all packets to service ports and actually dropped packets.

4.6.3 Towards Quantifying Collateral Damage

The identification of servers with stable top ports enables us to present a preliminary assessment of the collateral damage[1] during RTBH events. Note, that clients have a different top port for almost every day of activity. This makes a description of legitimate traffic patterns very difficult. In contrast, the detected servers have only a small list of frequently addressed top ports, which indicates legitimate traffic patterns.

For each detected server, we quantify the number of sampled packets sent to the identified top ports during RTBH events. Overall, we find 300 RTBH events with traffic including collateral damage for our 1000 detected servers. The (unnormalized) CDF for the number of packets to top ports is shown in Figure 4.18. We differentiate between all packets sent to top ports during an active RTBH event, *i.e.,* packets that should have been dropped, and those that were actually dropped. We deliberately decided not to quantify collateral damage as a relative traffic share. Expressing collateral damage in percent yields very small shares which only point towards large attack volumes, which are expected during DDoS events. Hence, in order to quantify the collateral damage, we show the absolute values. We observe a collateral damage of up to 10^6 packets. Note that we cannot differentiate between collateral damage and attack traffic sent to top ports, *i.e.,* application specific attacks. Thus this graph shows the upper limit, worst-case, of collateral damage for the detected servers.

[1]Please note that we define packets that were dropped (or should have been dropped) due to a RTBH announcement but represented legitimate traffic as collateral damage. This is why inferring legitimate traffic patterns is a prerequisite for this analysis. Moreover, this is only possible for server deployments since only these exhibit reoccurring, stable traffic patterns during normal operation, *i.e.,* peace times.

Understanding the Challenges of Future Work

Based on our analysis, we identify the following challenges for the assessment of collateral damage. First, we detected servers and clients as victims of DDoS. Since clients have variable usage patterns and might also receive dynamic IP addresses from IP address pools, finding stable patterns for these cases is very difficult.

Second, we see two sources of bias in our traffic captures. (*i*) Incoming traffic is biased by scans. End-hosts might receive traffic on ports although no application is listening on that ports. (*ii*) Outgoing traffic is biased by spoofing. Spoofed packets suggest traffic from ports, which the end-host actually never used.

Third, in most cases we have very sparse data outside of RTBH events. This impedes results that are statistically significant. Packet sampling does not only reduce the number of packets visible, but also the level of information. We only see header-data up to the transport layer without the possibility to interpret application payload for a finer service-detection.

Fourth, attack traffic is also present outside of RTBH. We deal with this challenge by inferring RTBH events. However, not all DDoS attack have to trigger a DDoS mitigation. It remains open, whether the RTBH information we collect from the route-server is sufficient for a reliable traffic classification.

Last, the patterns of legitimate traffic might change during a DDoS attack. For example, legitimate clients will send more Syn-requests to a server which is not responding due to being overloaded. This behavior has been also observed for stateless protocols such as DNS over UDP and was termed friendly-fire [105].

4.7 Discussion of Findings and Operational Practices

In this chapter, we have investigated different perspectives on RTBH from the scientific point of view. The actual usage patterns of RTBH, however, are strongly influenced by practical considerations of network operators. Therefore, we discuss how the findings in this chapter can give insights from a practical point of view.

4.7.1 RTBH Acceptance

Most BGP routers available on the market today support RTBH with small configuration adjustments. The default BGP configuration of virtually all devices, however, does not accept prefixes longer than `/24`, yet—including blackhole announcements.

Specific configuration settings are required to accept longer prefixes for blackholes only. Our investigation into the RTBH acceptance rates by prefix length show that RTBHs with prefix lengths between `/24` and `/32` exhibit especially low dropped packet shares. The most likely reason for that is that some operators specifically enable whitelisting of `/32` prefixes in their routers, but not the prefix lengths between `/25` and `/31`.

More importantly, our investigations also showed that the number of operators that do not accept `/32` blackhole routes is alarming. Surprisingly, this does not only affect small and mid-sized network operators, but some of the largest network operators connected to the IXP. Only 32% of the 100 top traffic source ASes accept host-specific blackhole routes, which are typically used to mitigate DDoS.

The deployment and usage of these incomplete RTBH configurations do not only lead to unpredictable protection against unwanted traffic, but may also shed light on the efforts required to enable RTBH. While low margins and high market pressure explain why many small- and mid-sized operators choose not to invest in these configuration adjustments, the reasons why global network service providers remain unclear. One reason might be that large network service providers use alternative mitigation approaches outside of the IXP ecosystem to handle DDoS attacks.

In any case, missing incentives are likely to play a role in the low acceptance of blackholing routes. The ASes that could gain the most from RTBH are under severe market and cost pressure and often lack the necessary skills to implement blackholing correctly. This can be addressed by additional free advanced training of the IXP community. The ASes that do not see the need to use blackholing, either because they can handle the load inside of their network or because they rely on third-party DDoS protection services, are less willing to invest into infrastructure modification that help other ASes only.

4.7.2 RTBH Collateral Damage Prevention

RTBH is generally a coarse-granular traffic filtering tool. Unfortunately, even the currently available options to reduce the collateral damage triggered by blackholing are not used. Targeted announcements could be used to specifically drop traffic from neighboring ASes that send attack traffic. As we showed, however, the usage of this feature is minimal in the investigated data. Therefore, we conclude that this feature is virtually ignored.

Furthermore, RTBHs could be announced and withdrawn in a timely manner to filter only attack traffic. A significant part of the blackholes, yet, stay active for a very long time (compare the relative RTBH event duration in Figure 4.19). For those announcements, we found almost no indication that they relate to the alternative explanation, prefix squatting. We therefore suspect that many of these blackholes were once manually triggered to prevent a DDoS attack and then have been forgotten.

We therefore conclude that preventing collateral damage caused by RTBH-based DDoS mitigation is not a high priority for users of blackholing today. Rather, RTBH is a simple-to-use tool to prevent DDoS attacks that are threatening the network of an operator. Ensuring appropriate reachability of the victims of the DDoS seems to play a minor role in these considerations. The results of these investigations are rather disillusioning, as we have showed that fine-grained blacklisting of attack traffic based on (transport layer) ports is very effective. In turn, detection

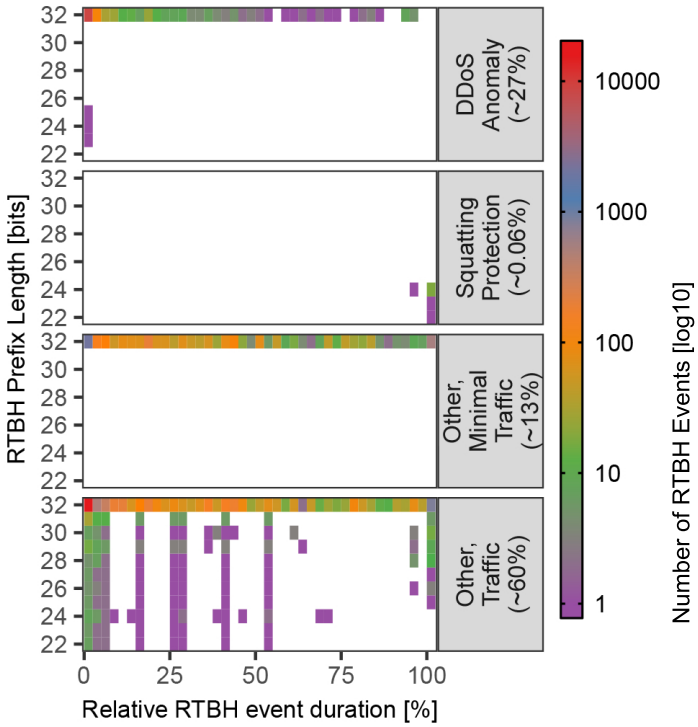


Figure 4.19: Classification of RTBH events according to different use cases.

of legitimate traffic patterns and whitelisting of such patterns during an attack is not possible due to highly variable client traffic.

4.7.3 RTBH Event Classification

We provide an overview of RTBH event classes in Figure 4.19. Note that we use the classes introduced in Table 4.1. In our data set, the major part of RTBH events with DDoS-like anomalies are highly likely to be infrastructure protection RTBHs and represent ≈ 27% of the total events. The potential use of RTBH for prefix squatting protection was found for four ASes and 21 prefixes. The *Other* RTBH events cannot confidently be classified into either use case. We find that for a significant part of these `/32` *other* events, or 13% of the total events, fewer than 10 packets are visible in our data set. Given that some of these prefixes were active through a complete measurement interval, we have to consider that at least a part of these prefixes are not kept intentionally active. Rather, we consider them *RTBH Zombies*, which were once manually triggered but now forgotten. These prefixes pose a risk for their owners, since they are likely to create operational issues for their potential users. For example, connectivity issues of these addresses may be very difficult to debug, since on average, they are only reachable for 50% of the traffic at the IXP.

Finally, 60% of the RTBH events do not match clearly with any common, well-known use-case. These events show constant traffic patterns with no anomalous changes. From the classification perspective, this result is not satisfactory and clearly shows the need for further research to completely understand how and why RTBH is used today.

Our results indicate that either not publicly understood use-cases of RTBH exist or that the IXP is not a self-sufficient vantage point. World-wide AS might announce RTBHs at all point-of-presence although only a small, local DDoS attack takes place. Overall, we do not think that the presented results are an artifact of our methodology. Related work shows similar trends with less then 30% RTBHs being related with DDoS attacks: Jonker et al. [211], [65] uses a distributed approach to link RTBHs with DDoS attacks. They are utilizing data from an Internet telescope, amplification honeypots and public BGP route collectors which allow, as opposed to our central vantage point, a very broad view. The authors hypothesize about missed attacks, as direct and unspoofed attacks are not detectable by them. Although being able to observe these additional attack types, we arrive at the same results.

4.8 Conclusions and Outlook

In this chapter, we took the first deep dive into the use patterns of remotely triggered blackholing (or BGP blackholing) at a large European Internet Exchange Point. We comprehensively analyzed a data set of control plane correlated with data plane measurements that spanned three months. We did not only consider the behavior of autonomous systems that trigger blackholing but also analyzed networks that received blackhole routes. To our surprise we found several disturbing operational practices which—if improved—could increase the reachability on the Internet infrastructure.

Our further analysis revealed intrinsic measurement challenges for answering important questions about the collateral damage introduced by RTBH. Full packet captures are not available because of privacy and performance reasons, in particular at highly popular IXPs. Therefore, our community relies on packet samples. We found that only a relatively small subset of the captured samples can be used to clearly identify the traffic mix before and during DDoS mitigation, and thus to quantify the collateral damage.

In future work, we will extend our methods to cover a larger portion of RTBH-protected DDoS events when quantifying the collateral damage. We also hope that our results illustrate the potentials of RTBH services to the operator community, which may lead to improved Internet infrastructure security in the mid- to long-term.

4.9 Ethical Considerations

The *control plane* information on remotely triggered blackholes is publicly available at multiple vantage points on the Internet and does not contain potentially privacy-affecting information.

The *sampled flow data* contains data from the network layer and the transport layer. This data potentially contains information that could be correlated or connected to individuals and therefore bears a potential privacy risk. Therefore, the collection and handling of flow data is conducted strictly in accordance to the privacy laws applicable to the collecting organization. The handling of potentially privacy-relevant data is strictly confined to dedicated computer systems that are isolated from the Internet. This data never leaves the premises and control of the collecting organization. Privacy-relevant data is aggregated and anonymized as early as possible in the analysis process. None of the results discussed in this work can be traced to individual IP addresses or other, privacy-related information.

Chapter 5

Quantifying QUIC Reconnaissance Scans and DoS Flooding Events

Abstract

In this chapter, we present first measurements of Internet background radiation originating from the emerging transport protocol QUIC. Our analysis is based on the UCSD network telescope, correlated with active measurements. We find that research projects dominate the QUIC scanning ecosystem but also discover traffic from non-benign sources. We argue that although QUIC has been carefully designed to restrict reflective amplification attacks, the QUIC handshake is prone to resource exhaustion attacks, similar to TCP SYN floods. We confirm this conjecture by showing how this attack vector is already exploited in multi-vector attacks: On average, the Internet is exposed to four QUIC floods per hour and half of these attacks occur concurrently with other common attack types such as TCP/ICMP floods.

5.1 Introduction

QUIC is a secure transport protocol originally developed by Google and tested in Chrome browsers since 2013 [83]. It has been recently standardized by the IETF as RFC 9000 [203] and at the same time enjoys rapidly growing deployment by major Web operators and browsers. In 2017, Google estimated that QUIC accounted for 7% of Internet traffic [83] and, by the end of 2020, Facebook announced that 75% of its Internet traffic is QUIC [186]. Despite its recent standardization, QUIC has already many implementations [121] and concurrently supported QUIC versions [135]. In 2021, scans of the complete IPv4 address space detected around 2 million QUIC servers [135].

Key design objectives in QUIC were privacy and security. QUIC was built to reduce the attack surface on the transport layer, which includes attacks such as reflective amplifications [266] and resource exhaustions. Security considerations in the QUIC RFC [203] span 18 pages and discuss properties against active and passive attackers.

In this chapter, we report about early observations that indicate regularly ongoing attacks based on QUIC. We argue that the strong security model in QUIC does not preclude misuse

and measure clear signals of DDoS attacks in Internet background radiation. We confirm these results by correlating our observations with several complementary data sources. Our findings indicate that QUIC servers are indeed prone to resource exhaustion attacks and these flaws are currently exploited in multi-vector attacks. We believe that it will be crucial to monitor such attack attempts early in the QUIC deployment phase before they enfold their full potential.

The main contributions of this chapter summarize as follows:

1. We present the first study on QUIC Internet background radiation as seen by a large network telescope.
2. We show a significant bias by research scanners but also detect scanning activity from non-benign sources.
3. Surprisingly, we find high-volume backscatter events suggesting that QUIC is used in multi-vector resource exhaustion attacks, targeting well-known companies.
4. We benchmark a popular web server implementation to test its DoS resiliency and validate our observations.
5. We show the efficacy of QUIC's built in defense mechanism with `RETRY` messages, which remains unused in the wild based on our measurements.

In the remainder of this chapter, we present background and related work in Section 5.2. We outline the QUIC attack scenarios that we base on in Section 5.3, and introduce our measuement method and data sources in Section 5.4. We analyse QUIC scanning and backscatter events in Section 5.5. Finally, we discuss our findings in Section 5.8.

5.2 Background and Related Work

One of the major design goals in QUIC is the decrease of latencies for client-server applications by reducing round-trip times (RTTs) due to multiple, independent handshakes [29]. To this end, sequential handshakes from TCP, TLS, and HTTP have been merged into a single, comprehensive handshake process. Furthermore, to overcome delay because of TCP head-of-line blocking, QUIC is based on UDP [83].

QUIC handshakes. QUIC utilizes various handshake procedures to set up and resume connections. In the best case, when cached information of a prior connection is available to the client, encrypted application data can be sent immediately leading to a 0-RTT handshake. In the worst case, the handshake requires 3 RTTs [135]: If the client offers unsupported QUIC versions, the server first enforces a version negotiation and then proceeds with a *typical* [46] handshake.

A typical handshake is depicted in Figure 5.1. First, the client sends an `Initial` message including a `TLS Client Hello`, which is answered with an `Initial` message including a `TLS`

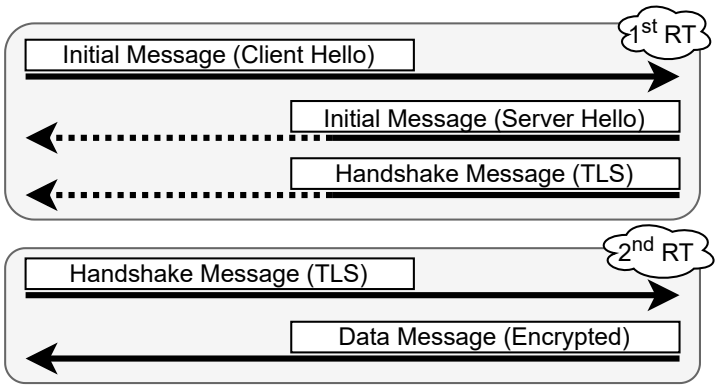


Figure 5.1: QUIC clients receive data with the second round-trip (RT). At the first RT, servers respond to unverified client IP addresses, which can be misused.

`Server Hello`. This message is immediately followed by a `handshake` message containing the rest of the TLS server information (*e.g.,* certificates). The setup is complete with a `Handshake` message from the client and is usually accompanied by a data request. Since the client is allowed to send data requests after the first round-trip, this handshake is described as a 1-RTT handshake. During the handshake, the server and client agree on a (*i*) source connection identifier (SCID) and a (*ii*) destination connection identifier (DCID), respectively, which can be used to identify the QUIC connection independently of the traditionally used connection 5-tuple.

`RETRY` to mitigate resource exhaustion. QUIC traffic is almost entirely encrypted with TLS 1.3 to prevent the ossification that middleboxes cause on protocols like TCP [58]. During the first RTT, a server responds to an unverified client IP address. This means that the server performs cryptographic operations for a potentially spoofed client. QUIC supports `RETRY` messages [203] to limit the attack surface of resource exhaustion attacks. `RETRYs` precede a typical QUIC handshake and force the client to respond with a unique token, which proves its authenticity. This mitigation, however, adds a complete RTT, which conflicts with QUIC original design goals. Recent QUIC server implementations such as NGINX or Picoquic support `RETRY` [200] but based on the backscatter we observe `RETRY` seems rarely deployed.

Related work. Prior work focusses on three aspects. First, the performance benefits of QUIC, especiially in low bandwidth, high latency, and high loss [17], [21], [28], [67] or multi-hop [27] scenarios. Second, the adoption of QUIC [94], [135], [144], [149]. Third, protocol security. Previous results [91] suggest that QUIC's security weaknesses, such as insufficient forward secrecy or susceptibility to replay attacks, are introduced by the mechanisms used to reduce latency. To the best of our knowledge, this is the first work that analyzes QUIC background radiation and reveals QUIC DoS traffic, showing that we still face a trade-off between small latencies and robust security guarantees in the wild.

5.3 QUIC Attack Scenarios

In this section, we briefly discuss those attacks that are most relevant in the context of this chapter. An attacker has two common options [99] to exploit weaknesses in QUIC [176], [252]. First, an attacker could initiate state-overflow attacks to harm QUIC servers. Second, an attacker could trigger reflective amplification attacks to harm the network.

State-overflow attacks. To launch a state-overflow attack, an attacker would act as a QUIC client and induce connection states at a QUIC server, by initiating full handshakes. Any QUIC server answers to the connecting client with a unique Source Connection ID (SCIDs) and its TLS certificate. This part introduces cryptographic load on the server and forces the allocation of resources to maintain states. It is noteworthy that during the first round-trip (RT) of the full handshake the client is still unverified, which limits protection mechanisms in QUIC. To artificially increase the number of states, an attacker would randomly spoof the source IP addresses, source ports, or SCIDs. Floods benefit from spoofed IP addresses in particular as the backtracking of spoofed traffic is challenging [184], [45], [85], [108].

This attack is very similar to TCP SYN floods [183], which exploit the fact that the network stack needs to maintain all currently active connections. Spoofed connections fill up the connection queues at the victim and cause the rejection of legitimate TCP requests.

A large content network has started to mitigate QUIC floods [176] but the extent of this attack in the wild has not been studied yet.

Reflective amplification attacks. Since QUIC is based on UDP an attacker could easily launch a reflective amplification attack. An attacker would send an `Initial` packet including a spoofed source IP address to a QUIC server. The server then replies with an `Initial` QUIC message and a TLS handshake. The TLS handshake is larger compared to the client's request, since it includes the server certificates. As long as the client is unverified (*i.e.,* the client did not yet sent messages that include information supplied by the server, see the QUIC RFC [203]) QUIC servers are only allowed to triple the bytes of a request in their response, though.

To increase the absolute number of bytes sent to the (spoofed) victim, an attacker would add padding bytes to the `Initial` message. Such strategy would not be suspicious. Sending large initial handshake packets is suggested in order to allow the server to accommodate full certificates in a single message and thus reducing delays [255].

Why amplification attacks are unlikely. QUIC has been designed based on experiences with amplification attacks. Thus, it limits the size of replies to unverified clients to a factor of `3`×. The major reason why QUIC is unlikely to be used for amplification attacks is the wide presence of other protocols that support much higher amplification factors (*e.g.,* NTP `500`× and DNS `60`× [133]). Given TLS certificate compression [192], only 1%-9% of the server replies do not fit in a single message, depending on the initial client message size [255]. Furthermore, attackers tend to reuse their existing attack infrastructure and adapt very slowly to new

protocols to conduct reflective amplification attacks [70]. As QUIC amplification attacks are currently unlikely, we focus on state-overflow attacks.

5.4 Measurement Method and Setup

We analyze QUIC background radiation, and in particular backscatter traffic, *i.e.,* responses to spoofed packets.

5.4.1 Method

Our vantage point is a network telescope, since network telescopes passively collect unsolicited traffic known as Internet background radiation (IBR). IBR consists of traffic resulting from research and malicious scans [103], [128], misconfigurations and bugs but also from responses sent by the victims of randomly spoofed, state-building attacks. Network telescopes have been used reliably to quantify DoS victims for more than 15 years [64], [102].

We identify QUIC traffic based on transport layer properties by selecting all UDP packets with a source or destination port `UDP/443`. This port-based classification has been proven sufficient in prior work [135]. To exclude false positives, we extend this common classification method by utilizing Wireshark payload dissectors, which also have been shown to be efficient for classification [115]. We verified QUIC detection by manually checking QUIC connections to well-known QUIC servers. All packets were correctly identified and dissected by Wireshark.

To group packets sent to the telescope into backscatter and scanning events, we mark all QUIC packets with source port `UDP/443` as responses (*i.e.,* backscatter) and all packets with destination port `UDP/443` as requests (*i.e.,* scans). These two sets are disjoint, as we do not find any packet with destination and source port set to `UDP/443`.

To find DoS events, we apply the notion of traffic sessions and DoS thresholds as defined by Moore *et al.* [102]. We acknowledge that these thresholds have been defined with the help of older traffic patterns, however, they are still used in recent work [64]. Along the line of our analysis we will learn that these thresholds (*e.g.,* the *timeout* parameter) are still appropriate for current traffic patterns. For details on the impact of the threshold configurations, see Section 5.6.

5.4.2 Data Sources

We utilize the UCSD Network Telescope [172] to observe both QUIC IBR and common TCP/ICMP IBR. This telescope is operated by the University of California San Diego and represents a `/9` network prefix., *i.e.,* 1/512 of the available IPv4 address space. We are thus able to capture at least 2 ‰ of any horizontal scan or randomly spoofed attack. In this work, we focus on one month, April 1-30, 2021. Overall, we find 92 million QUIC packets during our measurement period.

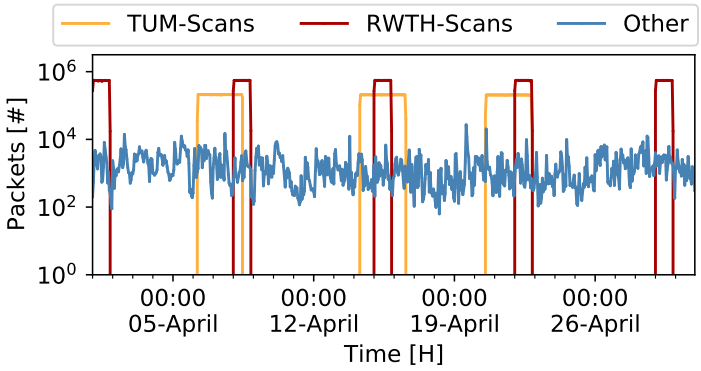


Figure 5.2: QUIC traffic seen at the UCSD network telescope. In the remaining analyses, we identify and remove the extreme bias of research scanners.

In order to bolster our results and put them into context, we correlate our observations with several complementary data sources. We reuse data from active scans that explore QUIC servers [135], correlate IP addresses with the GreyNoise Honeypot Platform, and use meta data from PeeringDB. The active scans provide a set of potential victims, GreyNoise helps to assess multi-vector attacks based on advanced threat intelligence, and PeeringDB provides additional network descriptions. We mention that all data sets are in sync, spanning the same period of time.

5.5 Analysis

We now analyze QUIC IBR traffic. First, we show an overview of all IBR traffic and then focus on high-volume backscatter events.

5.5.1 Overview of QUIC IBR Traffic

QUIC IBR is dominated by research scanners. We first inspect the total packet count. Since multiple QUIC messages can be embedded in a single IP/UDP packet, we emphasize that we count packets and not individual QUIC messages. Overall, we observe 92 million QUIC packets in April 1-30, 2021. This data set is dominated by periodic scans that target the complete IPv4 address space. Each Internet-wide, single-packet scan sends $2^{23} \sim 8 \times 10^6$ packets to the telescope. In total, 98.5% of QUIC packets are generated by research projects from two universities, Technische Universität München (TUM) and RWTH Aachen Universität (RWTH). We show the amount of packets per hour and compare research scanners with other traffic sources in Figure 5.2. Since these scanners clearly perform regular scans of the entire address

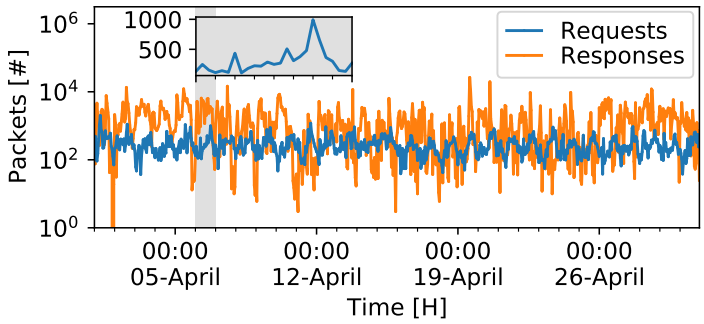


Figure 5.3: Number of QUIC packets by type. Requests exhibit rather stable, diurnal activities with peaks at 6:00am and 6:00pm UTC (see insert for representative day). Responses are very erratic, hinting at flood events.

space, we expect this bias also at other vantage points. We remove traffic from research scanners in the subsequent analyses to focus on the *other* traffic.

QUIC requests follow diurnal patterns, responses are erratic. In the sanitized traffic, we find 15% QUIC requests and 85% QUIC responses. Figure 5.3 shows the number of requests and responses per hour. Requests follow stable, diurnal patterns with peaks at 6:00am and 6:00pm UTC. We demonstrate this with the inlet for a representative day, April 06, 2021. Response traffic is very erratic, exhibiting high peaks and drops per event. This behavior might hint at DoS events [11], [18], which we will inspect in more detail in Subsection 5.5.2.

We can reuse established thresholds for QUIC sessions. In order to move from a packet-based to an event-based perspective, we now group singular packets into sessions. To this end, we aggregate packets using the source IP address and a `timeout` threshold, *i.e.,* packets from a specific source belong to a single session as long as the inactivity period between them is no longer than the `timeout`.

Figure 5.4 exhibits the number of detected sessions given a timeout value between 1 and 60 minutes. The lower bound of this plot is defined by `timeout`=∞, which groups all packets of a source into a single event. We see a significant reduction of sessions until ~5 minutes, which is why we choose this knee point as our threshold. This timeout is coherent with prior work that applied timeouts to IBR traffic [64], [102].

5.5.2 QUIC DoS Traffic

Request sessions are non-benign and originate from eyeball networks. Response sessions are DoS backscatter from content providers. We now inspect and contextualize the observed sessions. Overall, we find 18k sessions containing only requests, 26k sessions containing only responses. We do not observe sessions with both packet types. On average, a request session consists of 11 packets and response sessions of 44 packets.

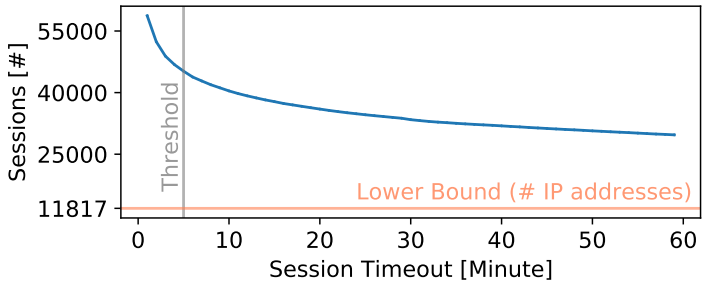


Figure 5.4: Influence of the timeout parameter on the number of sessions. We select 5 minutes as the final threshold to group correlated packets into sessions.

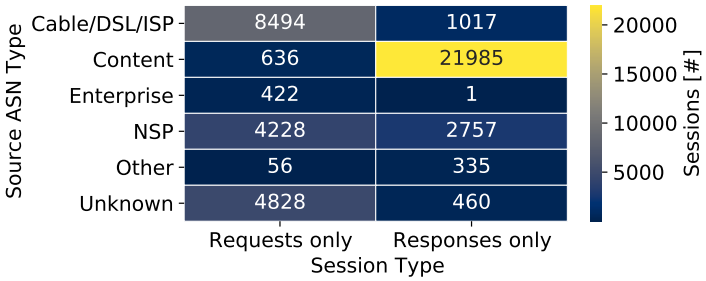


Figure 5.5: Source network types of sessions. Requests originate predominantly from eyeballs. Responses are received almost exclusively from content networks.

For each session, we map the source autonomous system number (ASN) and the AS network type using PeeringDB. We find that request sessions originate from eyeball networks and that response sessions are received from content provider networks, see Figure 5.5. This fits into the assumption that we (*i*) receive malicious scans from bots hosted in eyeball networks (*e.g.,* Mirai [8], [167]) and (*ii*) receive DoS backscatter from legitimate QUIC servers.

Taking a closer look at response sessions we find traffic that exhibits common DoS patterns, using, again, session timeouts and thresholds defined by Moore *et al.* [102]. To identify attacks, we select backscatter sessions with (*i*) more than 25 packets, (*ii*) a duration longer than 60 seconds, and (*iii*) a maximum packet rate of higher than 0.5pps, which is calculated over all 1-minutes slots of the respective event. Finally, we find 2905 attacks which correspond to 11% of all response sessions. Attacks target a total of 394 unique victims, with more than half beeing attacked only once, compare Figure 5.6. By correlating the victims with data obtained from active scans [135], we find that 98% of attacks target well-known QUIC servers. We take a closer look at the low-volume backscatter sessions and our threshold configuration in Section 5.6.

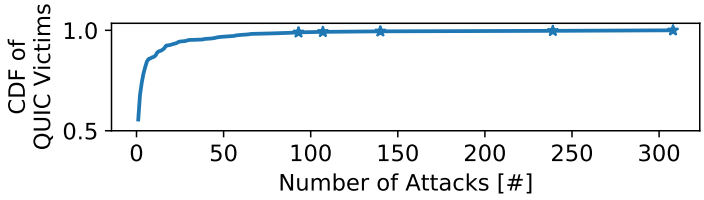
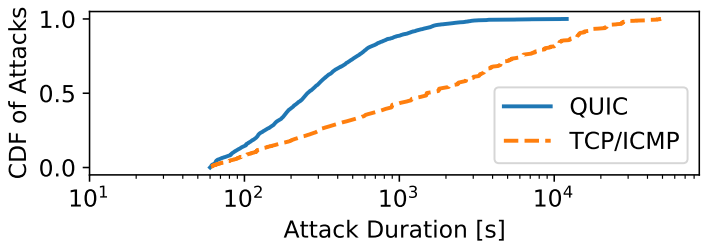
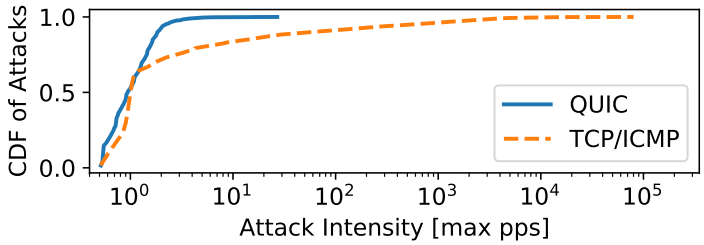


Figure 5.6: CDF for number of attacks per QUIC flood victim. More than half of the victims are only attacked once during our measurement period. Last 5 data points are highlighted.



(a) Flood Durations



(b) Flood Intensities

Figure 5.7: CDF of flood durations and intensities, comparing QUIC and TCP/ICMP. QUIC floods are shorter but the median intensity of QUIC floods is as severe as for common backscatter events.

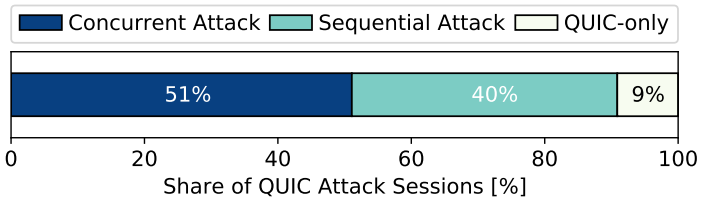


Figure 5.8: Multi-vector attacks: Half of the QUIC attacks occur concurrently with TCP/ICMP floods.

To bolster our observations we correlate the sources of request sessions with data from an reactive vantage point, the GreyNoise honeypot platform. Using the GreyNoise classification, we do not find any signs of benign scanners and 2.3% of the sources are tagged as known bruteforcers or part of a botnet such as Mirai or Eternalblue. Most request sessions originate from Bangladesh (34%), USA (27%), and Algeria (8%).

QUIC floods are shorter but on average as severe as common TCP/ICMP floods. We now compare QUIC DoS floods with floods for common protocols, *i.e.,* TCP and ICMP. The duration and intensity of attacks has been shown by Jonker *et al.* [64]. To allow for comparison, we reproduce the analysis based on our current setup. Overall, we find 282k attacks for common protocols.

Subfigure 5.7(a) shows the distribution of attack durations. QUIC floods are shorter than TCP/ICMP DoS attacks. The median QUIC flood lasts 255 seconds, for TCP/ICMP protocols we observe 1499 seconds. The median attack intensity, however, is close to 1 maximum packet per second (max pps) in both cases, see Subfigure 5.7(b). To estimate the traffic rate from the global Internet towards a victim, we may assume 512 × max pps since the UCSD telescope covers 1/512 of the total IPv4 address space.

QUIC floods are part of multi-vector attacks and highly correlated with TCP/ICMP floods. So far, we looked at QUIC floods in isolation. We now check whether they are part of a larger multi-vector attack towards a single victim. To our surprise, 51% of QUIC floods overlap in time with common (TCP/ICMP) DoS floods (see *concurrent* attacks in Figure 5.8). We require attacks to overlap for at least one second to label them as *concurrent*. Another 40% of QUIC floods target a victim in *sequence*, *i.e.,* the victim was also attacked by a TCP or ICMP flood during our measurement period but at a different time. In such cases, the gap between a QUIC and the nearest TCP/ICMP attack is 36 hours on average. More details about concurrent and sequential attacks, including an example, are presented in Section 5.7. Only 9% of QUIC attacks do not relate to any TCP/ICMP event.

We argue that the reasons for multi-vector attacks are twofold. First, multi-vector attacks are harder to detect as they keep the traffic volumes for each attack vector low. Second, multi-vector attacks are more difficult to mitigate since they require more complex filter rules.

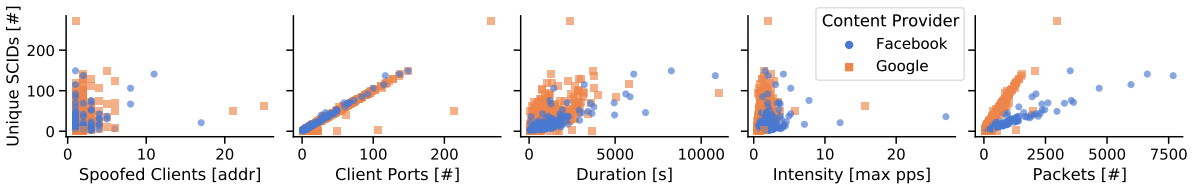


Figure 5.9: >83% of attacks target two content providers. QUIC floods utilize multiple client addresses and ports. Despite a lower packet count, Google reacts with more SCIDs, indicating higher server load.

Well-known content providers are under attack. We find that 58% of attacks target Google and 25% target Facebook. Figure 5.9 compares attack properties for these two content providers to showcase potential differences in attack patterns across content providers and their deployed QUIC variants. We consider SCIDs in the backscatter traffic. The SCID is a QUIC-specific feature, which may serve to assess allocated resources because a context is reserved for each unique connection. We do not show the number of DCIDs in this figure, since they are not available in our backscatter traffic and they are not required to route to the correct endpoint [203]. We carefully checked that the packets are valid, though, by verifying that the DCID length attribute is set to zero.

Overall, the number of spoofed client IP addresses is relatively low. The randomization of client ports, however, is the driving factor for new SCIDs at the attacked server. Despite a lower packet count, Google reacts with more SCIDs, which indicates a higher server load. We observe QUIC variant `mvfst-draft-27` (95%) for Facebook attacks, and `draft-29` (78%) for Google.

These results suggest that operators may protect against QUIC floods by filtering based on common transport protocol features (*i.e.,* ports) instead of using QUIC-specific features (*i.e.,* SCIDS), which eases the deployment of countermeasures.

5.6 Non-Attack Backscatter and Threshold Configuration

We reused thresholds defined by Moore *et al.* [102] to infer DoS events. We classified 11% of response sessions as attacks. Although this seems like an extreme reduction of events, we argue that underestimation is better than overestimation, because this prevents false positives. Nevertheless, we also checked the excluded response sessions for deviating trends. We do not find any anomalies. Excluded events have a median intensity of 0.18 max pps, a duration of 7 seconds, and consist of 11 packets. Such low-volume events point to misconfigurations and are most likely insignificant for our DoS analysis, hence, should be excluded.

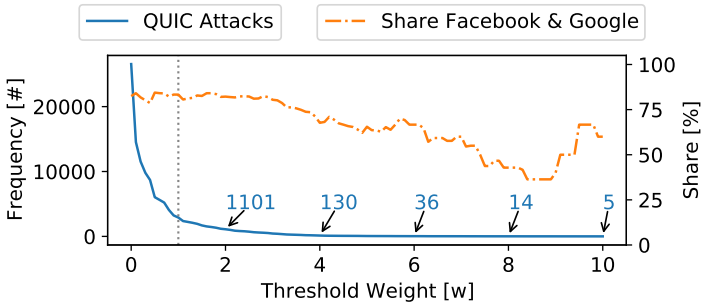


Figure 5.10: Varying the DoS threshold defined by Moore *et al.* [102] to show the impact on the number of detected attacks and the relative share of affected content infrastructures. Even for a very strict threshold configuration of $w = 10$, we find QUIC attacks.

To further understand the effects of our threshold configuration, we introduce the threshold weight w. We multiply each threshold by w, which leads to a more relaxed ($w < 1$) or stricter ($w > 1$) attack detection. If $w = 1$, the default threshold configuration as defined by Moore *et al.* [102] is used.

Figure 5.10 shows the number of detected attacks. We exclude many low-volume backscatter events for $w \leq 0.3$, but even for an extreme configuration of $w = 10$ we still classify five backscatter sessions as attacks. On the secondary y-axis, we show that the share of well-known content providers remains high independently of w. These results bolster our main insight that QUIC `Initial` floods are used to attack large content providers.

5.7 Details about Attacks

We introduced concurrent (*i.e.,* multi-vector) and sequential attacks in Subsection 5.5.2. In this section, we illustrate those attacks based on a concrete example and present more details about the time overlap between concurrent QUIC and TCP/ICMP attacks, as well as the time gap between sequential attacks.

5.7.1 Illustration of Multi-vector versus Sequential Attacks

In Figure 5.11, we illustrate concurrent and sequential attacks based on a snapshot for a single victim. First, the victim is attacked by one QUIC and one TCP/ICMP attack that take place at the same time (or *concurrently*). Please note that these two attacks have an almost perfect overlap but we classify any two attacks as concurrent if the respective time ranges overlap in at least a single time unit, *i.e.,* they share at least one mutual second. Such a perfect overlap is very likely, compare Subsection 5.7.2. We then observe five QUIC attacks in *sequence* to the first TCP/ICMP attack.

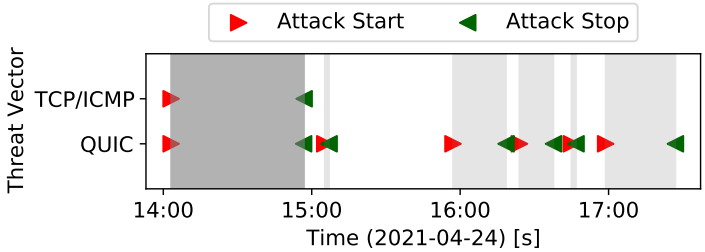


Figure 5.11: Attacks towards a single victim. We observe one concurrent usage of attack vectors, *i.e.*, a multi-vector attack, followed by five sequential QUIC floods.

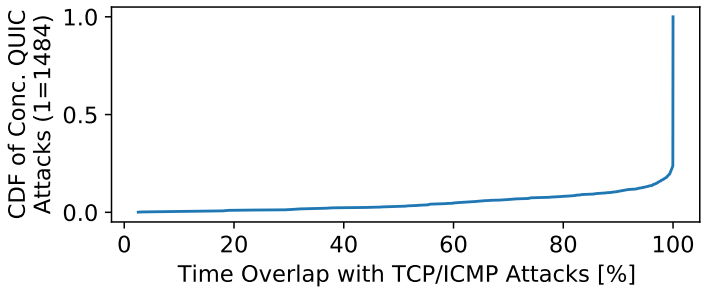


Figure 5.12: Attack overlap of multi-vector attacks. Most concurrent QUIC attacks almost completely overlap with attacks that use common protocols.

5.7.2 Overlap of Concurrent Attacks

We now investigate how QUIC attacks overlap with common TCP or ICMP attacks. To this end, we calculate the share of overlapping seconds for each QUIC attack that is part of a concurrent attack. Figure 5.12 shows the distribution. We find a high correlation between QUIC and TCP/ICMP attacks. Three quarters of all concurrent QUIC attacks occur completely in parallel to an TCP/ICMP attack (100% in the CDF). On average, multi-vector QUIC attacks share 95% of the attack time with common attacks.

5.7.3 Time Gaps Between Sequential QUIC Attacks and TCP/ICMP Attacks

We label an attack session *sequential attack* when we observe QUIC and TCP/ICMP attack traffic to same victim but QUIC and TCP/ICMP attacks do not overlap. Figure 5.13 exhibits the distribution of time gaps between both attack vectors. There is a break of more than one hour for 82% of the sequential attacks. In some cases, a break may take up to 28 days. These long time gaps suggest that sequential attacks are indeed not part of a multi-vector attack.

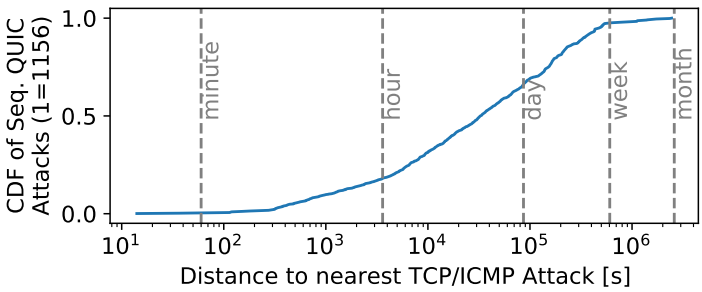


Figure 5.13: Distribution of time gaps between the end (or start) of a sequential QUIC attack and the start (or end) of a TCP/ICMP attack.

5.8 Discussion, Conclusion, and Outlook

Attack patterns are valid. Captured QUIC events that are suspect to DoS consist of 31% `Initial` and 57% `Handshake` messages on average. The `Initial` messages we observe do not contain an (unencrypted) TLS `Client Hello` and thus can be attributed to (encrypted) `Server Hello` replies. Hence, these packets match the backscatter pattern of a QUIC attack (see Section 5.2). QUIC sends multiple UDP packets in response to the `Initial` packet: The first packet contains one `Initial` QUIC packet carrying the `Server Hello` and one encrypted `Handshake` message followed by a second datagram with a single `Handshake` message. The ratio in these packets roughly matches the ratio of one third `Initial` packets and two thirds `Handshake` messages.

`RETRY` attack mitigation is not deployed. We did not capture any `RETRY` messages. The absence of `RETRY` indicates the lack of deployment of a defense mechanism. To validate this observation, we select the ten most frequently attacked servers from Google and Facebook. When actively connecting to these servers with a QUIC client we also do not receive any `RETRY` messages. This supports the observations that we made based on data in the telescope.

Although the QUIC implementations of Google and Facebook support `RETRY` messages (Google since mid 2019 [193], Facebooks since the end of 2020 [188]), it looks like these content providers deliberately decide to not utilize this feature. This decision is potentially due to the performance penalty of `RETRY` messages. However, for frequently utilized services as in the case of large content providers, this penalty could be alleviated by the session resumption feature in QUIC. Also, `RETRYs` could be deployed adaptively and only used when high load occurs.

QUIC servers quickly experience DoS. To check whether the observed QUIC floods can be attributed to DoS attacks we experimentally analyzed the impact of QUIC handshakes on a common web server implementation. For this, we set up NGINX on a modern 128-core machine with 512 GB of RAM, which connects to a client via Gigabit Ethernet. NGINX supports the QUIC RFC [203] and eBPF optimizations [185]. We record 500,000 packets

using the QUIC client `quiche`, version 0.9.0, Cloudflare's QUIC reference implementation. To simulate attacks, we then replay *only* client `Initial` messages at varying packet rates towards new server instances. Replaying avoids bias from hand-crafting QUIC packets.

Our results are summarized in Table 5.1 alongside the configuration (we use 1024 client connections per worker which is twice the default). Since each request elicits four datagrams in response (two datagrams with `Initial` and `Handshake` packets plus two keep-alive `PING` packets after a short delay) we expect four times as many server responses. To determine how many requests were answered we match the respective DCIDs and SCIDs and calculate the service availability ratio.

With four worker processes even small packet rates can lead to significantly reduced service availability. At 1,000 probes per second (pps) only 7% of our requests are answered. In auto-mode, NGINX deploys 128 workers, but larger attack volumes of 10,000 pps continue to impact its availability. Note that these attacks do not impair the general availability of the machine but focus on the service. Extrapolating our observed 27 pps from a `/9` IP prefix to the size of the Internet leads us to believe that attacks of more than 10,000 pps ($27 \cdot 512$=13,824) are ongoing. In our benchmarks, `RETRY` packets successfully mitigate these attacks but they come at the high cost of an additional round trip.

The vulnerability based on the QUIC stateful handshake is not specific to an implementation, but relates to the protocol design. We expect all implementations to be prone to this attack type, the exact pps rates might vary, though. A recent benchmark of picoquic [200] also observed DoS at around 10,000 pps but a successful attack mitigation with `RETRYs`. Latest interoperability tests [271] show that the majority of QUIC server and client implementations correctly support the `RETRY` option. This leaves the decision on robustness versus speed up to the service providers.

Attack duration and intensity. We found that QUIC floods are shorter but the median max packet rate is similar compared to TCP and ICMP. The max packet rate is an indicator of the attack intensity but it also reflects the capability of a victim to sustain under load—for well-provisioned victims we likely observe higher rates as those victims are still able to send data. This also applies to the observed durations. Backscatter events can stop for various reasons: (*i*) the attack has ended, (*ii*) a mitigation was initiated, or (*iii*) the attacked service is completely unresponsive. Hence, shorter durations might indicate that QUIC attacks lead to a faster resource exhaustion compared to common protocols. Analyzing this, *e.g.,* by using reactive scans or correlating with other data, will be part of our future work.

5.9 Artifacts

We support reproducible research. All artifacts of this chapter are available on https://doi.org/10.5281/zenodo.5504168.

Table 5.1: Tests on a local NGINX instance show that the backscatter volume we observed can significantly impact the responsiveness of the web server.

Attack	NGINX Config		Results			
Volume [pps]	QUIC Retry	Workers [#]	Client [# Req]	Server [# Resp]	Service Available	Extra RTT
10	✘	4	3,001	12,004	100%	✘
100	✘	4	30,001	81,680	68%	✘
1,000	✘	4	300,001	81,680	7%	✘
1,000	✘	auto=128	300,001	1,200,004	100%	✘
10,000	✘	auto=128	500,000	522,752	26%	✘
100,000	✘	auto=128	498,991	322,158	26%	✘
1,000	✓	4	300,001	300,001	100%	✓
10,000	✓	4	500,000	500,000	100%	✓
100,000	✓	4	500,000	500,000	100%	✓

Part III

Opportunities for Attack Prevention

Chapter 6

Industrial Control Protocols in the Internet Core: Dismantling Operational Practices

Abstract

Industrial control systems (ICS) are managed remotely with the help of dedicated protocols that were originally designed to work in walled gardens. Many of these protocols have been adapted to Internet transport and support wide-area communication. ICS now exchange insecure traffic on an inter-domain level, putting at risk not only common critical infrastructure, but also the Internet ecosystem (*e.g.,* by DRDoS attacks).

In this chapter, we measure and analyse inter-domain ICS traffic at two central Internet vantage points, an IXP and an ISP. These traffic observations are correlated with data from honeypots and Internet-wide scans to separate industrial from non-industrial ICS traffic. We uncover mainly *unprotected* inter-domain ICS traffic and provide an in-depth view on Internet-wide ICS communication. Our results can be used (*i*) to create precise filters for potentially harmful non-industrial ICS traffic, and (*ii*) to detect ICS sending unprotected inter-domain ICS traffic, being vulnerable to eavesdropping and traffic manipulation attacks. Additionally, we survey recent security extensions of ICS protocols, of which we find very little deployment. We estimate an upper bound of the deployment status for ICS security protocols in the Internet core.

6.1 Introduction

Industrial control systems (ICS) are used to monitor and control industrial environments. Deployments can range from a few controllers in a factory to large distributed systems that monitor critical infrastructures. The underlying ICS communication is based on specialized, often proprietary protocols.

Originally, ICS protocols were designed to operate in closed environments, which do not require authentication and encryption. The lack of security features in ICS protocols remained largely unnoticed due to the deployment in isolated (trusted) environments. This changed

recently when ICS protocols have been stacked onto IP, enabling the management of ICS controllers via the global Internet. Such communication requires protective measures, either via secure tunnels between trusted domains or end-to-end authentication and encryption. Visible (unencrypted) ICS traffic is particularly dangerous since it is prone to eavesdropping and manipulation attacks. Traffic traces also hint attackers to potentially open ICS services without the need to perform suspicious scans. Figure 6.1 sketches encrypted and visible traffic flows between ICS. It also shows a passive vantage point and an active scanner, which might be blocked by a firewall. Note that the firewall does not help in the case of man-in-the-middle manipulation attacks.

In this chapter, we provide the first comprehensive analysis of the visibility of *unprotected* ICS traffic across network domains. In contrast to related work [170], [98] which reveals reachable ICS services, we explore the communication of the whole ICS ecosystem, from the ICS controllers to the management stations. We show that ICS systems are controlled remotely without any protective mechanisms, harming both the Internet as well as the industrial infrastructure. Our results attract attention to the insecure usage of ICS protocols and motivate secure ICS deployments based on amendments such as DTLS and encrypted tunnels. To explore the deployment of encrypted ICS traffic, we extend our previous work [114] and provide methods and analysis that reveal limited secure ICS protocols.

In detail, our contributions are the following.

1. We present the first analysis of inter-domain ICS traffic at two central Internet vantage points, an Internet Exchange Point and an Internet Service Provider, covering 6 months.
2. We find new unprotected ICS deployments which are undetected by recent scan projects.
3. We classify industrial and non-industrial ICS traffic based on cross-correlations with other data sources such as honeypots.
4. We assess common tools for implementing our proposed methodology to allow for future long-term monitoring and mitigation.
5. We survey recent security extensions of ICS protocols and assess the potential to detect encrypted ICS protocols in Internet traffic.
6. We analyze the deployment status of encrypted ICS protocols seen from the Internet core.

The remainder of this chapter is structured as follows. Section 6.2 presents a taxonomy and related work about ICS protocols. Section 6.3 introduces our methodology and data sources to identify ICS traffic. Section 6.4 presents basic properties of ICS traffic seen at the IXP and ISP. Section 6.5 proposes a method to separate industrial and non-industrial ICS traffic. autorefsec:pub-ics-extension:trafficFeatures analyzes industrial ICS traffic in detail. Section 6.7 provides an upper bound of the usage of recent ICS protocol security extensions. Section 6.8 concludes our findings.

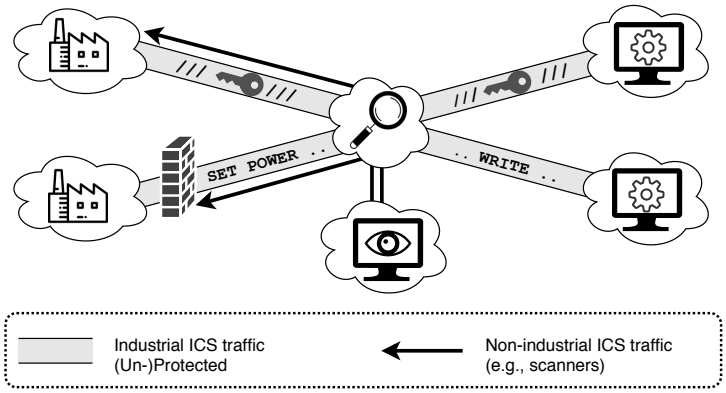


Figure 6.1: Analyzing unprotected ICS protocols.

6.2 Background and Related Work

6.2.1 ICS Protocol Taxonomy

ICS protocols are deployed in four major application areas [98]: (*i*) process automation, (*ii*) building management, (*iii*) smart grids including power plants, and (*iv*) metering infrastructures, an overview is presented in Table 6.1. All of these scenarios require security support when the ICS devices are interconnected via untrusted networks.

The most common use case for ICS protocols is process automation using Programmable Logic Controllers (PLC), which support manufacturing facilities by assisting production. PLCs are configured and queried by ICS protocols. Well-known protocols in this field are Modbus (general industrial networks), Siemens S7 (automobile), Ethernet/IP (time-critical applications), and HartIP (legacy wiring). Equipment and manufacturing facilities also rely on proprietary PLCs that utilize protocols such as Omron, GE-SRTP, Melseq-Q, ProConOS, or PCWorx. The Crimson protocol is used exclusively for Human Machine Interface (HMI) communication related to Red Lion units.

Remote management of buildings is significantly based on two protocols, BACnet and Niagara Fox. They are deployed to control heating, air-conditioning, lighting, fire detection etc. BACnet is used to communicate directly with controlling components. In contrast, Niagara is in use between management workstations, which then subsequently communicate with the controlling components.

Electrical and water companies use protocols such as DNP3, IEC60870-5-104, IEC61850 (goose, mms), Codesys, and ICCP to monitor and automate their power systems. DNP3 is a set of sub-protocols that were released in the early 1990s before the standards IEC60870-5-104 and IEC61850 have been established which became prevalent in this application domain.

Smart meters record the consumption of electric energy and communicate that information to billing centers. The standard protocol for this application in North-America is ANSI C12.22, which delivers measurement data as clear-text tables.

6.2.2 A Glimpse into ICS Protocol Security

Vulnerable ICS deployments have been highlighted since several years [214], [240]. The first reported incident is an unauthorized manipulation of an ICS which led to a pipeline explosion back in 1982 [97]. Although the absolute number of reported ICS incidents is fairly low [97], a single incident can be hazardous. To understand and improve the protection of ICS deployments, multiple efforts have been undertaken, including (*i*) the development of honeypots, (*ii*) Internet-wide scans to find open ICS devices, (*iii*) the improvement of intrusion detection systems for ICS, and (*iv*) the modeling and surveying of the ICS ecosystem.

ICS specific honeypots have been developed [14], [151], [289], [290] to understand the origin, frequency, and sophistication of attacks on ICS services. ICS services are popular victims. ICS honeypots receive significantly more requests after being listed on public scanning sites such as Shodan [140].

Two well-known scan projects, Censys and Shodan, detect globally reachable ICS services [170], [98]. Such scan results can be used to asses the security of ICS in individual countries [175]. ICS scans are dominated by few recurrent scanners [44] and captured within few days by honeypot deployments [170]. Mirian *et al.* [98] measured the increase of open ICS services of up to 20 % in 4 months.

Dedicated intrusion detection systems (*e.g.,* for smart meters [15]) and extensions to common IDS tools (*e.g.,* Snort and Bro [10], [86], [104]) have been proposed. Valdes [150] introduces an architecture that monitors ICS traffic for irregular patterns. Taking into account recent, distributed ICS deployments, Zhang [159] proposed a distributed multi-layered system.

ICS traffic patterns have been compared with SNMP traffic [12]. Both, ICS protocols and SNMP, show stable, periodical traffic patterns with a small number of constant host changes. However, ICS traffic does not present diurnal patters or self-similar correlations, features known from traditional network protocols [13]. In contrast to our approach, the data for this comparison was collected directly at the corresponding edge-network (traditional network, ICS-facilities). So no protocol classification was necessary.

ICS have been surveyed in several publications introducing historical background, taxonomies, and current security vulnerabilities [60], [88], [124], [134], [161], [162]. The number of Common Vulnerabilities and Exposures (CVEs) for ICS implementations grows steadily. Vulnerabilities are often discovered by simple fuzzing techniques [182], [142]. Also, the ICS ecosystem requires a secure supply chain [59]. Recent studies show the high DDoS potential of BACnet by analysing IXP and ISP packet samples over a period of 48 hours [47]. Yet, still open is a longitudinal analysis of unprotected ICS communication deployed in the global Internet.

Table 6.1: Overview of ICS Protocols. [ND/HD: Normal/Heuristic Dissector, C: Censys, S: Shodan, R: Rapid7, K: Kudelski]

Standard / Protocol	Ports	Use Case	Wirshark Dissectors	Min. # Bytes to identify protocol	Scan Projects	Honeypot Software	# CVEs
Modbus	502	Process automation	ND	74 B	C/S/K	✓	23
Siemens S7	102	Process automation	HD	93 B	C/S	✓	7
Ethernet/IP	2221, 2222, 44818	Process automation	ND	74 B	S	✓	9
BACnet	47808-47823	Building management	ND	46 B	C/S/R/K	✓	7
DNP3	20000	Smart grids	ND/HD	62 B	C/S	✘	39
HART IP	5094	Process automation	ND	78 B	S	✘	6
IEC60870-5-104	2404	Smart grids	ND	76 B	S	(✘)	0
ANSI C12.22	1153	Metering	ND	n/a	✘	✘	0
OMRON FINS	9600	Process automation	ND	54 B	S	✘	7
IEC61850 (mms)	102	Smart grids	ND/HD	144 B	✘	✘	0
Codesys	2455	Smart grids	✘		S	✘	20
GE-SRTP	18245, 18246	Process automation	✘		S	✘	7
Niagara Fox	1911, 4911	Building management	✘		C/S/K	✘	5
MELSEC-Q	5006, 5007	Process automation	✘		S	✘	2
ProConOS	20547	Process automation	✘		S	✘	1
PCWorx	1926	Process automation	✘		S	✘	0
Crimson	789	Process automation	✘		S	✘	0
ICCP-TASE.2	102	Smart grids	✘		✘	✘	8

6.2.3 The Problem of Unprotected ICS Protocols

Most of the common ICS protocols lack protection by design and are susceptible to eavesdropping and traffic manipulation attacks. The only exception is Niagara Fox, which provides authentication. However, authentication alone is insufficient. Attackers can scout their target and prepare a targeted attack without communicating with the ICS devices at all. Recent malware [277] exploits passive recording of ICS traffic traversing small enterprise routers. Such eavesdropping of unprotected ICS traffic is also possible on the inter-domain level.

Furthermore, it is important to note that infrastructure-based protections such as firewalls or NAT only partially help. They may prevent discovering ICS devices by *active* scanning but do not protect against *passive* listening and spoofed replay attacks.

In this chapter, we analyze the highly vulnerable part of the ICS ecosystem; those cases where operators interconnect their systems without any protection. This is challenging because unprotected industrial ICS traffic is suppressed by noise such as scan traffic.

6.2.4 ICS Scans Seen from an Internet Telescope

To motivate our aim for a detailed classification of ICS traffic, we briefly analyse data from the CAIDA/UCSD network telescope. This data source captures backscatter traffic from randomly spoofed DDoS attacks or Internet-wide scans of the /8 CAIDA/UCSD darknet. Any incoming traffic to the telescope is inter-domain and non-industrial.

Figure 6.2 shows the daily activity for Modbus (TCP/502), measured at the telescope. There is almost no activity visible until the beginning of 2014. Then, the amount of destination IP addresses that received data on the Modbus port increased by three orders of magnitude. The number of source IP addresses that sent data to the telescope increased by roughly one order of magnitude, indicating scanning from a small set of hosts. The sudden upturn in scan activities can be explained by (*i*) increased media coverage of ICS systems and (*ii*) increased research interest and consequently publicly available scan tools. Our observations correlate with the start of the ZMap and Censys projects. We saw no correlation with Shodan, which started to index ICS infrastructures in 2009 and added ICS protocols in 2012.

This brief analysis does not only highlights the increasing interest in ICS protocols but also the need for a careful methodology to analyse ICS traffic.

6.3 Identification of ICS Traffic

Two challenges need to be tackled for analyzing inter-domain ICS traffic. First, we need to reliably identify ICS traffic in global packet traces. Second, we need to distinguish industrial (*i.e.,* transferred by real deployments) from non-industrial ICS traffic (*e.g.,* scanning) . In this section, we propose our methodology to solve the first challenge, and tackle the second challenge in Section 6.5.

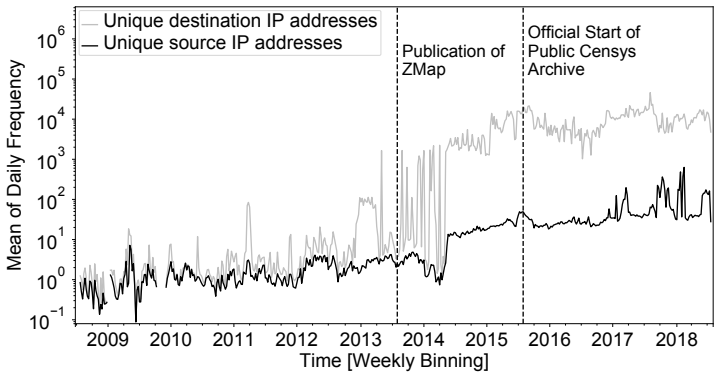


Figure 6.2: Internet-wide scanning of Modbus (TCP/502) observed at the CAIDA network telescope. We highlight research activities around one of the most common ICS scanners.

Table 6.2: Effects of data sanitization process and the ratio of remaining ICS packets by vantage point.

	Remaining Packets	
	IXP	ISP
Sanitizing steps after Wireshark ICS detection	100%	100%
❶ Removal of tunnel packets	99%	99%
❷ Removal of malformed packets	15%	52%
❸ Removal of NDPI fingerprintable packets	14%	51%
Comparison with vanilla approach		
Port-based detection relative to Wireshark	3950%	1340%

6.3.1 Collecting Traffic at Central Internet Vantage Points

We passively collect traffic at two different Internet vantage points, an IXP and an ISP. The two data source allow us to inspect traffic from two different perspectives, a rich interconnection fabric and an upstream provider.

Internet exchange Points (IXP) are centralized network infrastructures where heterogeneous domains intertwine. We receive data from a large, regional IXP from Europe with over 100 member networks with a daily traffic peak of 560 GBit/s. Due to the large traffic volume, flow data is not fully recorded but selectively sampled. We analyse non-anonymized packets collected from October 2017 until April 2018 with a sample rate of $\sim 2^{14}$. The sampled packets are truncated after 128 bytes. Flows from an IXP are inherently inter-domain.

Our second data source is the *Measurement and Analysis on the WIDE Internet* (MAWI) archive [239]. This archive contains daily traces describing 15 minutes of full traffic captures from a transpacific Internet link between Japan and the United States. We received a private MAWI data set with non-anonymized IP addresses and payload (96 bytes) for the same time range.

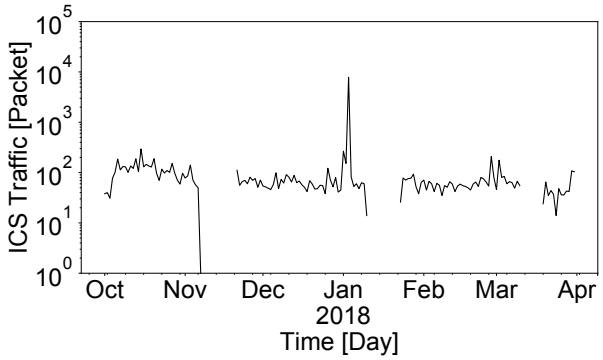
Non-anonymised flows allow for mapping with additional meta data, such as autonomous systems (AS). Please note that we are not allowed to release our data due to privacy constraints.

6.3.2 Identifying ICS Traffic Candidates

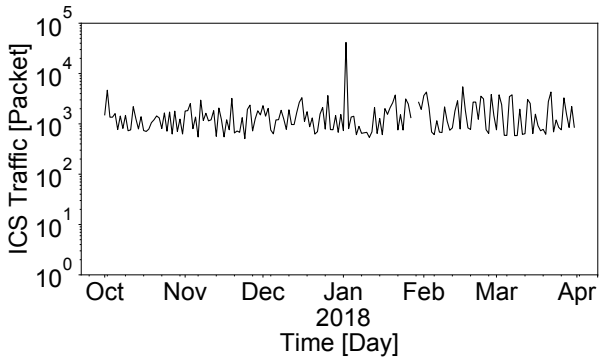
We explicitly do not want to implement new traffic classifiers as this conflicts with maintainability and reproducibility on the long-term. Instead, we want to leverage existing tools. We use Wireshark dissectors to find ICS traffic candidates. Half of the ICS protocols can be dissected by Wireshark, as shown in Table 6.1. Wireshark distinguishes between normal and heuristic dissectors (ND, HD). Normal dissectors identify protocols based on well-known port numbers and check whether the packets comply with simple sanity checks. If they fail, they forward the data to heuristic dissectors which apply pattern matching on protocol fields.

To verify the correctness of the Wireshark dissectors, we apply them on public ground truth data [292] and manually inspect the dissection of packet headers. All dissectors except one work accurately and map operation codes to protocol actions, such as *read* or *write*.

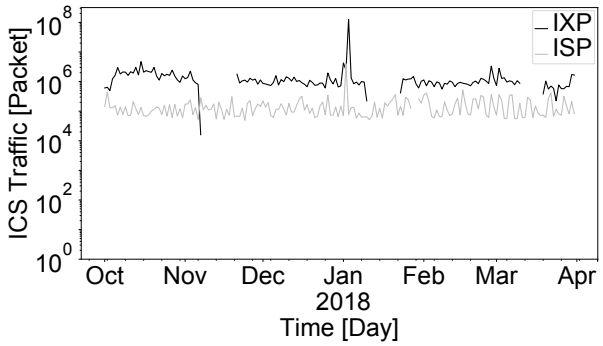
Packet sampling at our vantage points does not store complete packets but only a pre-configured fixed size of the overall packet. This limitation can lead to inaccuracies in identifying the application layer protocol because parts of the corresponding headers are missing. For each protocol, we reduce the packet length of the ground-truth data byte-wise and detect the minimal packet length required to identify the protocol correctly. All but one protocol dissector require less than 96 bytes, see Table 6.1. Considering that packets are truncated after 128 bytes at the IXP and 96 Bytes at the ISP, we can identify the ICS traffic candidates reliably.



(a) Internet Exchange Point (IXP)



(b) Internet Service Provider (ISP)



(c) Extrapolation by sampling rates.

Figure 6.3: Number of inter-domain ICS packets per day at two different vantage points.

6.3.3 Sanitizing ICS Traffic Candidates

We do not rely blindly on the Wireshark dissectors. We perform three data sanitizing steps to improve data quality: ❶ We remove tunnel traffic so that we only obtain plain end-to-end traffic. This step mainly excludes ICMP unreachable messages, which encapsulate the original UDP packets. Such backscatter packets are misclassified by Wireshark as ICS traffic. ❷ We remove packets which Wireshark marks as *malformed* or cases in which the dissector reports an *error*. This occurs when the protocol detection of a packet is successful, but the complete dissection fails due to header fields that do not comply with the protocol specification. ❸ We cross-validate our data by applying NDPI [253], a leading open-source deep packet inspection software. NDPI detects a broad range of protocols, but no ICS protocols. We exclude every packet that NDPI is able to map to a known protocol since we consider such a packet to be a false positive.

In Table 6.2, we quantify the remaining packets after applying our sanitizing steps. The data is shown relatively to the overall amount of identified ICS packets per vantage point. 85% of the packets at the IXP are classified malformed, and 48% at the ISP. Wireshark detects ICS protocols although many header fields are set to unspecified values, such as unknown operation codes. This highlights that Wireshark dissectors are rather optimistic and sanitizing is required for a reliable analysis. The removal of packets identified by NDPI accounts for less than 1%, which indicates a very low false-positive rate of our approach. Finally, we compare our approach with a pure port-based detection. Identifying ICS traffic only based on port numbers is not feasible as it leads to significant overestimation.

6.4 Properties of ICS Traffic

6.4.1 Daily Patterns and Prevalence of Inter-Domain ICS Traffic

During our measurement period, we identified 19k ICS packets at the IXP and 310k ICS packets at the ISP after sanitization. Figure 6.3 shows the number of daily ICS packets at the IXP and ISP. For better comparison, we consider the different sampling intervals and extrapolate the values (see Subfigure 6.3(c)). The daily ICS traffic at the IXP and ISP is constant apart from one anomalous peak at each vantage point. The traffic peak at the IXP is due to a large number of Ethernet/IP packets (217.5 MB/s traffic peak) during 10 minutes on January 3, 2018. The destination is a single IP address and the traffic is sent from several sources located in two autonomous systems. The traffic peak at the ISP consists of BACnet messages from 76 source IP addresses to 41,000 destination IP addresses. This event took place one day before the IXP peak. We observe uniformly distributed BACnet read messages, which indicates load balancing between scanning nodes. All sources relate to Rapid7 Sonar, a company that performs regular Internet-wide BACnet scans.

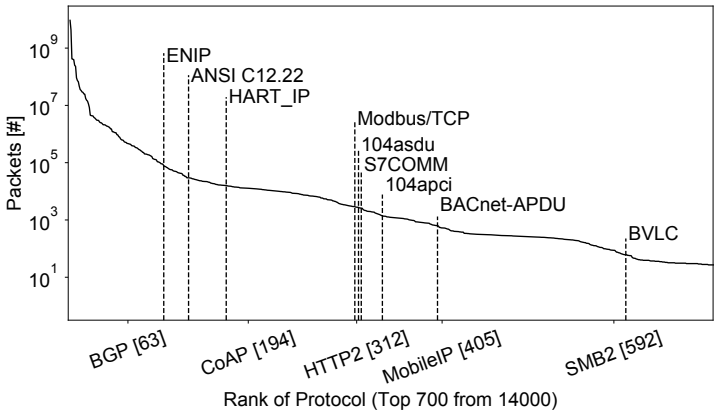


Figure 6.4: Protocols ranked by packet frequency as reported by Wireshark (non-sanitized), observed at a big national IXP during 6 months. ICS protocols are emphasized among some well-known protocols. Ranks are noted in brackets.

Compared to the total traffic volume, ICS inter-domain traffic is low. ICS packets only account for $\approx 0.0001\%$ of all sampled packets at the IXP and $\approx 0.002\%$ at the ISP. However, putting ICS traffic into perspective of well-known non-ICS protocols, ICS traffic is likewise prevalent, which we show in Figure 6.4. To allow for comparability, this graph visualizes the non-sanitized data set because implementing a sanitization process for non-ICS protocols would be out of scope of this work. This result emphasizes that ICS traffic should not be neglected.

6.4.2 ICS Message Types: Request vs. Reply

We refer to packets sent to a known ICS port as requests, and packets originating from a known ICS port as replies. Protocols with balanced request-reply ratios are likely to be used in a legitimate way since ICS communication patterns follow a common client server scheme. Observing significantly enhanced requests may have two reasons: (*i*) heartbeats sent from sensors to central servers that do not confirm the reception; (*ii*) scan traffic that reaches hosts which do not offer the corresponding service.

We analyze the ratio of requests and replies per protocol in more detail, check left-hand side of Table 6.4. We observe a tendency towards requests exceeding replies. Only at the IXP, HartIP and C12.12 show a balanced request-reply ratio. Strikingly, BACnet is very request-heavy across both vantage points. This might be an indication for non-industrial ICS traffic, which we will investigate further in Section 6.5.

6.4.3 ICS Traffic Sent to and Received from Autonomous Systems

To better understand the ICS ecosystem from a networking perspective, we map each source and destination IP address of a sampled packet to autonomous system numbers (ASN). We use

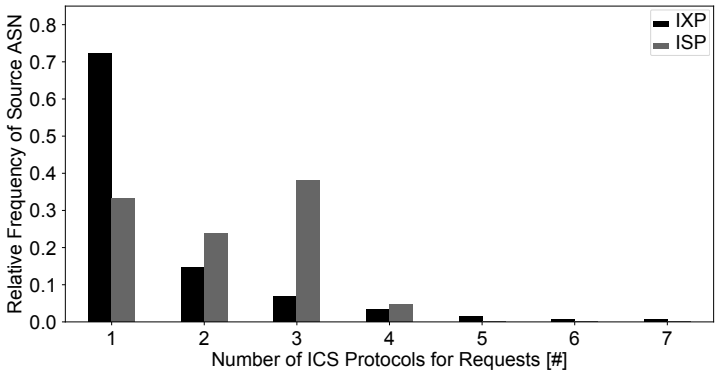


Figure 6.5: Number of ASes sending different ICS protocol requests. Since ICS deployments are rather specific deployment and bound to a single manufacturer, we rate several ICS protocols originating from a single AS as suspicious.

Table 6.3: Amount of successful reverse DNS lookups of source IP addresses per scan project.

	IXP	ISP
# Unique source IP addresses	1504	223
# Resolvable Censys IP addresses	105	n/a
# Resolvable Rapid7Labs IP addresses	7	56
# Resolvable Kudelski Sec. IP addresses	0	0
# Resolvable Shodan IP addresses	23	25

daily data from the RIPE RIS project and topological information from the IXP for assigning ASNs.

Autonomous systems which are the origin of request traffic via multiple ICS protocols host either scanners or heterogeneous ICS monitoring services. In our sanitized data sets, more than 70% of the ASes host nodes that deploy a single ICS protocol, see Figure 6.5. We find four cases of ASes creating requests for >4 distinct ICS protocols. Three are eyeball providers and one is a webhoster. These types of networks are common to connect scanners, which we detect in Section 6.5.

6.5 Identification of Industrial and Non-industrial ICS Traffic

Separating non-industrial from industrial ICS traffic allows us to identify the vulnerable part of the ICS ecosystem more precisely. We classify ICS traffic at our vantage points as non-industrial if the captured IP addresses belong to scan projects or have been observed at honeypots, as those indicate non-ICS hosts.

Table 6.4: Relative amount of industrial ICS traffic after applying different filter rules on the observed ICS traffic.

	Request response ratio				Traffic share after applying filters							
	# ICS Packets		Share of Requests		Excluding scanners		Excluding captured honeypot data				Excluding both	
	IXP	ISP	IXP	ISP	IXP	ISP	IXP		ISP		IXP	ISP
							HP_{ICS}	HP_{all}	HP_{ICS}	HP_{all}		
Total	19,060	310,996	81%	99%	97%	46%	97%	96%	15%	1.5%	96%	1.5%
BACnet	568	89,922	98%	100%	15%	7%	25%	11%	40%	1%	10%	1%
C12.22	1,559	24	63%	29%	100%	100%	100%	99%	100%	100%	99%	100%
DNP3	2	2,424	100%	100%	100%	99%	0%	0%	0.4%	0.1%	0%	0%
Ethernet/IP	9,171	171,804	94%	99%	99%	75%	98%	98%	5%	0.02%	98%	0%
HartIP	126	46,783	62%	92%	65%	9%	62%	62%	9%	8%	62%	8%
IEC60870	2,511	13	13%	38%	100%	100%	100%	99%	100%	100%	99%	100%
Modbus	2,547	–	95%	–	100%	–	100%	100%	–	–	100%	–
Siemens	2,576	–	99%	–	100%	–	100%	100%	–	–	100%	–

6.5.1 Filter Traffic of Common Scan Projects

Several projects scan for ICS hosts on a regular basis and thus contribute to non-industrial inter-domain ICS traffic. The most common projects are Censys, Shodan, Rapid7, and Kudelski. Censys, Rapid7, and Kudelski publicly document the IP prefixes from which they initiate scans. We use these prefix lists to identify scanners by marking an observed source IP address as scanner if the source IP address is covered by one of the prefixes.

To identify scanners that are not part of the documented IP prefixes, we perform reverse DNS lookups on all source IP addresses captured at our vantage points. By reviewing the assigned names manually, we find Censys, Rapid7, and Shodan scanners (*e.g., pirate.census.shodan.io* and *scanner2.labs.rapid7.com*). Note that we cannot identify any names that relate to Censys at the ISP because Censys performs scans between ≈ 8:00am and ≈ 6:00pm (UTC), whereas the ISP dumps include 15 minutes packet captures starting at 5:00am (UTC).

Table 6.3 shows the amount of successful reverse DNS lookups. The IXP and ISP share 86 source IP addresses, predominantly Shodan and Rapid7 scanners. The 5 most common source IP addresses at the ISP resolve to Shodan names and are located in Quasi Networks, an autonomous system which is also well-known for hosting malicious nodes [283].

6.5.2 Filter Traffic of Other Non-ICS Hosts

To account for other hosts that create non-industrial ICS traffic (*e.g.,* attackers), we leverage data from honeypots. Conpot [248] is the de-facto standard ICS honeypot but supports only five ICS protocols, one currently under development. Conpot implements limited variances in responses, which makes it easy to unmask as a honeypot. Thus, we argue to utilise transport layer honeypots in order to measure a broad scope of activities on ICS ports.

We deploy Honeytrap [276] in (*i*) a university network and (*ii*) a darknet, a network not offering any public services. Based on these honeypots, we identify suspicious source IP addresses. We create two lists: HP_{all}, which stores all IP addresses observed at the honeypots, and a subset of this list, HP_{ICS}, which stores IP addresses that sent requests to at least one ICS port. HP_{all} consists of 244k IP addresses and HP_{ICS} of only 3700 IP addresses (1.5%) from 619 ASes. It is worth noting that our honeypots also capture sources of the well-known scan projects. 224 IP addresses in HP_{all} are from Censys scanners.

We now correlate ICS traffic from our vantage points with the honeypot data. For every observed ICS packet, we check whether the source or destination IP address is present in HP_{all} or HP_{ICS}, see Table 6.4.

At the IXP, the overlap is minimal, which means that a significant amount of industrial ICS traffic is visible. 96% of ICS traffic is industrial based on HP_{all}, 97% based on HP_{ICS}. We perform a comparison per protocol and correlate 506 BACnet packets with HP_{all}, which represent 89% of the total BACnet packets at the IXP. These packets are classified as non-industrial ICS traffic and filtered. The results are stable, even if we only consider HP_{ICS}.

Table 6.5: Successful transport and application layer handshakes during Censys scans.

	# ICS hosts detected by Censys	
Protocol	Transport Scan	Application Scan
BACnet	31,735	31,154 (98%)
Modbus	8,400,058	126,984 (2%)
Siemens S7	7,202,828	24,946 (0.5%)

At the ISP, less industrial ICS traffic is visible. Filtering by HP_{all}, we find only 1.5% of the traffic to be industrial. However, the filtering is less effective if we only consider HP_{ICS}, especially for BACnet. The results indicate that it is beneficial to include honeypot information from non-ICS ports.

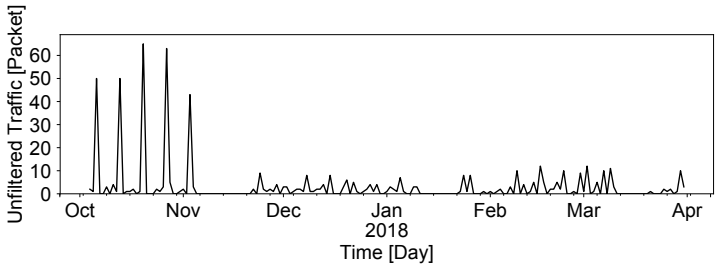
6.5.3 Benefits of Combining Filter Rules

To summarize the results from our previous filter steps, we provide an overview of the impact of the different filters. Table 6.4 shows the relative amount of ICS traffic that remains when traffic from scanners (identified by DNS names and IP prefixes), honeypots, or both is excluded.

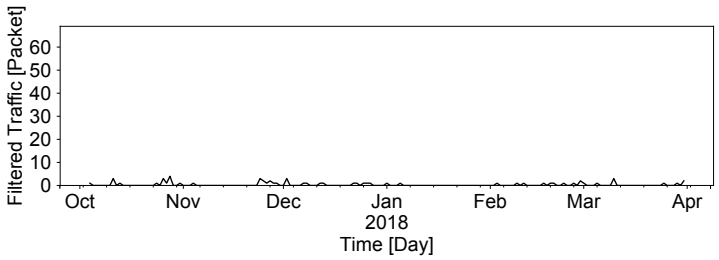
While we classify 96% of the traffic at the IXP as industrial, we see only 1.5% of industrial traffic at the ISP. Interestingly, more than half of the traffic at the ISP can already be classified as non-industrial only by excluding public scanners, *i.e.,* without maintaining a dedicated infrastructure such as honeypots. However, even though maintaining a honeypot introduces additional complexity, its data is necessary to provide a more complete view on distinguishing industrial and non-industrial traffic.

ICS protocols show similar trends for the the share of non-industrial traffic across both vantage points. The substantial difference for Ethernet/IP is caused by a Shodan scan of a complete prefix range at the ISP.

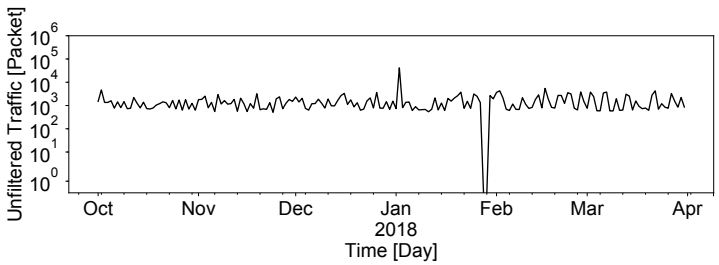
We show the potential effects of filtering non-industrial ICS traffic over six months in Figure 6.6. This enables us to describe the impact of non-industrial traffic over time. At the IXP we focus on BACnet as this protocol is severely affected by non-industrial activity. We make two observations: (*i*) At the IXP, non-industrial traffic consists mainly of ephemeral spikes at the beginning of our measurement period. (*ii*) At the ISP, the non-industrial traffic shows a very constant daily activity. After filtering at both vantage points, we obtain only a few industrial ICS packets per day which allows even for manual inspection of the ICS traffic.



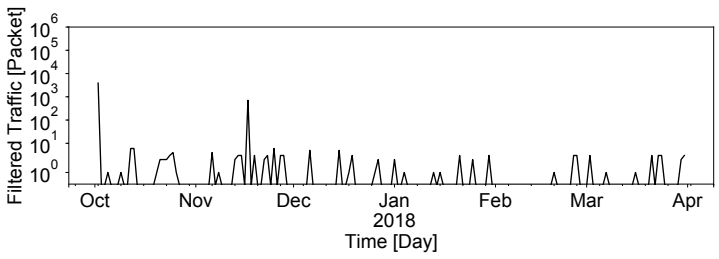
(a) IXP – Total BACnet Traffic.



(b) IXP – Industrial BACnet Traffic.



(c) ISP – Total ICS Traffic.



(d) ISP – Industrial ICS Traffic.

Figure 6.6: Daily amount of all ICS traffic versus industrial ICS traffic visible at the IXP and ISP.

Table 6.6: Ratio of ICS hosts observed at the IXP and Censys.

	% ICS hosts that overlap with Censys	
Host Type at IXP	Transport Scan	Application Scan
BACnet Source	0%	0%
BACnet Destination	0%	0%
Modbus Source	3%	0%
Modbus Destination	35%	0%
Siemens Source	0%	0%
Siemens Destination	65%	65%

6.6 Properties of ICS Industrial and Non-Industrial Traffic

6.6.1 Detecting ICS Hosts Protected by Firewalls

ICS devices might be protected by firewalls which grant access only from specific hosts. We analyze this by comparing IP addresses observed in our passive data with IP addresses of ICS devices revealed by active scans. To reduce overhead on the Internet infrastructure [69], we do not implement our own active probing but use data from Censys. Censys continuously scans the entire public IPv4 address space fast [38], [98], implements full transport and application layer handshakes [38], and releases weekly snapshots. We compare 3 ICS protocols for which we found industrial traffic and which are scanned by Censys during our measurement period: Siemens S7, Modbus, and BACnet.

First, we check how many ICS hosts are detected by Censys on the transport and application layer (see Table 6.5). Despite many successful transport layer handshakes, Modbus and S7 exhibit a very low success rate on the application layer. We argue that this is related to the use of lower port numbers that are more likely to be used by other applications which listen on the corresponding port. This complies with our previous results which showed that port-based ICS detection is misleading (see Subsection 6.3.3).

Now, we compare with ICS hosts observed at our vantage points. We compute the fraction of source or destination IP addresses that have been discovered by Censys (see Table 6.6) and for which we see communication in our passive data, *i.e.,* completely unprotected nodes. At the IXP, 35% of the Modbus and 65% of Siemens destinations are already known because of the transport layer scan. At the ISP, we do not find any correlation, *i.e.,* none of the ICS devices that are visible in our ISP traffic data set have been captured by active scans. This is very likely due to port-based access control lists which only allow communication between pre-configured hosts.

We find 3 source IP addresses that respond to Modbus transport layer scans but do not establish successful application layer sessions based on Censys. However, based on our traffic traces, each of these hosts has sent about 45 Modbus packets. One host is sending packets

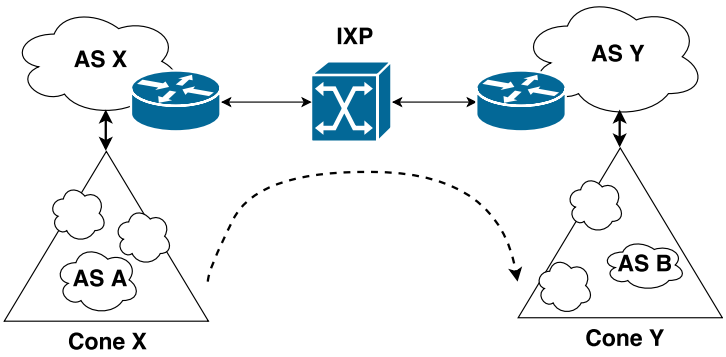


Figure 6.7: Example of cone to cone communication with ingress AS X and egress AS Y.

to a solar energy consulting agency. These results indicate cases of secure ICS services but unprotected ICS traffic.

6.6.2 Host Stability of Industrial ICS Traffic

Host stability describes how often a host is visible at our vantage points with respect to an activity span. For each destination IP address in the industrial ICS traffic, we calculate the size of the activity window w in days (*i.e.,* time-lag between first and last day of occurrence) and the number of active days n within this time window.

We assume that as soon as an ICS network is in place an embedded ICS device and an ICS control station will frequently exchange ICS traffic. Furthermore, we assume static assignment of IP addresses to those devices as this will ease operational maintenance (*e.g.,* configuration of firewall rules). Following both assumptions, hosts will achieve high host stability in case of real ICS networks, *i.e.,* the same IP address will appear for several days.

The IXP and ISP results differ significantly. In the IXP data set, the most stable host communicates almost every day ($w = 179, n = 146$). In contrast to this, in the ISP data set, hosts communicate less than 4%, relatively to the overall activity span.

To better understand whether stable hosts belong to a real ICS deployment, we map IP addresses to additional meta data: reverse DNS records and `whois` data. Based on this, we find that hosts are operated by a building company (*max-boegl.de*; $w = 179, n = 146$), a trade and transport company (*Handel Uslugi Transport Ewa Cielica*; $w = 159, n = 98$), and a industrial service and consulting company in the field of solar energy (*enerparc.com*; $w = 90, n = 36$). The high number of active days, despite the sampling, indicates a high exchange of messages. Interestingly, these hosts are not marked as ICS hosts by Censys, indicating the role of an ICS monitoring station. In the data set of our transnational ISP, we do not find evidence for ICS companies.

Table 6.7: Relative ratio of traffic transitions for three ICS protocols at IXP. Non-industrial traffic originates exclusively from cones and thus is not local.

	Industrial			Non-Industrial		
	BACnet	HartIP	Ethernet/IP	BACnet	HartIP	Ethernet/IP
Member to Member	30%	22%	5%	0%	0%	0%
Member to Cone	24%	51%	29%	0%	0%	0%
Cone to Member	19%	9%	6%	46%	79%	52%
Cone to Cone	27%	18%	60%	54%	21%	48%
# Flows	59	78	9006	509	48	165

6.6.3 Locality of Non-Industrial Traffic

We analyze the locality of industrial and non-industrial ICS traffic. Less local traffic is more likely to be part of Internet-wide scanning activities, whereas some ICS stakeholders may consider locality as reason not to protect (industrial) ICS traffic. We distinguish between topological and geographical locality.

Figure 6.7 shows a typical inter-domain topology at an IXP. In addition to a source and destination AS, packets may traverse *ingress* and *egress ASes* directly connected at the IXP. ASes which send or receive packets over an IXP member are in the cone of this member. We refer to traffic as IXP local, if the following condition applies:

$$Source\,AS == Ingress\,AS \quad \wedge \quad Egress\,AS == Destination\,AS$$

From a topological point of view, IXP local traffic is more *trustworthy*, because both ASes peer directly with each other (maybe via a route server). In contrast to this, communication from cones is rather expected from Internet-wide scanners, which are located in edge networks. At the IXP, 90% of the BACnet and 40% of the HartIP traffic is non-industrial. Comparing peering transitions for these two protocols with Ethernet/IP, which exhibits only 2% non-industrial traffic, shows a clear distinction, see Table 6.7. Non-industrial traffic originates only from the cones of the IXP-members, hence is not local at the IXP.

Assuming that critical infrastructures are scanned by malicious hosts and proxies from ASes located in foreign countries, we also check how often traffic is locally bound to a country. We do this by mapping the IP addresses to country codes based on MaxMind [123]. If the source and destination IP addresses are located in the same country, we call the traffic *domestic*. Table 6.8 presents the results of our analysis of domestic traffic. Although industrial traffic is also exchanged across country borders (which might happen in the case of, *e.g.,* global transport companies), there is a clear trend for non-industrial traffic: Non-industrial traffic is strictly non-domestic, which highlights globally distributed scanning activities. On the other hand, up to 29% of the industrial traffic is local, which makes it easy to contact and train the ICS network operators in charge.

Table 6.8: Relative ratio of domestic traffic for three ICS protocols, compared to the overall traffic of each protocol at the IXP.

Industrial			Non-Industrial		
BACnet	HartIP	Ethernet/IP	BACnet	HartIP	Ethernet/IP
29%	24%	1%	0%	0%	0.5%

6.7 Encrypted ICS Traffic

6.7.1 ICS Protocols and (D)TLS Extensions

To reduce the attack space of the vulnerable, traditional ICS traffic, ICS protocols have been recently extended to support Transport Layer Security (TLS) and Datagram TLS (DTLS). While TLS works on top of TCP, DTLS works on top of UDP. These extensions ensure three security objectives:

1. integrity, *i.e.,* manipulated data is rejected,
2. authenticity, *i.e.,* messages from untrusted devices are rejected,
3. authorization, *i.e.,* not allowed actions are rejected.

The most recent TLS standard is version `1.3`, which provides major improvements in the areas of security, performance, and privacy. Most strikingly, TLS `1.3` enhances the handshake behavior by encrypting more of the initial negotiation to protect privacy-sensitive data from eavesdroppers. Also, an entire round trip from the connection establishment phase is removed.

We are aware of three ICS protocols that are extended by (D)TLS: Ethernet/IP [166], DNP3 [259], and Modbus [268]. All of these protocols use a different default transport port in the encrypted version compared to the unencrypted version (see Table 6.9). Ethernet/IP and Modbus enforce the TLS standard `1.2`. This does not allow TLS downgrades during handshakes, which makes both protocols vulnerable to older TLS-based attacks [272]. DNP3 uses a proprietary security extension called Secure Authentication (SA), in addition to TLS. Please note that DNP3 SA only provides fine-grained device authentication and message integrity [242], [132]. Authentication can be performed in either direction (outstation or master) and access control lists allow to enforce roles within an organization. However, as DNP3 SA does not provide encryption, it does not protect from eavesdropping or prevents ICS detection by passive traffic analysis. In this analysis, we only focus on fully encrypted traffic based on TLS and DTLS.

6.7.2 Attack Vectors for Encrypted ICS Traffic

Unencrypted Transport Headers. ICS traffic can be secured on three different layers, the network layer (IPsec), transport layer (TLS), and within the application (*e.g.,* SA) [132].

Table 6.9: Security extensions for ICS protocols.

ICS Protocol	Extension	Default Port
Ethernet/IP Secure	TLS & DTLS	2221
DNP3 Secure	TLS & SA	19999
Modbus Secure	TLS	802

Extending each protocol based on (D)TLS has the advantage of minimal setup requirements. (D)TLS, however, does not prevent eavesdroppers from dissecting network and transport layer headers. Thus, attackers are able to conduct a port-based analysis, trying to detect ICS deployment. Limiting the traffic to this subset, *i.e.,* focusing on (encrypted) traffic on the respective ports only, reduces computational complexity and it becomes easier for attackers to detect interesting targets. After detecting ICS deployments, attackers can utilize other attack vectors in addition to traffic manipulations to disturb operations, *e.g.,* volumetric DDoS attacks [266] or IP prefix hijacks.

Machine Learning Classifiers of ICS Traffic. An application protocol can be identified in encrypted traffic even if not only the application but also the transport and network layer are covered, *e.g.,* in RDP tunnels [39]. Usually, characteristics that allow for fingerprints are extracted based on statistical analysis. Such methods use a rich training data set and then apply the trained classifier to identify features on a target data set [152]. Machine learning approaches can also be used in the context ICS protocols, *e.g.,* DNP3 message types can be identified with high precision in encrypted IPsec tunnels [148]. We discuss, however, that hose approaches conflict with inter-domain traffic analysis as they are challenged by sampled data.

We will now inspect traffic activities on default ports of encrypted ICS protocols. Then, we will evaluate the potential of statistical fingerprints, *e.g.,* by machine learning, of ICS traffic at IXPs.

6.7.3 Traffic Activities on New ICS Ports

We now analyze the traffic volume at the IXP for default ports of ICS protocols that support encryption extensions, see Table 6.9. We do this as a longitudinal study of more than two years so that traffic changes due to the specification and market release of the new extensions become noticeable. It is worth noting that we count packets on ports independent of the transport layer payload. Also, we do not exclude non-industrial ICS traffic in order to observe potentially increased scanning activity.

Overall, we did not observe any significant increase of traffic in the last two years on the respective ports (see Figure 6.8). The total number of sampled packets to or from the new ports remains small. Only 0.013% of the total daily packet volume at the IXP can be attributed to ports that belong to encrypted ICS protocols. At the beginning of the measurement period, we observe synchronized valleys on the Ethernet/IP and DNP3 Secure ports. Also, the Ethernet/IP

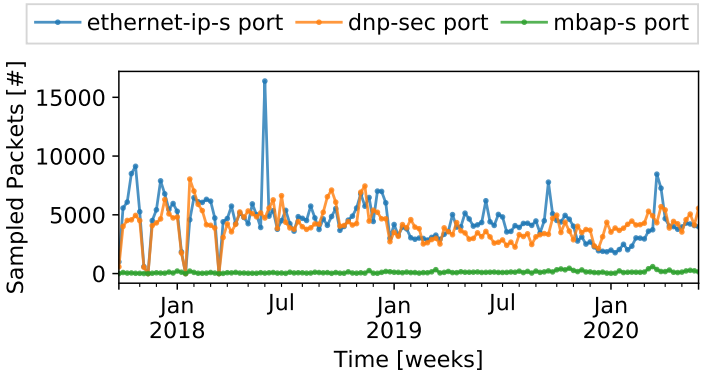


Figure 6.8: Number of packets associated with the ports of ICS protocols with encryption extensions.

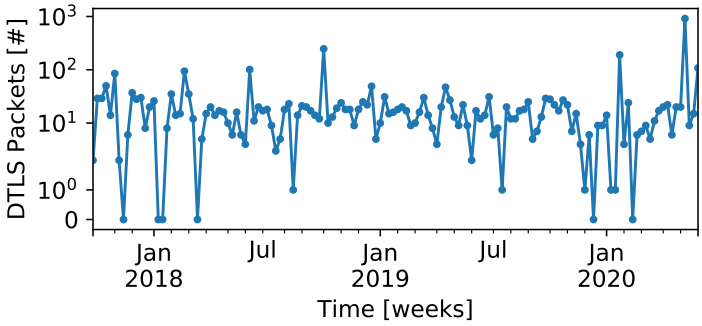


Figure 6.9: Number of DTLS packets associated with the Ethernet/IP secure port.

port exhibits one extreme traffic peak in mid 2018. Unfortunately, we could not find any links between these events and see no affiliation to ICS deployments. Based on that, we conclude two findings:

1. The encrypted versions of the ICS protocols have not yet been incorporated by ICS operators and also (potentially malicious) scan projects.
2. A simple port filter allows attackers in the Internet core to reduce the number of potential ICS candidates substantially, *i.e.,* the analysis becomes less computationally heavy.

Motivated by the second finding, we now inspect the encrypted application data for specific fingerprints.

6.7.4 Application Fingerprints at the IXP

We now leverage Wireshark to fingerprint (D)TLS in the detected traffic. Using ground truth data, we verified that Wireshark is able to detect (D)TLS, even in scenarios that include truncated packets. Wireshark detects (D)TLS traffic heuristically, *i.e.,* by inspecting (D)TLS record

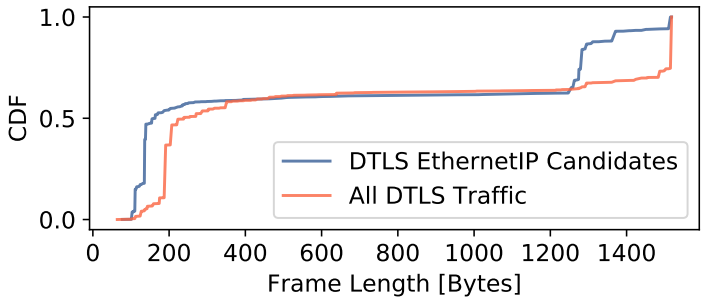


Figure 6.10: Packet sizes for all DTLS packets and Ethernet/IP secure traffic candidates.

headers for valid content types and TLS versions, which are represented as a 1-complement. At the IXP, we do not find any TLS packet related to the ICS ports but we do find on average 26 sampled DTLS packets per week (3.7k in total). All these packets include the Ethernet/IP source or destination port, compare Figure 6.9. Please note that this plot represents the upper bound for Ethernet/IP secure packets which are sent using default configuration.

We try to infer whether the encrypted traffic is indeed Ethernet/IP. Unfortunately, common features used in machine learning to infer application types are not available at the IXP due to aggressive packet sampling and truncation. Such features include the packet inter-arrival times, bi-directional traffic flow analysis, bit rates, *etc.* [152]. Even though we are challenged by truncated packets, we can still determine the packet sizes of the IP packets by inspecting the IP length field, which is not truncated. In case of ICS packets, we expect that these DTLS packets are smaller compared to all other DTLS traffic. We observe a mostly bi-modal distribution for both traffic types, exhibiting different sizes of classes (170 vs. 200 bytes, 1250 vs 1500 bytes), though, see Figure 6.10. Overall, if encrypted Ethernet/IP traffic is present in our candidate sets, it remains well hidden and cannot be inferred by statistical and machine learning analysis at the Internet core.

6.7.5 Stable ICS Deployments and Encrypted Traffic

As an additional crosscheck, we test whether encrypted traffic is sent by stable ICS deployments which previously exchanged unencrypted ICS traffic. To this end, we look for IPsec, *i.e.,* Encapsulating Security Payloads (ESP), and again DTLS traffic for such deployments. We find no DTLS traffic. We find, however, two ICS deployments, which used an IPsec point-to-point tunnel and exchanged 74k and 260 sampled packets, respectively. To better understand the underlying deployment, we map IP addresses of these packets to their origin autonomous systems. One of the tunnel end points is connected to an eyeball provider, the other to an architecture office. Based on these observations we suspect that the tunnels primarily carry office-related traffic.

6.8 Conclusion

In this chapter, we analyzed the unprotected traffic of protocols that interconnect industrial control systems (ICS). Our key results obtained from an IXP and an ISP perspective, *i.e.,* the Internet Core, read:

ICS traffic identification is painful. Common open source tools for traffic classification and analysis do not identify ICN traffic reliably. Due to the limited deployment of ICS protocols, there is a lack of fingerprinting tools. We introduced and explored an advanced but lean approach to detect ICS protocols. Our methodology is based on common Wireshark dissectors, but introduced several sanitizing steps that reduce the number of false positives. Given that we have identified ICS scanners as well as industrial ICS deployments in our traffic traces, we are confident with our true positives.

Unprotected ICS traffic is visible at the IXP. After sanitizing our data, we found over 330k ICS packets and one anomalous traffic peak at each vantage point. As Internet-wide ICS scanners operate since several years, it comes as no surprise that inter-domain ICS traffic exists. Hence, we developed a classification mechanism to differentiate between industrial and non-industrial ICS traffic. The 96%-share of unprotected industrial ICS traffic at the IXP is alarming. Since we observe a regional IXP, cooperating companies from the same region might exchange ICS traffic. In contrast, our ISP data represents a transnational link between USA and Japan, representing the bridge between geographically distributed transit networks. Intuition suggests, that distributed ICSs are rather local in deployment. Our results confirm this intuition. With only 1.5% industrial ICS traffic, the ISP is mainly confined to scans.

New, stable ICS deployments detected. We isolated (non-) industrial ICS traffic, and could classify ICS packets that were exchanged by hosts such as known from the Censys scan project. We also discovered previously undetected ICS devices, though, that belong to real ICS eco-systems. We identified cases of very stable hosts, *i.e.,* hosts that exchange ICS traffic regularly. Such hosts are vulnerable to traffic manipulation attacks on a daily basis. We spotted topological features for non-industrial ICS traffic. Such traffic originates at IXP-cones and is not domestic, *i.e.,* source and destination are not located in the same country.

ICS Security extensions are disguised. We present a first study of ICS Security extensions at the IXP, with a focus on DTLS traffic. We do not find tangible signs of encrypted ICS traffic. Nonetheless, we present an upper bound for encrypted ICS traffic at the Internet core. First results suggest that (transport layer) ports registered for the security extensions experience substantially less traffic than other ports. This reduces the data size and hence computational complexity for attackers, which attempt to find new ICS deployments.

Raising awareness of potential ICS attacks. The insights of this chapter help to find unprotected ICS traffic and inform responsible stakeholders for improving protection. They also allow to deploy a long-term monitoring system that can observe malicious inter-domain ICS

activities. Solutions already exist (such as SSH tunnel, VPN, *etc.*), but are not yet deployed, leaving ICS data exposed to eavesdropping and traffic manipulation attacks.

Future Work. In the future, we hope that ICS deployments will upgrade from unprotected configurations to secure ones. Hence, we expect increased traffic volumes on the default ports of the secure ICS protocol variants. This will pave the way for more extensive analysis, including machine learning methodologies which require larger data sets. Overall, observing ICS traffic from the Internet core will remain relevant (*i*) to quantify malicious scanning activities and (*ii*) to detect misconfigured ICS deployments, even with security extensions.

Chapter 7

Transparent Forwarders: An Unnoticed Component of the Open DNS Infrastructure

Abstract

In this chapter, we revisit the open DNS (ODNS) infrastructure and, for the first time, systematically measure and analyze transparent forwarders, DNS components that transparently relay between stub resolvers and recursive resolvers. Our key findings include four takeaways. First, transparent forwarders contribute 26% (563k) to the current ODNS infrastructure. Unfortunately, common periodic scanning campaigns such as Shadowserver do not capture transparent forwarders and thus underestimate the current threat potential of the ODNS. Second, we find an increased deployment of transparent forwarders in Asia and South America. In India alone, the ODNS consists of 80% transparent forwarders. Third, many transparent forwarders relay to a few selected public resolvers such as Google and Cloudflare, which confirms a consolidation trend of DNS stakeholders. Finally, we introduce DNSRoute++, a new traceroute approach to understand the network infrastructure connecting transparent forwarders and resolvers.

7.1 Introduction

The open DNS infrastructure (ODNS) [139] comprises all components that publicly resolve DNS queries on behalf of DNS clients located in a remote network. This "openness" makes the

Table 7.1: Comparison of known open DNS components.

	2014	2020	2021			
	[81]	[117]	[174]	[278]	[273]	This Work
# Rec. Resolvers	n/a	20K	50K	n/a	n/a	32K (2%)
Forwarders						
# Recursive	n/a	1.4M	1.7M	n/a	n/a	1.5M (72%)
# Transparent	0.6M (2%)	n/a	n/a	n/a	n/a	0.6M (26%)
All ODNSes	25.6M	1.42M	1.75M	1.8M	1.6M	2.125M

ODNS system a popular target for attackers, who are in search for amplifiers of DNS requests, for periodic DNS scan campaigns, which try to expose the attack surface, and for researchers, who want to learn more about DNS behavior.

Originally observed in 2013 [238], *transparent* DNS forwarders have not been analyzed in detail since then, but fell off the radar in favor of *recursive* forwarders and resolvers. This raises concerns for two reasons. First, the relative amount of transparent forwarders increased from 2.2% in 2014 to 26% in 2021 (see Table 7.1). Second, as part of the ODNS, they interact with unsolicited, potentially malicious requests.

In this chapter, we systematically analyze transparent forwarders. Our main contributions read as follows:

1. We characterize transparent forwarders and review DNS measurement methods. (Section 7.2)

2. We experimentally show that popular DNS scanning campaigns do not expose transparent forwarders and thus provide an incomplete view on the ODNS threat landscape. (Section 7.3)

3. We measure the impact of transparent forwarders and find diverse deployments, heavily dependent on the hosting country. For example, configurations of forwarders in South America and Asia greatly contribute to DNS consolidation. (Section 7.4)

4. We introduce `DNSRoute++`, a new traceroute approach that leverages the behaviour of transparent forwarders and explores interconnectivity in the ODNS. (Section 7.5)

5. We discuss transparent forwarders in a broader context. Most of the transparent forwarders are CPE devices, either serving single end customers or larger networks. (Section 7.6)

7.2 Background and Related Work

Open DNS (ODNS). Various DNS stakeholders [5] such as domain owners and network operators operate autonomously and pursue different goals. A common view on the DNS is the ODNS infrastructure [139], client-side DNS speakers that openly accept requests from any host (not related to oblivious DNS, *i.e.,* ODoH). ODNS components have been previously classified into *recursive resolvers* and *forwarders* [6], [139]. Recent Internet-wide scans show that the majority (95%) of ODNSes are forwarders [117] but prior work does not further distinguish between recursive and transparent forwarders and mainly assume only the presence of recursive forwarders [80].

Figure 7.1 shows the expected behavior of all three ODNS components, which are commonly used by stub clients. Recursive resolvers send queries recursively to authoritative name servers

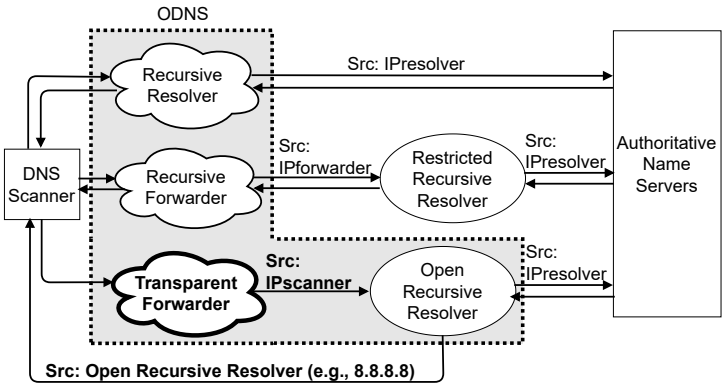


Figure 7.1: Overview of various ODNS components and their relation to common Internet-wide scan setup.

and respond with the final answer to the original client (*e.g.,* scanner). In contrast, forwarders do not use DNS primitives to resolve names but forward queries to a recursive resolver [198]. Recursive forwarders can relay to restricted resolvers, however, transparent forwarders must forward to open resolvers to act as an ODNS (otherwise, the resolver rejects the scanner IP address). Upon receiving a final answer, a forwarder may cache it and relays it to the client. Forwarders are often susceptible to fragmentation [160] or side-channel [95] attacks.

Introducing Transparent Forwarders. In this chapter, we argue that there are DNS forwarders that do not receive DNS answers because they operate completely transparent. Such deployment makes the distinction between recursive forwarders and transparent forwarders necessary.

A *recursive forwarder* replaces the original source IP address of the client by its own IP address. A *transparent forwarder* keeps the IP address of the original requester (*e.g.,* $IP_{Scanner}$). The relaying behavior of transparent forwarders has two implications. First, answers are sent back directly from resolvers to the original requester, *i.e.,* they are neither observed nor cached by the forwarder. Second, transparent forwarders spoof the IP address of the requester.

Surprisingly, Internet-wide, single packet scans lead to multiple answers from the same host, *e.g.,* 314k responses from `8.8.8.8`. Our study verifiably links these to transparent forwarders. Prior work removes these in a sanitizing step [117] or describes them as *unexpected* [89] but falls short to identify the root cause. So far, transparent forwarders have been treated as an artifact which can be utilized to measure missing outbound source-address-validations [72], [81].

Comparison of ODNS Classification Methods. Two methods are common to distinguish recursive resolvers and forwarders: (*i*) Destination-specific DNS queries from a scanner, which encode the destination IP addresses as a subdomain into the query name (*e.g.,* `203-0-113-1.mydomain.com`). (*ii*) Source-specific responses from an authoritative name server, which inserts the IP address of the immediate client (*e.g.,* `203.0.113.1`) into a dynamic `A` resource record of the query

Table 7.2: Comparison of forwarder detection methods.

	Custom Queries [4], [72], [81], [139]	Custom Responses [174], [117], this work
Utilization of caches	None	High
Load auth. name server	High	Low
Forwarder detection	At server	At client
Forwarder classification	At client	At client

name (*e.g.,* `mydomain.com A 203.0.113.1`). This method can utilize two `A` resource records, a client-specific record and a static control record to check for DNS manipulations.

The first method enables an analysis at the name server. If the IP source address of an immediate client matches the encoded IP address within the query name, then the scanned destination is a recursive resolver, and a forwarder otherwise. The second method requires an analysis at the node that originally sent the query (*e.g.,* a DNS scanner). If the IP source address of the response matches the IP address within the `A` record, then the scanned destination is a recursive resolver, otherwise the scanned node is a DNS forwarder. This condition does not hold true anymore for transparent forwarders as recursive resolvers reply directly to the scanning node.

Table 7.2 summarizes the (dis-)advantages of both methods. The query-based method is particularly useful when the measurement objective needs to prevent caching, because the query name is unique for each target. This increases the load at the authoritative name server, though. The response-based method keeps the load at the authoritative name server low since it allows to utilize caches. Although the first method allows to detect forwarders already at the name server, classifying forwarders into recursive and transparent is only possible at the scanning node because the source IP address of the response has to be evaluated. Such a classification requires a correlation of DNS requests and responses to reflect the full DNS transaction. Hence, we will deploy the latter method in Section 7.4.

7.3 Popular Scanning Campaigns and Transparent Forwarders

Censys [174], Shadowserver [278], and Shodan are popular scanning campaigns to reveal ODNSes. To verify our assumption that these campaigns underestimate the current number of ODNSes because they only record responses without correlating with the original target IP addresses of requests, we conduct a controlled experiment.

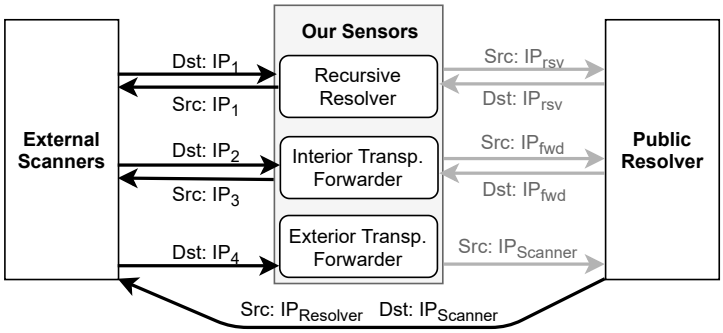


Figure 7.2: DNS sensors. Black arrows indicate DNS messages visible to external scanning campaigns.

7.3.1 Controlled Experiment

We develop and deploy three ODNS honeypot sensors, see Figure 7.2.

Sensor 1: Recursive Resolver. The first sensor behaves exactly like a public recursive resolver. The sensor answers using the same IP address at which it also has received a DNS request, IP_1. This configuration is a baseline measurement. We expect every viable Internet-wide DNS scanning campaign to find this sensor.

Sensor 2: Interior Transparent Forwarder. We utilize two IP addresses, IP_2 to receive DNS requests from a scanner and IP_3 to send responses. Both IP addresses are part of the same `/24` prefix. This configuration allows for the following inferences: (*i*) Scanners that report IP_2 ignore the different IP address IP_3 in the response. They are RFC-compliant [245], and implement DNS transactional scans. (*ii*) Scanners that report IP_3 only evaluate the responses independently of the sent requests, which is a strong indicator for stateless, response-based analysis. This sensor mimics the key behavior of a transparent forwarder, but, as both addresses belong to the same IP prefix, the setup is easy to deploy. It does not require special network configuration such as disabled source address validation. Moreover, we can ensure that a reply is sent to the scanner.

Sensor 3: Exterior Transparent Forwarder. The third sensor implements a transparent forwarder which relays spoofed packets to an external, public resolver. This sensor is reachable at IP_4 and forwards a request using the source IP address of the scanner. To allow for spoofing, this sensor should be connected to a network that does not deploy source address validation [190] and peers directly with the network of the public resolver. In contrast to the previous setups, we do not receive the answer from the public resolver since the answer is sent directly to the scanner. Similarly to sensor (2), we can infer the following: (*i*) Scanners that report IP_4 ignore the different IP address in the response, indicating transactional scans. (*ii*) Scanners that report

Table 7.3: Detection of our DNS sensors by popular scans.

	Detected			
	Sensor 1	Sensor 2		Sensor 3
Scanner	IP_1	IP_2	IP_3	IP_4
Shadowserver	✓	✘	✓	✘
Censys	✓	✘	✘	✘
Shodan	✓	✘	✘	✘

the public resolver will miss our forwarding sensor. This is because multiple responses from the same source will be aggregated into a single DNS speaker.

Deployment Details. Our sensors resolve incoming requests using Google's public resolver. We verify that source address validation is not deployed in our network. Moreover, our network peers directly with Google at an Internet eXchange point (IXP), so we are not exposed to filters from upstream providers, as required for sensor 3. We confirm the correct operation of all sensors by sending DNS queries and analyzing replies at the scanner. To prevent amplification attacks [266], we configure a strict rate limiting such that each sensor is allowed to answer one request every 5 minutes per source `/24` prefix. We use a rate limiting based on the client prefix since it also prevents DoS carpet bombs [52]. We deploy our sensors for multiple weeks and then inspect the scan project results.

7.3.2 Results

All three sensors received scans from Censys, Shadowserver, and Shodan, but those scanners did not identify all of our sensors as an ODNS component. We use Censys' and Shodans public search API to check which IP addresses of our sensors have been discovered. As owner of the IP prefix that we used for our sensors, we have been informed by Shadowserver about our sensors.

All measurement campaigns discovered our public resolver (Sensor 1). None of them found one of our DNS forwarders, see Table 7.3. Shadowserver reported the replying IP address IP_3 of Sensor 2, which, in real deployment, would represent the address of a recursive resolver. However, Censys and Shodan did not report IP_3, which indicates that the responses did not pass a sanitizing step, respectively. We conclude that transparent forwarders are currently missed by these scanning campaigns. Given that the measurement results of these campaigns are used by third parties, the impact of ignoring transparent fowarders is large. National CERTS, for example, rely on data from Shadowserver to identify local ODNS systems.

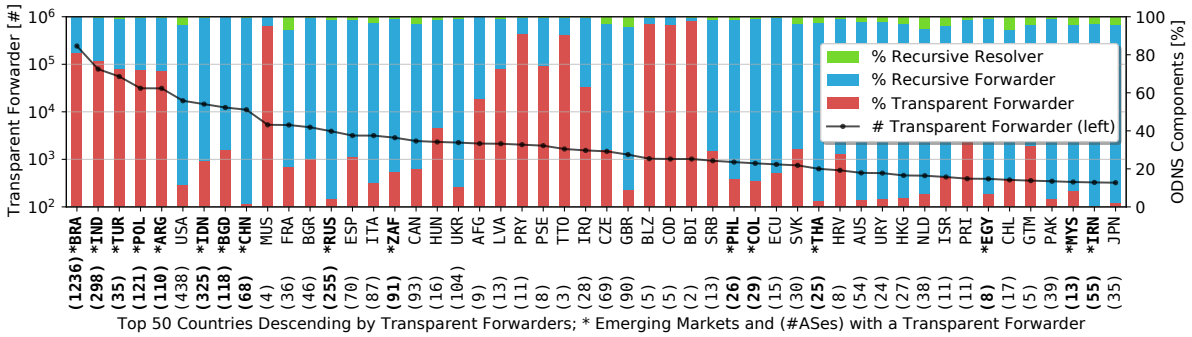


Figure 7.3: Top-50 countries with transparent forwarders. Countries with emerging markets exhibit more transparent forwarders.

7.4 Measuring and Analysing Transparent DNS Forwarders

7.4.1 Measurement Method and Setup

Method. To identify transparent forwarders, we need to correlate requests and responses at the scanning node. Our method aims for easy deployment, low measurement overhead, and robustness against manipulations. It requires two steps. First, mapping replies to requests of our scans. Second, classifying ODNS components.

To implement the first step, our scanner records the complete DNS transaction, *i.e.,* source and destination IP addresses, client port, and the ID used in the DNS header [245]. Assigning replies to requests based only on IP addresses would introduce ambiguity since replies triggered via transparent forwarders will include the source IP address of the resolver. Furthermore, to enable Internet-wide parallel scans, we ensure unique tuples of transport port and ID similar to other asynchronous scanners [38]. Then, even if we receive replies from the same resolver used by different transparent forwarders, we can clearly map responses to requests (for a detailed example, compare appendix Figure 7.7).

Our scanner requests a static name that belongs to a DNS zone which we control. The corresponding authoritative name server replies with two `A` records similar to other approaches using client-specific responses (details see Section 7.2). Performing full DNS transactions and using a control resource record also helps us to identify distortions introduced by middleboxes [62]. After receiving replies, we correlate the client port number and DNS transaction ID of responses and previously recorded request data. We use a conservative DNS timeout of 20 seconds. Note that this and the subsequent analysis of forwarders is part of post-processing the data. It does not affect the speed of scanning.

We then classify ODNS components. Utilizing the destination address of the request (IP_{target}), the response source address ($IP_{response}$) and dynamic `A` resource record ($A_{resolver}$), we apply:

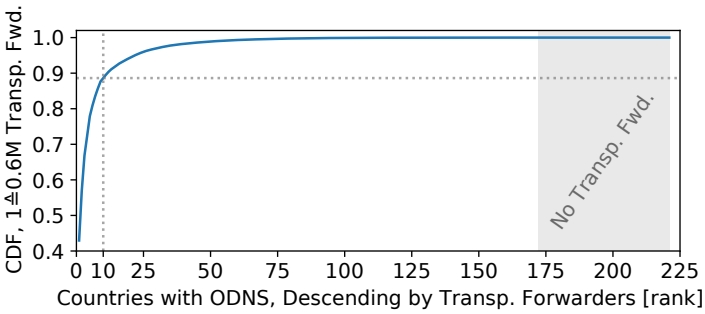


Figure 7.4: CDF of transparent forwarders per country. Top-10 countries exhibit ∼90% of all transparent forwarders.

Transparent Forwarder if

$$IP_{target} \neq IP_{response}$$

Recursive Forwarder if

$$IP_{target} = IP_{response} \wedge IP_{response} \neq A_{resolver}$$

Recursive Resolver if

$$IP_{target} = IP_{response} \wedge IP_{response} = A_{resolver}$$

Setup. We deploy our scanner in a network, which allows for high packet rates without triggering a DoS attack mitigation such as packet drops or rate limiting. We probe any public IPv4 address and use moderate scanning rates, *i.e.,* we need 18 hours for a full pass. Our authoritative name server is implemented based on a common high-performance DNS library [243], which supports up to 20k pps.

7.4.2 Results

The subsequent results are based on an Internet-wide scan from April 20, 2021. Ongoing, more recent scans find the same results.

Detailed Comparison with Shadowserver. We find ≈536k transparent forwarders, identified by distinct IP addresses. Compared to Shadowserver [278], which does not detect transparent forwarders, this reveals ≈ 18% more ODNS components (compare Table 7.1).

It is worth noting that we identified, in sum, fewer recursive resolvers and recursive forwarders compared to Shadowserver, because we require responses to include both `A`-records, with the static control record being unaltered. Shadowserver requires only one correct `A` record. Omitting this step in our method leads to similar numbers than Shadowserver. To be robust against manipulation, we keep our more strict requirement and still detect more ODNS components in total due to consideration of transparent forwarders.

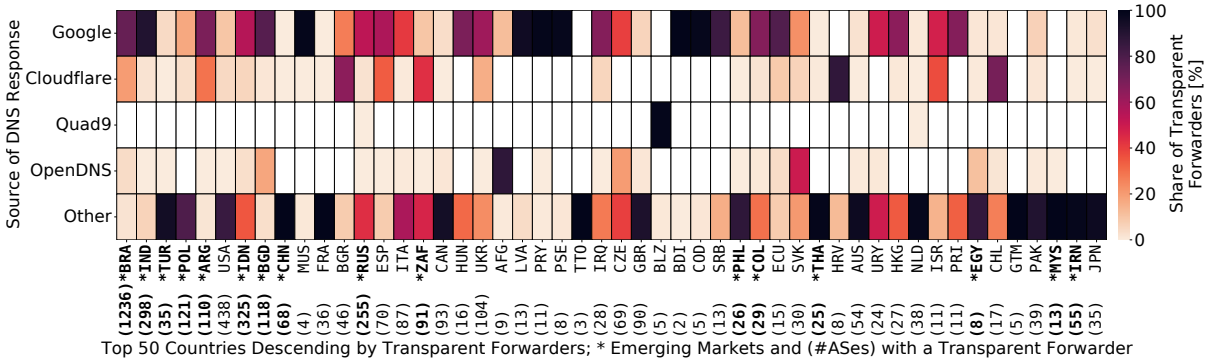


Figure 7.5: Popularity of public resolver projects. Google & Cloudflare are commonly used by transparent forwarders.

Geo-Location of Transparent Forwarders. We now try to understand whether the deployment of transparent forwarders is more popular in specific countries. To this end, we successfully map 99.9% IP addresses to ASes based on Routeviews dumps. Then, we map ASes to country codes with *whois* data und MaxMind. Figure 7.4 depicts the cumulative number of forwarders per country. Roughly 25% of countries with at least one ODNS component do not exhibit any transparent forwarder (highlighted in gray). We find, though, that ten countries host 90% of all transparent forwarders .

Countries that only expose transparent forwarders to the ODNS may be missed completely by scanning campaigns. Considering our complete data set, we do not find those cases. We find 5 countries hosting over 90% transparent forwarders, 4 of them are among the top-50 countries (see Figure 7.3). 8 out of 9 countries with over 10k transparent forwarders are classified as an emerging market [173], such as Brazil and India. With respect to all ODNSes in these two countries, transparent forwarders account for more than 80%.

Common Public Resolvers used by Transparent Forwarders. DNS consolidation directly correlates with how difficult it is to detect transparent forwarders. This is because the higher the consolidation, the more forwarders are *hidden* behind individual resolver IP addresses. Hence, we analyze the used resolvers and assess the relative popularity of four large public resolver projects (Google, Cloudflare, Quad9, and OpenDNS) per country. Figure 7.5 unveils that Google and Cloudflare are most common. Almost all transparent forwarders in India relay requests to Google, for example. This aligns with recent complementary studies, which show that 19% of Google DNS users are located in India [201]. Following these results we can conclude that current scanning campaigns, which only consider DNS replies, underestimate the amount of ODNS components per country since they observe responses only from `8.8.8.8` or other public DNS projects. Comparing Shadowserver and our data, the ODNS rank of the top-20 countries varies up to 12 positions (details see Subsection 7.7.1).

Table 7.4: Top-10 countries with highest "other" share (absolute) in Fig. 7.5. We show (*i*) the ASNs from which our scanner received most of the "other" responses, (*ii*) the number of transparent forwarders, (*iii*) the share of responses in "other" for which the ASN of $A_{resolver}$ belongs to one of the four common resolver projects.

Country	Top ASN	# Transparent Forwarders	Indirect Consolidation
Turkey	9121	52,663	0.3%
Poland	5617	24,879	1.4%
United States	209	14,546	18%
China	4812	11,030	0.9%
France	5410	5,268	0.8%
Indonesia	4622	5,154	27%
India	3356	5,037	48%
Brazil	262462	4,920	48%
Canada	21724	2,303	21%
Italy	3269	1,824	35%

Alternative Resolvers used by Transparent Forwarders. We find countries in which transparent forwarders do not use one of the four common resolver projects (see "other" in Figure 7.5). In order to understand the usage of alternative resolvers, we analyze the top-10 countries with most transparent forwarders in the "other" share. Our results are summarized in Table 7.4. We detect two trends. First, countries such as India and Italy that already use popular resolver projects frequently (direct DNS consolidation) also deploy complex forwarding chains. In those cases, at our scanner, we receive DNS responses from IP addresses belonging to the AS of the transparent forwarder. Analysing the IP address in the $A_{resolver}$ record reveals, however, that our authoritative name server received the request from an IP address outside this AS. This unveils a dependency chain in which transparent forwarders relay to local recursive forwarders, which then forward to a popular resolver project (indirect consolidation). Second, we identify countries (Poland, France, China, and Turkey) that tend to not use public resolvers at all. Here, we find larger forwarder pools but those forwarders use only 1 to 10 local resolvers. For example, a single DNS resolver (195.175.39.69, Turkish Telecom) is serving almost all transparent forwarders from Turkey, which again masks their existence (for stateless scans).

7.5 DNSRoute++

In this section, we introduce *DNSRoute++*, a tool to explore network properties around transparent forwarders, and present two results.

Measurement Approach. *DNSRoute++* is a traceroute application that exploits the behavior of transparent forwarders. In contrast to common `traceroute`, *DNSRoute++* sends DNS requests and continues incrementing the TTL when the target is reached. If the target IP address is a transparent forwarder, we expect to receive `TTL Exceeded` messages from hosts beyond the forwarder. In detail, *DNSRoute++* (*i*) reveals all hops between a scanner and the (target) transparent forwarder, then (*ii*) all hosts between the transparent forwarder and the recursive resolver used by the forwarder. This works because the IP stack of the transparent forwarder replies when the TTL is exceeded (which stops forwarding) and forwards a DNS request internally to the upper layers otherwise (which reveals hosts beyond a forwarder). We scan all transparent forwarders.

Path Lengths to Public Resolvers. We compare path lengths from transparent forwarders to their recursive resolvers, see Figure 7.6. We obtain over 70k paths to 1.1k ASNs after sanitization. Our sanitization removes incomplete paths due to host churn or traceroute anomalies. Short path lengths indicate sound anycast deployments.

We find that Cloudflare exhibits the shortest paths compared to Google and OpenDNS. On average, Cloudflare resolvers are reachable in 6.3 hops. In case of Google and OpenDNS, we observe 7.9 and 9.3 hops, respectively.

Doan *et al.* [35] performed similar path measurements using 2.5k RIPE Atlas probes in 729 distinct ASes. They also observe shorter paths to the Cloudflare resolver but a reverse ranking in case of Google and OpenDNS. This difference might be due to the location of measurement probes. RIPE Atlas probes are more likely located in North America and Europe, transparent forwarders are more common in South America and Asia. It is worth noting that our measurement approach only requires transparent forwarders and no deployment of dedicated probes in external networks. Hence, our methodology is complementary.

AS Relationship Inference. Paths acquired with *DNSRoute++* may help to infer AS relationships. The autonomous system (AS) before the AS of a forwarder indicates an inbound network (AS_{in}) and the AS after a forwarder the outbound network (AS_{out}). If $AS_{in} = AS_{out}$, we can assume a provider-customer relationship, since our scanner is outside the customer cone of AS_{in}. After sanitizing AS mappings, we can utilize 27k paths and observe $AS_{in} = AS_{out}$ for 62% of the paths. We detect 41 provider-customer relationships that are currently unclassified by CAIDAs relationship inference [171].

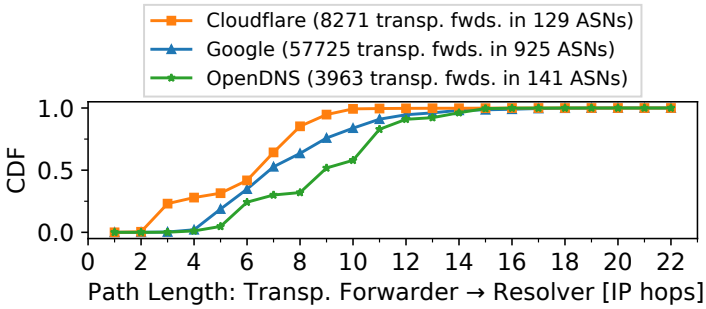


Figure 7.6: Distribution of path lengths between transparent forwarders and their recursive resolvers, separated by common recursive resolver projects.

7.6 Discussion

What is the purpose of transparent forwarders? Transparent forwarders differ from intentional DNS manipulations. First, they are not part of transparent interception [63], [87], [126], [288], which forwards queries to alternate resolvers and spoofs responses. Also, they differ from DNS redirection, which changes response records for the sake of advertisement [77], [156] or censorship [7], [80]. Lastly, they are not part of DNS tunneling, which carries ancillary information [61] not related to name resolution.

We conjecture that transparent forwarders are misbehaving CPE devices, either serving a single end customer or larger networks. To support this hypothesis, we perform an (*i*) AS-based, (*ii*) device-based, and (*iii*) prefix-based classification. For details about the classification, we refer to Subsection 7.7.3.

Considering the top-100 ASes by the number of transparent forwarders, we find 79% ASes are eyeball ISPs, 7% of other types, and 14% remain unclassified. 65 ASNs are 32-bit numbers [286], *i.e.,* belong to more recent AS deployments.

MikroTik produces low-cost routers and CPE devices which are often affected by vulnerabilities [168], [23] and have been previously identified as DNS forwarders [81]. MikroTik's price policy seems to attract countries with emerging markets [173]. Overall, we attribute about 18k hosts (23%) to MikroTik.

Finally, we find that 26% of transparent forwarders are located in a `/24` IP prefix that hosts less than 25 transparent forwarders. Such sparse population indicates that those forwarders belong to CPE devices (*e.g.,* home gateways) of individual end customers. On the other hand, we also find that 36% of the transparent forwarders cover a `/24` network completely. 50% of the MikroTik routers we identified can be assigned to such a scenario.

All these observations strongly indicate that most of the transparent forwarders are misconfigured CPE devices. Whether these devices serve as a middlebox for a single customer or

as router for a larger network does not affect our results regarding consolidation and attack potential.

Should scanning campaigns deploy transactional scans? Yes. Based on our measurements, current implementations of stateless DNS scans miss transparent forwarders, which account for 26% of all ODNS components. Interestingly, some countries host a disproportional amount of transparent forwarders which makes them far more exposed to misuse than previously assumed. For those 15 countries, we find that they host at least 50% of transparent forwarders and twice as much ODNSes as comparable studies detect.

Our transactional scans show that revealing transparent forwarders does not conflict with fast, stateless scans. Transactional scans require little-to-none changes to existing scanning infrastructures. Required changes include (*i*) the recording of outgoing scan traffic and (*ii*) a lightweight post-analysis, which matches queries and responses based on the client port and DNS transaction ID. These changes do not impair the scanning rate itself. We focus on DNS over UDP [245] as we do not expect transparent forwarding for DoT [199] and DoH [197] since their connection-based requests conflict with IP spoofing. Also, for benevolent scanning campaigns, we recommend utilizing custom responses and *not* custom queries for a forwarder classification to limit adverse effects. Encoding the IP addresses of targets leads to cache pollution due to negative caching [198] and cache evictions of popular, legitimately used names, which resembles random-subdomain [43] and water-torture [90] attacks. We find resolvers serving >40k forwarders, which would introduce >40k cache entries to a single resolver.

What is the misuse potential? Transparent forwarders can be misused as invisible diffusers for reflective amplification attacks as they relay the source IP address of the DNS request as-is. Hence, spoofed packets (allegedly from the victim) are forwarded with the source address spoofed by the attacker. Booters offering DDoS services utilize centralized attack infrastructures to reduce costs and maintenance [137]. Misusing transparent forwarders (*i*) allows to reach multiple PoPs of anycast DNS providers despite their centralized infrastructure (*e.g.,* Google allows `ANY` requests) and (*ii*) impedes attribution by further obfuscating the origin of spoofed traffic.

Overall, transparent forwarders likely belong to domestic setups but interact with unsolicited, external requests, which might lead to impaired performance, security risks and liability implications.

7.7 Additional Analysis

7.7.1 Ranking Countries by ODNS Components

In this work, we showed that transparent forwarders amount to more than 25% of all ODNS components. Common ODNS scan campaigns such as Shadowserver rank countries based on the number of ODNS components but miss transparent forwarders (see Section 7.3). Table 7.5

Table 7.5: Top-20 countries ranked by number of ODNS components, comparing this work and Shadowserver.

	This Work		Shadowserver		DifferenceΔ	
Country	Rank	#ODNS	Rank	#ODNS	Rank	#ODNS
China	1	632428	1	717706	0 -	85278 ↓
Brazil	2	297828	6	49616	4 ↑	248212 ↑
United States	3	144568	2	137619	1 ↓	6949 ↑
India	4	102910	8	33510	4 ↑	69400 ↑
Russia	5	93498	3	102368	2 ↓	8870 ↓
Turkey	6	76168	18	19298	12 ↑	56870 ↑
Indonesia	7	59972	5	56319	2 ↓	3653 ↑
South Korea	8	49143	4	73790	4 ↓	24647 ↓
Argentina	9	43648	20	16974	11 ↑	26674 ↑
Poland	10	43431	10	29175	0 -	14256 ↑
Bangladesh	11	40917	16	22940	5 ↑	17977 ↑
Taiwan	12	37550	7	38525	5 ↓	975 ↓
Iran	13	36659	9	33444	4 ↓	3215 ↑
France	14	25320	12	25763	2 ↓	443 ↓
Italy	15	24766	14	24483	1 ↓	283 ↑
Vietnam	16	21407	15	24266	1 ↓	2859 ↓
Ukraine	17	20780	13	25307	4 ↓	4527 ↓
Thailand	18	19694	17	20474	1 ↓	780 ↓
Bulgaria	19	18443	n/a	16239	>1 ↑	2204 ↑
Germany	20	16243	19	17788	1 ↓	1545 ↓

shows the change of ranks for the top-20 countries when considering the complete ODNS infrastructure by including transparent forwarders.

7.7.2 Measuring Transparent Forwarders

Figure 7.7 illustrates our response-based measurement method, which we explain in detail in Section 7.4.

7.7.3 Details on the Deployment of Transparent Forwarders

AS Classification. We classify the top-100 ASes by transparent forwarder count. These ASes cover 50% of all transparent forwarders. For each ASN, we map the network type using PeeringDB. 37 ASes are `Cable/DSL/ISP` networks. Since the majority of ASes is not classified in PeeringDB, we also perform a manual classification. We also perform a manual check whether ASes that are classified as `NSP` provide eyeball Internet services. Based on our manual inspec-

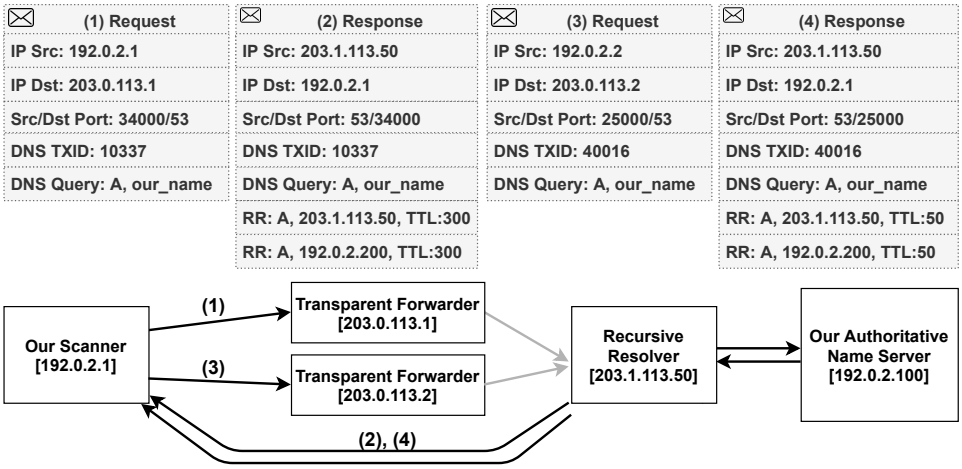


Figure 7.7: Two transparent forwarders trigger DNS responses from the same recursive resolver, identified by the same source IP address. Black arrows indicate DNS messages observed by our infrastructure.

tion, we identify 42 additional ISPs. In total, out of the top-100 ASes, 79 can be considered `Cable/DSL/ISP` networks.

Device Fingerprinting. For device fingerprinting, we use Shodan [273] and Censys [174]. To this end, we analyze all open ports and banner grabbing information. Shodan provides information for 80k of 600k queried hosts. Inspecting the open port distribution, we find a strong correlation for 10 MikroTik ports [23]. OS and product information collected by Shodan confirm our observations becasuse the most common tags specify MikroTik. Censys data confirms our results and also identifies the hosts as MikroTik devices.

Distribution in `/24` Prefixes. We map each transparent forwarder to a (non-overlapping) covering `/24` IP prefix and count the number of forwarders per prefix. If all IP addresses of a prefix reply to our transparent forwarder scans, we may assume that these replies are initiated by a single device (*e.g.,* some kind of middlebox that serves the whole prefix). In contrast, for sparsely populated prefixes, we may assume multiple deployed devices (*e.g.,* several CPE devices that serve differnet customers).

41k distinct IP prefixes cover 0.6M transparent forwarders. Figure 7.8 shows the distribution of the number of transparent forwarders in each `/24` prefix. Overall, we observe a mixed picture. 26% of all transparent forwarders are located in sparsely populated prefixes (≤ 25 transparent forwarders in a `/24` prefix), and 36% in completely populated prefixes (≥ 254 transparent forwarders in a `/24` prefix). Only 806 prefixes are completely populated. In those cases, we argue that a CPE device serves as a router for larger networks instead of a single end customer. Using CPE devices outside of individual DSL/cable customers is not uncommon because CPE devices are cheap and some implement routing protococols (*e.g.,* MikroTik even BGP). In any

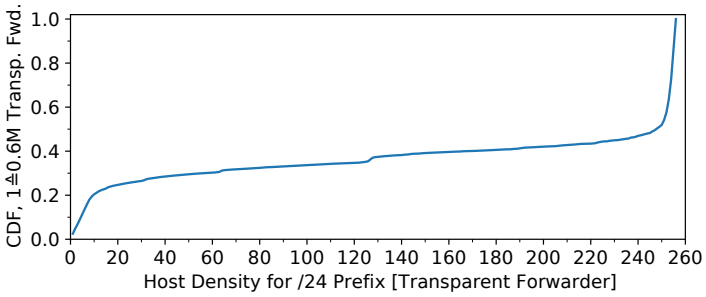


Figure 7.8: We map all transparent forwarders to a covering `/24` prefix. Some transparent forwarders belong to individual end customers, others may serve a larger network.

case, whether the transparent forwarder function runs on a device that serves a single end customer or a larger network, our results hold, the transparent forwarder interacts as an ODNS component and uses the resolvers we observed.

7.8 Conclusion

We showed that the open DNS infrastructure comprises transparent forwarders—in addition to its recursive components. These forwarders intensify the perceived threat potential of the ODNS. We argue to include them in on-going and future measurements as they account for a relevant impact and share. Our results bolster current concerns regarding consolidation of the DNS, at least for countries that massively host transparent forwarders.

7.9 Ethical Considerations

We presented a method to discover a new type of public DNS forwarders, which may be misused by attackers. We are in contact with federal security offices to include transparent forwarders in their on-going measurements that inform network operators about vulnerable devices.

7.10 Artifacts

This section gives a brief overview of the artifacts of this chapter. We contribute tools to conduct follow-up measurements as well as raw data and analysis scripts to reproduce the results and figures presented in this chapter.

Hosting. All artifacts are available in the following repository:

`https://github.com/ilabrg/artifacts-conext21-dns-fwd`

This public repository provides up-to-date instructions for installing, configuring, and running our artifacts. We also archive the camera-ready version of our software on Zenodo:

https://doi.org/10.5281/zenodo.5636314

Future Measurements. We plan to continue our experiments. Future measurement results will be available on https://odns.secnow.net.

Chapter 8

On the Interplay between TLS Certificates and QUIC Performance

Abstract

In this chapter, we revisit the performance of the QUIC connection setup and relate the design choices for fast and secure connections to common Web deployments. We analyze over 1M Web domains with 272k QUIC-enabled services and find two worrying results. First, current practices of creating, providing, and fetching Web certificates undermine reduced round trip times during the connection setup since sizes of 35% of server certificates exceed the amplification limit. Second, non-standard server implementations lead to larger amplification factors than QUIC permits, which increase even further in IP spoofing scenarios. We present guidance for all involved stakeholders to improve the situation.

8.1 Introduction

The QUIC protocol [203] was designed to improve Web performance and reduce access latency [29], [143] while keeping communication confidential [32]. A key approach is the reduction of initial roundtrip times by integrating the QUIC handshake with the TLS 1.3 handshake and coalescing multiple QUIC packets into one UDP datagram. At the same time, security concerns about the UDP-based QUIC protocol demanded to limit the amplification potential, *i.e.,* the

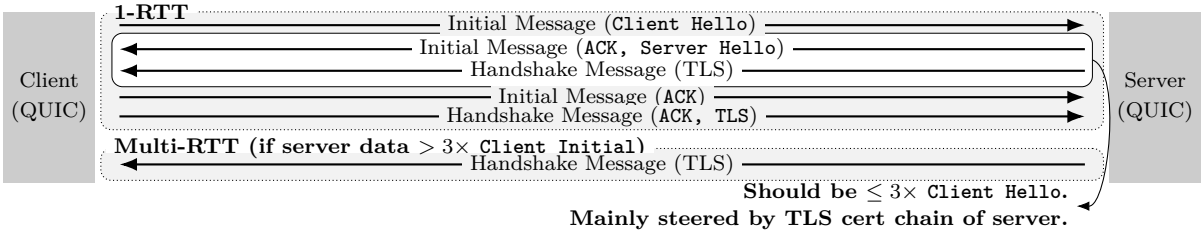


Figure 8.1: In QUIC handshakes, server replies are limited to 3× the size of the client `Initial` until the client is verified.

byte ratio of the server answer to the client request, for the initial reply to an (unauthenticated) client.

In QUIC, the details of a connection handshake depend on a variety of factors: version negotiation, QUIC retry option, packet coalescing, and the size and compression of TLS certificates. In this work, we focus on the latter because it has great relevance for the handshake process, see Figure 8.1. Using active and passive measurements we observe significant effects on latency, amplification, and protocol behavior in current deployments.

In detail, we contribute the following.

1. Background on the interplay between the QUIC handshake and TLS certificates and prior work (Section 8.2).
2. A measurement method to systematically analyze the problem space (Section 8.3).
3. Analysis of QUIC server behaviors for different sizes of client `Initial` messages. The majority of QUIC servers incorrectly amplify handshakes or require multiple RTTs, even for common `Initial` sizes used by Web browsers (Subsection 8.4.1).
4. An in-depth study of QUIC handshake behavior that shows that multi-RTT handshakes are caused by large certificates and missing packet coalescence. Furthermore, some certificates unnecessarily contain cross-signed certificates instead of self-signed versions or include their trust anchors (Subsection 8.4.2).
5. Empirical results highlighting the benefits of certificate compression during the handshake. 99% of certificate chains would remain below the allowed amplification factor (Subsection 8.4.2).
6. A major reason why large amplification factors may appear during connection setups in the wild. For large CDNs, we observe up to 45× amplification for spoofed handshakes (Subsection 8.4.3).
7. Guidance to improve the situation (Section 8.5) and our artifacts, which are publicly available (Section 8.8).

8.2 Background & Related Work

In this section, we briefly recap the QUIC protocol mechanics of the connection setup, introduce challenges of TLS certificates that undermine fast setups, and discuss related work.

The QUIC handshake and amplification mitigation. QUIC [203] was designed to provide low latency, reliability, and security on top of UDP. A crucial part is the initial connection setup, which should be fast [157] and prevent attacks related to amplification [133], [266] or state exhaustion [110]. For this purpose, the QUIC handshake integrates TLS within the protocol

handshake. A client starts with an `Initial` that is answered by an `Initial` from the server and a `Handshake` packet that can be sent in a single UDP datagram (*packet coalescence*). The client confirms receipt with an `Initial ACK` and then sends an its `Handshake` message. The server resends unconfirmed `Initial` and `Handshake` packets. Protection against state exhaustion is achieved when a server uses `RETRY` packets but such protection is rarely deployed [110].

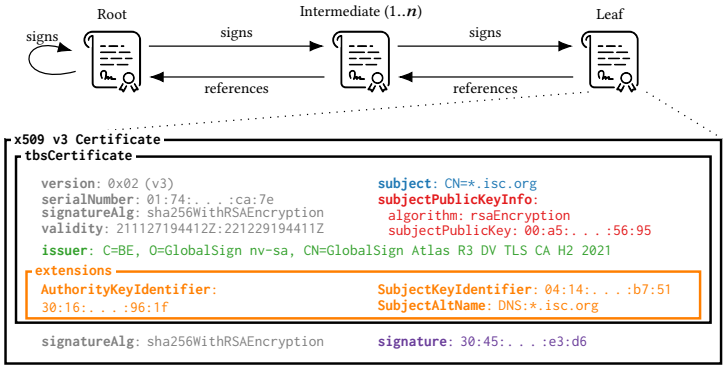
To prevent amplification attacks, a server must not reply with more bytes than the QUIC *anti-amplification factor* allows until the client IP address is verified. RFC 9000 [203] limits the data size from the server to 3× the bytes that have been received in the client `Initial`, see Figure 8.1, and includes padding and resent bytes [202]. After the server validates the client by a complete roundtrip, it is free to send any amount of data. The factor of three is low compared to the amplification potential of other protocols [133], [266]. We recap the IETF design of the threshold in more detail in Subsection 8.6.2.

QUIC TLS connection setup. QUIC integrates TLS 1.3 [263] to cater for authenticated confidentiality and integrity [281]. A TLS 1.3 handshake is initiated by a client sending its supported cipher suites, key parameters, and other metadata in the first `Initial` message to the server, which in turn replies with its own parameters and an X.509 certificate [178] used to authenticate its identity [281, §4.4.].

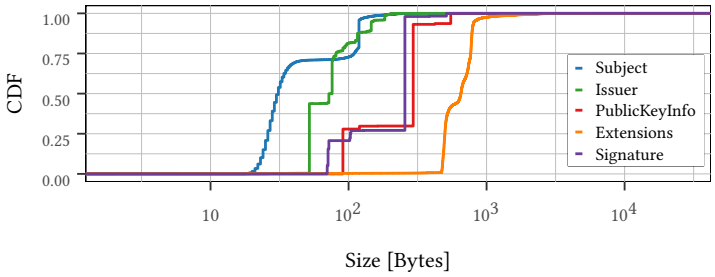
In contrast to TLS over TCP, the sum of *first responses* of a QUIC server must not be larger than the anti-amplification factor. This limit reduces the amplification attack surface but poses a challenge for benign QUIC peers to achieve the goal of low latency and low connection overhead. Either the client sends an `Initial` that is large enough to allow the server to accommodate its reply within the anti-amplification limit, or the server adapts responses to be small enough. Please note that the size of a server reply is mainly determined by its certificate, *i.e.,* the complete certificate chain sent by the server. Subfigure 8.2(a) illustrates the structure of a TLS certificate.

Popular browsers use between 1250 and 1357 bytes in the `Initial` message, which can easily conflict with common sizes of server certificate (see Section 8.4 for details). Depending on public key and signature algorithms in use, sizes of issuer and subject names, as well as extensions (*e.g.,* subject alternative names), the total size of a certificate may vary by an order of magnitude. Subfigure 8.2(b) depicts the size distribution from our data corpus; certificate extension fields followed by signature and public key fields are the most space consuming in certificates. A server can apply optimizations to its own certificate sizes, but it has no control over intermediate certificates that it delivers as part of the certificate chain of trust. To compress the entire chain, TLS 1.3 provides certificate compression [192]. To take effect, client and server must support compression. While the adoption of TLS 1.3 is well studied [56], analysis of certificate compression deployment is not included in prior research.

QUIC performance and adoption. QUIC provides good performance [17], [21], [28], [67] and can outperform TCP. Prior work suggests that some security trade-offs were specifically



(a) X.509 certificate



(b) Size distribution

Figure 8.2: Example of a TLS certificate and our observed distribution for various X.509 certificate field sizes.

made in favor of improved latency [91]. The handshake, however, can suffer from additional latency if client and server do not agree on a version directly [46].

Prior deployment studies mainly focus on the availability of QUIC services. QUIC adoption started before the finalization of the standard [94], [144], [149] and continues since then [163], led by hypergiants [247], [135]. DNS over QUIC lacks wide adoption and exhibits inefficient handshakes due to large certificates if `Session Resumption` is not used [74], [75]. Most closely related is [255], showing that 40% of QUIC handshakes with uncompressed certificates may trigger an additional roundtrip, based on data from a specific CDN. To the best of our knowledge, this work is the first that systematically assesses the impact of TLS certificates on QUIC performance, leveraging comprehensive measurements.

8.3 Measurement Method and Setup

We search for (*i*) common HTTPS services and (*ii*) QUIC-services, to collect related TLS certificates and compare performance of protocol design and deployment choices. In this chapter, we use the term *service* to quantify the number of domains served via a specific protocol, irrespectively of whether these domains are delivered by the same IP host, *i.e.,* we present a domain-centric perspective. The point of departure for our scans is the Tranco list [122] generated on September 10, 2022, since the Tranco list provides a good compromise [138] between reflecting popularity and robustness. Subsequently[1], we scan 1M domain names to broadly capture what clients receive when contacting a web domain.

When conducting our measurements, we leverage existing tools where possible and minimize extensions to achieve maintainability and ease reproducibility. Unfortunately, there is no single tool available that implements all necessary features. We present an overview of our toolchain in Section 8.8, Figure 8.14.

8.3.1 TLS Certificate Scans via HTTPS

Not all names in the Tranco list resolve to web servers that allow for TLS over TCP connections. For each name in the list, first, we try to resolve IPv4 addresses using Google public resolver `8.8.8.8`. Upon success, we then try to establish HTTP connections on ports 80 and 443 and follow any redirects using HTTP(S) (status code 3xx) and HTML (`meta` tag with `http-equiv` attribute). For every secure domain, including all redirects, we collect TLS certificates.

We were able to resolve 976k (out of 1M). For 13k names, the domain query returned `SERVFAIL`, 9k could not be resolved (`NXDOMAIN`), and the remaining either timed out (10s) or refused the answer (`REFUSED` [179], [245]). About 866k names returned an IP address (`A`

[1] We immediately trigger our twofold scans (TLS/HTTPS scan followed by a QUIC scan) after the publication and retrieval of the Tranco list. Facing the trade-off between aggressive, likely more precise scans, *e.g.,* due to (*i*) TLS certificate rollovers or (*ii*) traffic engineering, and friendly, low-impact scanning rates, we decide upon rates that finished both scans within the same working week, representing our main measurement period.

record). For every domain name which resolved to an IP address we tried to establish an HTTP connection on both ports 80 (HTTP) and 443 (HTTPS). After following redirects, we collected 821k unique certificates for more than 1.1M names along the redirection path.

8.3.2 QUIC Scans

We analyze QUIC handshakes in two scenarios: (*i*) a complete handshake including client verification and (*ii*) an incomplete handshake imitating an unverified client, *e.g.,* when clients spoof IP addresses.

Complete handshakes. We scan all domains discovered during our certificate collection via HTTPS and assign each successful handshake to one of the four groups:

1. **1-RTT (optimal):** Handshakes that complete within 1-RTT and comply with the anti-amplification limit.

2. **RETRY (less efficient):** Handshakes that require multiple RTTs because the `Retry` option is used [203, §8.1.].

3. **Multi-RTT (unnecessary):** Handshakes that do not use `Retry` but require multiple RTTs because of large certificates.

4. **Amplification (not RFC-compliant):** Handshakes that complete within 1-RTT but exceed the anti-amplification limit.

To conduct QUIC handshakes and assign groups, we use `quicreach` [241], extended by `RETRY` support. We find 272k QUIC services (∼25%). To investigate the effect of client `Initial` sizes on server handshake behavior we vary the client `Initial` size between 1200 bytes (mandated minimum [203]) and 1472 bytes (dictated by our MTU since QUIC forbids fragmentation) in steps of 10 bytes. Handshakes targeting the same domain service pause 30 minutes to avoid side effects such as DDoS mitigation.

`quicreach` does not provide access to certificates nor does the underlying stack support certificate compression. We rescan with (*i*) `QScanner` [284] to access TLS certificates sent over QUIC and (*ii*) extend `quiche` [196] to support three popular TLS compression algorithms in QUIC.

In the majority of cases (96.7%), we find that the same TLS certificates are used in both QUIC and HTTPS deployments for the same domains, which confirms prior work [163]. For the remaining 3.3% of QUIC services, certificates differ from TLS over TCP. These differences are mainly due to certificate rotations during the period of time between our HTTPS and QUIC scans, leaving only 0.47% of QUIC services with different certificates because of other

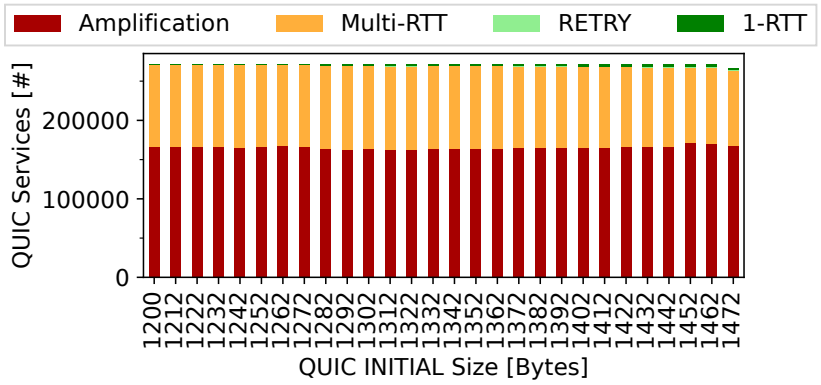


Figure 8.3: Influence of QUIC `Initial` sizes on the QUIC handshake. With respect to all names, we find almost no effect.

reasons. To sanitize inconsistent data, we decide to base our QUIC certificate analysis on the TLS certificates gathered via HTTPS.

Incomplete handshakes. To analyse the performance when a client successfully initiates but does not complete a handshake, *e.g.,* because of malicious activities, we conduct two measurements. First, we collect QUIC backscatter from a telescope during January 2022. Since telescopes do not emit any traffic, we can observe server behavior to non-responding, spoofed client IP addresses. Here, we group QUIC traffic by major content providers and source connection IDs (SCIDs). Second, we send a single `Initial` of 1252 bytes to the servers without sending `ACK` messages, using ZMap [38].

8.4 Results

In this section, we (*i*) present our analysis of complete handshakes, (*ii*) study TLS certificates as potential reason for performance drawbacks in more detail, and (*iii*) show results that reveal QUIC amplification potentials in the wild.

8.4.1 Classifying QUIC Handshakes

Overview. Figure 8.3 shows the absolute number of handshakes types for all QUIC-reachable names, depending on the `Initial` size. For an `Initial` size of 1362 bytes, which is similar to common browser default values (see Table 8.1), we find that 61% of handshakes are classified as amplifying and 38% as requiring multiple RTTs. Worryingly, the `Retry` and 1-RTT handshakes account for only 0.07% and 0.75%, respectively. This means that a priori DoS protection and fast handshakes are rare, unveiling that the QUIC design goals have not been met in the wild, yet.

Table 8.1: Comparison of QUIC INITIAL packet sizes and support for TLS 1.3 certificate compression in popular browsers.

		Init. Size	Compression		
Browser	Version	[Bytes]	Algorithm[3]	Rate[4]	Services[5]
Firefox	`101.x`	1357	–	–	–
Chromium-based[1]	`105.x`	1250[2]	brotli	73%	96%
Safari (macOS)	`15.5`	no QUIC	`zlib`	74%	0.05%
			`zstd`	72%	0.05%

[1] Chrome `102.x`, Brave `V1.39`, Vivaldi `5.3.x`, Edge `102.x`, Opera `88.0.x`.
[2] Recently reduced from 1350 [194]. [3] Tested with TLS 1.3 in TCP.
[4] Mean rate observed by our Quiche client. [5] Out of 272k QUIC services.

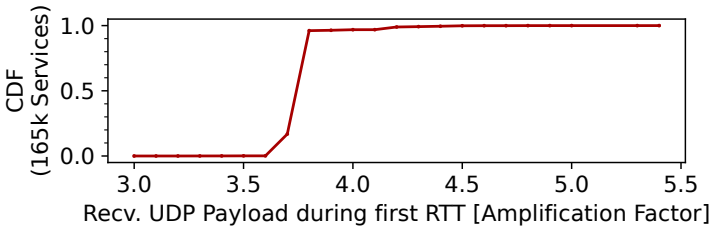


Figure 8.4: Amplification factor during first RTT. For complete client handshakes, the amplification is relatively small.

We now investigate the effect of different `Initial` sizes. We find that amplifying handshakes occur independently of the `Initial` size. However, we observe an interdependence between multi-RTT and 1-RTT handshakes. With larger `Initials`, multi-RTT handshakes are less likely and 1-RTT handshakes more likely (de- and increase by ∼1%). This nicely illustrates the performance impact of the interplay between `Initial` sizes and deployed certificate sizes.

We also observe that the reachability of QUIC services is reduced by 1.2% for large `Initials`, as indicated by the decreased height of the stacked bars. Interestingly, this effect is more pronounced for top-ranked services (not shown). The top 1k and top 10k domains are seeing a 25% and 12% drop in reachability, respectively. We argue that this corresponds to load-balancer deployments that are more likely to be used for very popular names. Load-balancers utilize packet tunneling to distribute the load across multiple, redundant server instances. Packet encapsulation used during tunneling adds bytes due to additional headers, which then exceed the local MTU. Our observations of reachability issues comply with prior measurements [83]. Other than that, we find little differences across ranks, compare Subsection 8.6.3.

1-RTT exceeding anti-amplification limit. Independently of the `Initial` size, the majority of handshakes exceed the anti-amplification limit in the first RTT. We calculate the amplification factor for our default `INITIAL` scans of 1362 bytes by dividing UDP payload

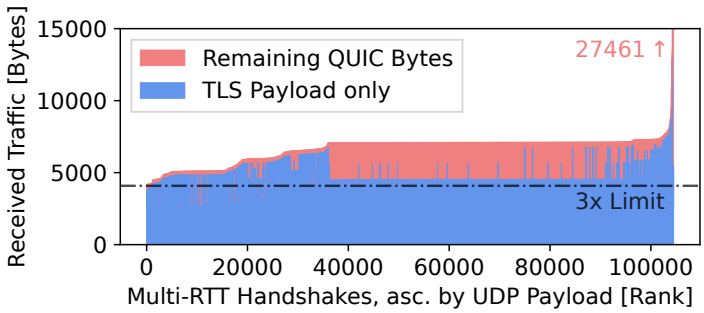


Figure 8.5: Payload exchanged during multi RTT handshakes. TLS bytes almost always exceed the limit but also QUIC padding can have a significant impact.

bytes received by the UDP payload bytes sent by the client. Figure 8.4 shows the amplification distribution. The amplification factor, although exceeded, remains relatively small below `6x`.

Cloudflares missing coalescence explains amplification. Based on TLS information and additional IP prefix mapping, we find that 96% of the amplifying handshakes are completed with Cloudflare servers and subject to the same implementation behavior. Surprisingly, we observe exactly 2462 superfluous QUIC padding bytes for ≈157k handshakes. In these cases, although the TLS data can vary in size, the remaining QUIC bytes are constant in size. Cloudflare servers do not support packet coalescence at two levels: (*i*) `Initial` flags are sent separately, leading to two UDP datagrams. The first containing the `ACK` and the second the `ServerHello` flag, both of which are padded resulting in 2462 extra bytes, although only the latter elicits `ACKs` and thus requires padding. (*ii*) We do not observe any coalescence of `Initial` and `Handshake` messages. The extra bytes account for ≈60% of the limit but are (incorrectly) not considered during amplification limit checks. We report this implementation shortcoming to Cloudflare.

`Retry`. We observe ≈200 services that predominantly request a `Retry` to authenticate client addresses. We conclude that always-on DDoS mitigation is currently not widely adopted, however, `Retrys` might also be triggered adaptively based on the current server load.

Multi-RTT (no `Retry`). Due to rare deployment of *always-on* `Retry` messages, we assume that multi-RTT handshakes are caused by other factors. We analyze these factors, *i.e.,* TLS certificates, in more detail in the next section.

8.4.2 Impact of TLS Certificates

We presume that TLS certificate data causes multi-RTT handshakes. To verify our assumption, we divide the bytes exchanged during a handshake into TLS payload and QUIC-related payload, *e.g.,* QUIC header and padding.

In the majority of cases (87%), TLS payloads alone exceed the amplification limit (see Figure 8.5). The distribution of (uncompressed) certificate chain sizes exchanged over TLS is

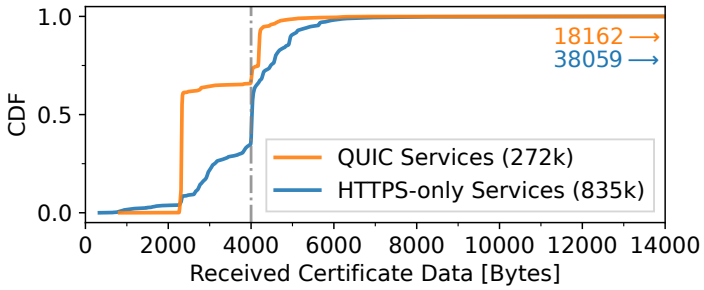
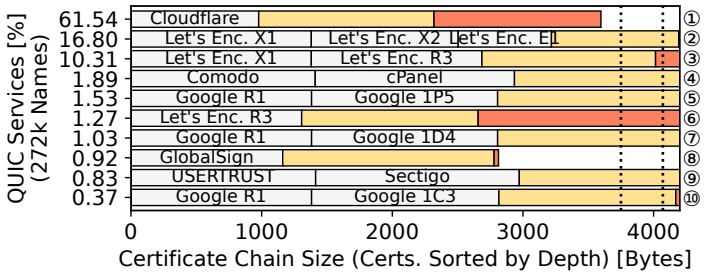
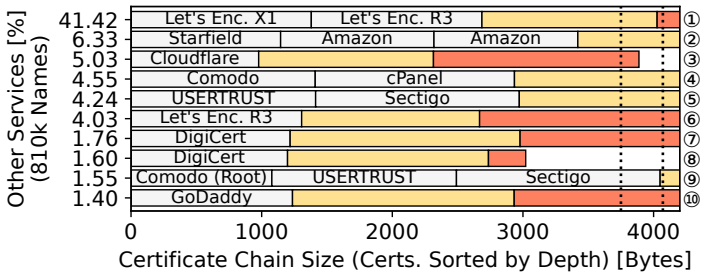


Figure 8.6: Distribution of certificate sizes grouped by QUIC support. QUIC domains use smaller certificates.



(a) QUIC services



(b) Only HTTPS services

Figure 8.7: Certificate chain sizes, depths and their dependency. ■ represents median leaf size, and ■ the additional bytes required for the maximum leaf size. Dotted lines represent the max allowed reply sizes of a server given common client `Initial` sizes. The x-axis is cut off at 4200 bytes. Average sized certificate chains are likely to exceed QUIC amplification limits.

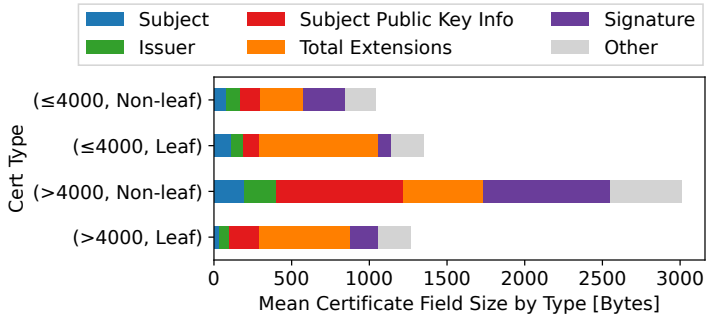


Figure 8.8: Mean sizes of certificate fields for QUIC domains. Non-leafs contribute most bytes to large chains.

shown in Figure 8.6. Overall, we observe a median of 2329 bytes for certificate chains delivered by QUIC domains compared to 4022 bytes for other names. We find that 35% of all certificate chains exceed even the larger of the two common amplification limits (3·1357 bytes). This means that domains without QUIC support will be affected negatively when they decide to support QUIC in the future and continue to use existing certificates.

Popular parent certificates for QUIC unveil consolidation. By zooming into certificate chains, we examine how the choice of a specific CA can impact the size of the certificate chain that a service provider needs to deploy. For this analysis, we exclude certificate chains that are not ordered correctly. Figure 8.7 exhibits the top-10 certificate chains deployed. Each white box represents the sizes of the certificates in the chain (excluding leaf certificates), yellow boxes (■) and orange boxes (■) represent the median sizes and the largest leaf certificate that we observed in that chain.

Overall, we find that 7 out of 10 parent chains, together with the median leaf size, exceed common amplification limits (5 out of 10 for HTTPS-only services).

For both QUIC and non-QUIC services the shortest chains, *i.e.,* the smallest number of intermediates, are issued by Cloudflare followed by Let's Encrypt R3, GlobalSign, DigiCert, and GoDaddy. We also observe cases in which cross-signed certificates are redundantly included in chains while the self-signed version of the same public key is already included in client trust stores. For example, row ② and ③ in Subfigure 8.7(a) include the cross signed version [220], [269] of `ISRG Root X1` (signed by `DST Root CA X3`) instead of relying on the self-signed variant [220], [270], as in row ⑥. In other cases, servers superfluously include trust anchors (*i.e.,* root) certificates (*e.g.,* row ⑨ in Subfigure 8.7(b)).

Furthermore, a high consolidation trend for QUIC services is visible, as the top-10 parent chains cover 96.5% of QUIC services. For HTTPS-only services, this trend is less pronounced with only 72% of services. Consequently, to improve the deployment of QUIC services, optimizing the parent chains can have a significant, beneficial effect but only needs to involve a

Table 8.2: Relative ratio of crypto algorithms and key lengths [bits] in use (limited to types with a frequency of > 1%). HTTPS-only domains depend heavily on RSA.

		RSA		ECDSA	
Service	Certificate	2048	4096	256	384
QUIC	Non-leaf	15.1%	22.4%	40.4%	22.1%
	Leaf	19.2%	1.4%	78.9%	0.0%
HTTPS-only	Non-leaf	63.3%	32.1%	2.7%	1.6%
	Leaf	81.4%	8.1%	7.8%	1.9%

small number of stakeholders. Certificates delivered by QUIC servers tend to use more efficient crypto algorithms, though, compared to non-QUIC Web services (see Table 8.2).

Non-leaf certificates bring the heavy load. We find very large certificate chains requiring transmissions between 18k and 38k bytes, indicated by the long tail above 4000 bytes in Figure 8.6. We proceed to use this value as a threshold to classify certificate chains.

Figure 8.8 depicts the mean size of various TLS certificate fields divided into leaf and non-leaf certificates. We observe that for large chains the sum of public key and signature sections on non-leaf certificates has the biggest impact on the chain size. This again shows the negative effects of selecting a large non-leaf parent chain, even if the related leaf certificate has a reasonable size.

We also find that large cruise-liner leaf certificates [20] are rarely used in QUIC deployments, details see Subsection 8.6.4.

Compression helps. Compressing certificate chains can avoid exceeding anti-amplification limits and thus improve the situation in the future. Our synthetic experiment of compressing collected certificate chains shows a median compression rate of ≈65%. This keeps the size under the amplification limits for 99% of TLS chains, which in turn prevents multi-RTT QUIC handshakes.

We find that 96% of QUIC services currently support the brotli algorithm, which is used by Chromium derivatives. The support of multiple algorithms, however, is very rare with only 0.05% of QUIC services offering all three. These services relate to Meta.

The mean compression ratio in the wild is 73%, which is close to our synthetic experiments. Here, 99% of all compressed certificates fit below a common anti-amplification limit (3·1357 bytes).

8.4.3 Examining Amplification Potential

Our previous analysis considered the behavior of servers when handshakes complete successfully. Now, we consider the case when a client fails to send an `ACK` to a server response. This would cause a resend of data by the server. Since the client IP address is not verified all resends must

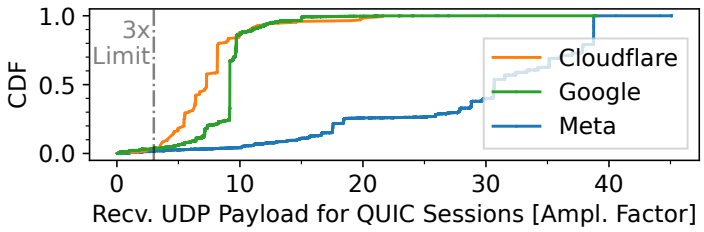


Figure 8.9: QUIC amplification factors including resends when clients do not respond (*e.g.,* due to spoofing). Amplification increases drastically.

comply in sum with the 3× amplification limit. A resend occurs, for example, when malicious actors initiate a handshake with a spoofed IP address. All resends, *i.e.,* amplified traffic, are reflected to the spoofed address belonging to a victim.

Figure 8.9 depicts amplification factors of handshakes collected at our telescope vantage point. We sum all bytes received from a server for a specific SCID, and divide by an assumed client `Initial` of 1362 bytes. All hypergiants exceed the amplification limit due to resends. The majority of Cloudflare and Google backscatter traffic remains below factors of 10×. Worryingly, traffic from Meta servers lead to amplification factors of up to 45×. As a crosscheck, we inspect the duration of backscatter sessions for Meta. We find a median of ∼51s and a maximum of 206s. This indicates that the amplified traffic is received within a short time frame and the observed amplification factors are not biased by *e.g.,* reused, overlapping SCIDs.

To confirm that Meta servers do not comply with the current QUIC specification [203], we conduct active scans as follows. We send a single QUIC `Initial` but do not acknowledge the response. We focus on the `/24` subnet of a Meta point-of-presence and identify three groups of IP addresses:

1. No response or ≤150 bytes, due to no QUIC HTTP3 service.

2. Responses of ≈7k bytes, which corresponds to an amplification factor of over 5×. IP addresses that typically serve `facebook.com` (`*.35, *.36`) belong to this group.

3. Responses of ≈35k bytes, which corresponds to an amplification factor of over 28×. This amplification factor is similar to what can be achieved using popular amplification protocols [133]. We find IP addresses that relate to Instagram and WhatsApp (`*.60, *.63`) belonging to this group.

Overall, our active scans confirm the telescope observations. Current deployments of Metas QUIC implementation `mvfst` [187] do not respect the 3× limit in case of resends. Those deployments can be misused as amplifiers in attacks.

8.5 Discussion & Guidance

Should the QUIC protocol specification be updated? Our results suggest that the QUIC anti-amplification limit specified in RFC 9000 [203] is indeed tight but large enough to achieve 1-RTT handshakes. The limit does not need to be increased to foster better deployments when network conditions are reliable. In the case of packet loss and necessary resends, the anti-amplification limit challenges performance, though. It allows for at most one single retransmission of all server `Initial` and `Handshake` messages, given current certificate deployments including small ECDSA certificate chains and certificate compression. Dealing efficiently with loss of messages during the connection setup seems an open challenge.

Next to protocol design challenges, we also find non-standard QUIC implementations that amplify during the 1-RTT handshake and increase significantly for incomplete QUIC handshakes. More comprehensive testing of QUIC implementations is clearly needed.

Does certificate compression help? We found that certificate compression is an impactful extension to allow servers staying below the amplification limit. Unfortunately, popular TLS implementations such as OpenSSL do not support certificate compression. Given that recent QUIC implementations (*e.g.,* Microsoft QUIC) depend on existing TLS libraries, compression may remain in far reach and alternate measures are required to improve the situation.

Can a QUIC client mitigate lack of compression? To be independent of certificate compression, a QUIC client could maintain a cache that includes certificate sizes of servers that the client frequently requests. For entries in the cache, the client can then adapt the size of `Initial` requests to comply with the anti-amplification limit of the servers and achieve low latency connection setup.

Guidance for certificate authorities. We argue that carefully created TLS certificates and certificate chains can positively influence the QUIC protocol performance. ECDSA certificates lead to substantially smaller certificates chains. They can, however, not unfold their potential because especially root certificates are secured by RSA algorithms. Our results show that updating these certificates can have beneficial cascading effects.

Guidance for QUIC implementations. We infer the following guidelines when implementing QUIC network stacks: First, at the server side implementation, bytes that result from padding or `Resend` must be included in anti-amplification limit checks. Second, enabling packet coalescence at the server is recommended to omit padding and thus free space for TLS certificates reducing the need for additional round trips. However, this can increase the latency when large-scale deployments deliver certificates by servers others than those providing content. Third, we recommend the integration of a TLS library that supports compression to compensate large TLS certificates, which currently trigger multi-RTT handshakes.

8.6 Responsible Disclosure and Additional Analysis

8.6.1 Ethical Considerations

This work may raise the following ethical concerns.

Educating attackers. We discovered deployment behavior that conflicts with DDoS mitigation required by RFC 9000 and hence enables misuse. We follow a responsible disclosure policy and aim for fixing the bugs in collaboration with Cloudflare and Meta.

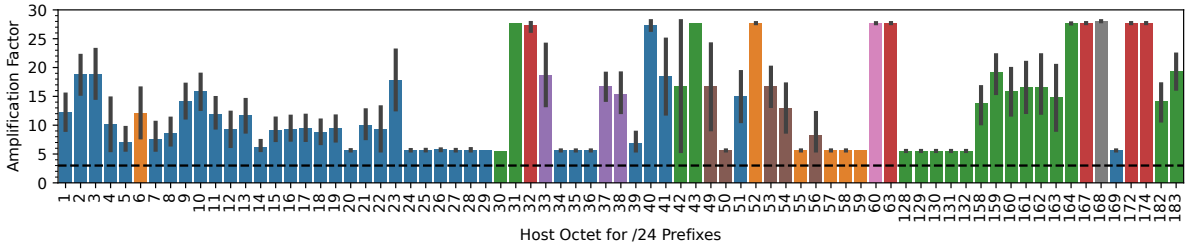
Responses from hypergiants. Meta significantly improved their QUIC deployment in October 2022. By rescanning all host addresses in `/24` on-net prefixes, we now observe homogeneously configured servers that limit the amount of QUIC retransmissions in case of unverified clients. However, with a mean amplification factor of 5×, the responses still exceed the anti-amplification limit specified in RFC 9000. We show the results in Figure 8.10, including 95% confidence intervals.

Cloudflare has responded, and explains the reason to exceed the limit is to help improve client performance, while respecting production constraints that are omitted from the QUIC specification. Specifically, in production environments the information needed to populate the `ServerHello` is contained in certificates that may be managed separately from connection termination, and unavailable at the moment of arrival of the client's `Initial`. The delay affects client estimates of RTT. Cloudflare mitigates the delay by immediately responding to client `Initials` with an `ACK` padded at the UDP layer. This occurs once, so the amplification factor is bound.

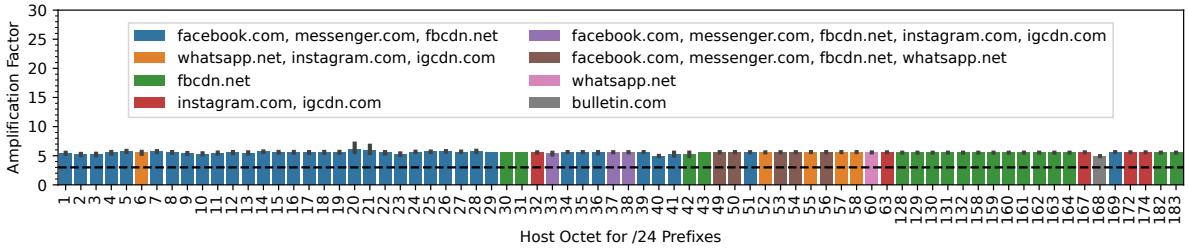
8.6.2 QUIC Anti-Amplification Limit

In Table 8.3, we show the historical development of QUIC amplification mitigation as proposed in the different versions of the QUIC Internet Draft. Although amplification attacks have already been mentioned in Draft 01 [208], no limitations to reduce the attack potential have been specified for servers. Draft 02 [209], at least, specifies that clients must ensure that the first packet in a connection, *i.e.,* commonly an `INITIAL`, meets the requirement of minimum packet size. This requirement limits the overall amplification factor since any attacker needs to invest a minimum amount of data.

In Draft 09 [204], the first restriction for servers is introduced. A server may close a connection with an error code in the case of a too small client `INITIAL`. Otherwise, it must not respond or behave as if any part of the offending packet was processed as valid. In Draft 10 [205], a server is limited by the number of `HANDSHAKE` packets a server is allowed to send to unverified clients, even though this is not explicitly noted in the context of amplification mitigation. Since Draft 15 [206], the anti-amplification limit is specified relative to the client. Since Draft 33 [207], including the current RFC [203], this limit has been specified to three times of received *data*.



(a) August 2022 (before disclosure)



(b) October 2022 (after disclosure)

Figure 8.10: Mean amplification factors for Meta services observed at all point-of-presences. We see a significant improvement after the responsible disclosure of our results. The anti-amplification limit is still slightly above the allowed threshold.

Table 8.3: Descriptions of amplification mitgation in the different versions of the IETF QUIC Internet Draft, leading to the 3× anti-amplification limit. [Bold highlighting by us.]

IETF Spec	Date	Proposed Limit
Draft 09	01/2018	"A server MAY send a CONNECTION_CLOSE frame with error code PROTOCOL_VIOLATION in response to an Initial packet smaller than 1200 octets."
Draft 10 – 12	03/2018 – 05/2018	"Servers MUST NOT send more than three **Handshake** packets without receiving a packet from a verified source address."
Draft 13 – 14	06/2018 – 08/2018	"Servers MUST NOT send more than three **datagrams** including Initial and Handshake packets without receiving a packet from a verified source address."
Draft 15 – 32	10/2018 – 10/2020	"Servers MUST NOT send more than three times as many **bytes** as the number of bytes received prior to verifying the client's address."
Draft 33 – 34, RFC 9000	12/2020 – 01/2021, 05/2021	"[...] an endpoint MUST limit the amount of **data** it sends to the unvalidated address to three times the amount of data received from that address."

We find little discussion about the limit on the IETF mailing lists. In March 2018, 3600 ($= 3 \cdot 1200$) bytes have been discussed as "decently large" [282] to carry TLS certificates. A recent question on the exact motivation behind the 3× remains unanswered [237].

8.6.3 Influence of Top List Ranks

We verify whether our results depend on some type of popularity of the QUIC-based Web service using the Tranco list [122]. To this end, we split the Tranco list in groups of 100k (ranked) names and initiate QUIC and HTTPS handshakes for each name.

Figure 8.11 exhibits the relative amount of servers that are reachable via QUIC or only via HTTPS. On average, 21% of domains per rank group are reachable via QUIC. On top of this, $\approx$ 59% of additional names own a TLS certificate and are reachable over HTTPS. The popularity of a server has no influence on the popularity of QUIC deployment, as we observe a small standard deviation of $\sigma = 3$ across rank groups.

We also check whether the QUIC handshake classification is stable across ranks by mapping responses to a QUIC handshake type (amplification, multi-RTT *etc.*) and counting the relative number of servers per type. Figure 8.12 visualizes the results. Again, we find no significant differences across rank groups. The only exceptions are 1-RTT handshakes, which appear more popular among the 100k most popular QUIC servers (3.02% vs. ¡0.95%).

Both analysis indicate that our results are independent of the specific Tranco rank.

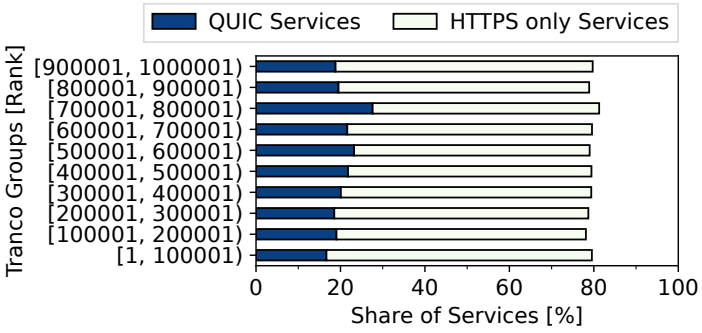


Figure 8.11: Service popularity across tranco rank groups. QUIC and HTTPS deployment rates are stable across rank groups.

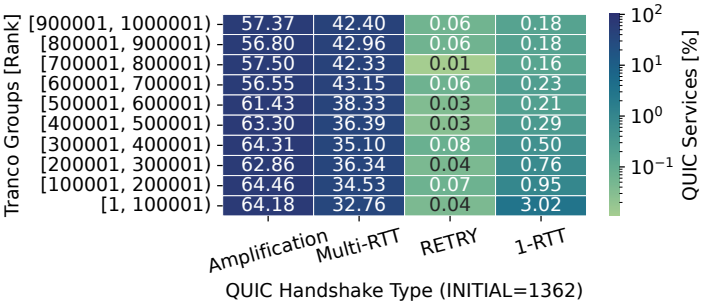


Figure 8.12: QUIC handshake classification per tranco rank group. Handshake types are mostly stable across rank groups.

8.6.4 Cruise-Liner TLS Certificates

Cruise-liner certificates [20] are certificates that are large in size due to many subject alternate names (SANs). We now check whether QUIC services are affected by cruise-liner certificates. To this end, we analyze all the leaf certificates received for all QUIC services. We inspect the total certificate size and the share of bytes required by all SANs. The results are visualized in Figure 8.13.

Overall, most SANs amount for less than 10% of bytes. Taking a closer look at the top 1% of certificates by SAN byte share, we find that they require at least 28.9% of bytes (horizontal threshold). Worryingly, 0.1% of certificates exhibit a high SAN byte share and exceed a common QUIC amplification limit (vertical threshold).

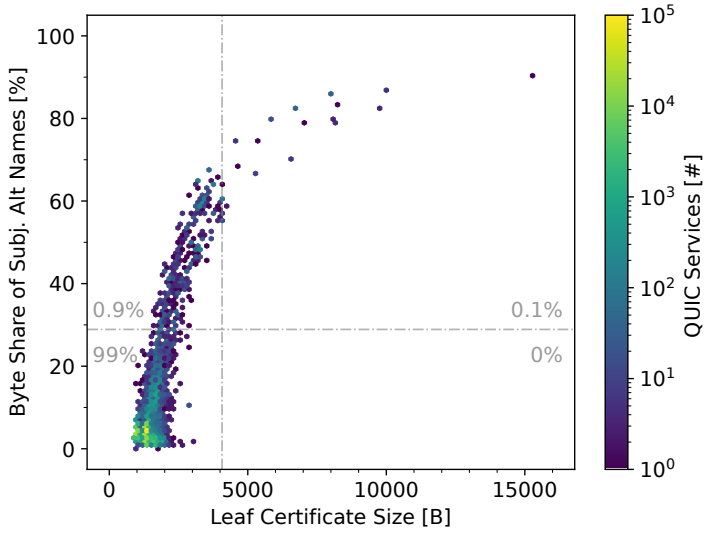


Figure 8.13: Relative size of subject alternative names (SANs). Cruise-liner certificates are rare for QUIC services.

8.7 Conclusion and Outlook

In this chapter, we measured and analyzed the QUIC handshake processes in the wild and found that the current Web certificate ecosystem challenges the QUIC design objective of a 1-RTT quick connection setup at low amplification potential. As a consequence, large portions of QUIC connection setups are either multi-RTT, do not comply to the amplification limit, or both. Future work shall closely monitor the evolution of the QUIC ecosystem and analyze the impact of measures to reduce certificate sizes effectively.

Responses from Hypergiants. We contacted Meta as well as Cloudflare. Details about our responsible disclosure policy are explained in Subsection 8.6.1.

8.8 Artifacts

All artifacts are available via the following public repository:

`https://github.com/ilabrg/artifacts-conext22-quic-tls`

This public repository provides up-to-date instructions for installing, configuring, and running our artifacts. We also archive the camera-ready version of our software on Zenodo:

`https://doi.org/10.5281/zenodo.7157904`

We present an overview of our toolchain and related data flow in Figure 8.14.

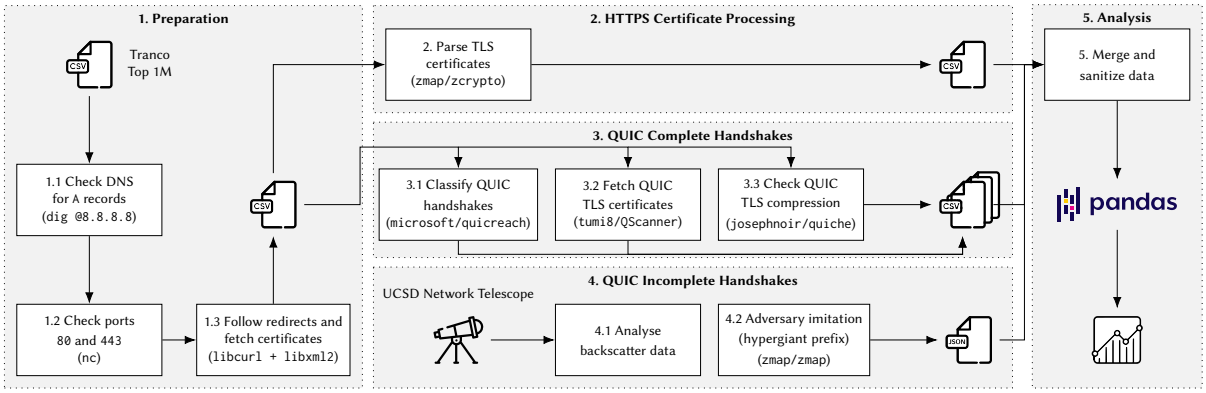


Figure 8.14: Overview of our HTTPS and QUIC analysis toolchain as well as related data flow.

Chapter 9

Conclusion and Outlook

In this thesis, we explored the complete DDoS landscape including attack coverage, mitigation, and prevention. We focused on methodologies that are directly applicable to both the Internet core and edge. This led to contributions and operational recommendations for commonly deployed and emerging measurement methods, mitigation solutions, and protocols, which ultimately improve today's Internet.

Chapter 2 investigated the completeness of honeypot observations. Honeypots are a prevalently used measurement tool to quantify reflective amplification attacks. Contrary to common belief, we found that honeypots detect only a fraction of baseline attacks, which we received from a DDoS mitigation provider. This highlights the importance of challenging and validating long-held beliefs in our research community. It also emphasizes the size and distributed nature of the Internet, which enables attackers to operate locally, *i.e.,* hiding in specific areas and misusing subsets of infrastructure for attacks. This impedes attack observations. To counteract this, we need a better model of the attacker behavior.

Chapter 3 explored reflective amplification attacks in the Internet Core. We introduced a passive DNS attack detection method for IXPs that works despite limitations such as packet sampling and truncation. Based on DNS fingerprinting, we identified a major attack entity not visible to honeypots. Surprisingly, we not did only detect prior unseen attacks but also observed a qualitative difference in attack properties, enabling a recommendation on DNSSEC key rollover practices. This illustrates the potential of correlating orthogonal measurement methods in the future, instead of simply scaling up current methods.

Chapter 4 analyzed remotely-triggered blackholing events at a large European IXP. This enabled us to assess the efficacy of this DDoS mitigation technique deployed in the Internet core. We found that on average blackholing only drops 50% of unwanted traffic because announcements with hyper-specific prefixes are not accepted by IXP members. Moreover, we found that DDoS attacks also target clients situated in ISP networks, *i.e.,* they are present now in the private sector. This emphasizes the importance of Internet measurements to validate current and emerging technologies, especially if they extend established protocols (RTBH overloads BGP semantics at IXPs) and require cooperation from multiple network participants. Even more if

such measurements refute upfront assumptions, *e.g.,* only servers with financial relevance are under attack.

Chapter 5 unveiled that attackers already utilize randomly-spoofed, state-building QUIC floods to attack CDN servers at the Internet edge. We showed this by reusing a method that was previously used to detect conceptually-similar TCP `SYN` floods, *i.e.,* Internet telescopes capturing backscatter traffic. We also performed synthetic tests of QUIC server implementations to confirm the efficacy of the built-in `RETRY` option. However, `RETRYs` are barely used in favor of fast connection setups. Our results indicate that although our community improved protocol design based on DDoS experience from the last decades, there is an inherent trade-off between performance and security that has to balance out legitimate and malicious scenarios.

Chapter 6 discovered that unprotected, operational traffic belonging to Industrial Control Systems (ICS) transits the Internet core. Such traffic is vulnerable to Man-In-The-Middle attacks, which include eavesdropping and traffic manipulation. Manipulations to ICS systems, *e.g.,* factories and power plants, must be prevented as they can lead to hazardous events. Our analyses remind the community that any system inter-connecting with the Internet, *i.e.,* is extended by IP, must be carefully examined with respect to its security features. But they also underpin that the Internet is part of critical infrastructure.

Chapter 7 measured open DNS components deployed in the Internet Edge. We found that the open DNS ecosystem now consists of 26% of transparent forwarders, which can be misused to launch reflective amplification attacks. Internet-wide scanning campaigns miss transparent forwarders due to common optimizations. However, by still being continuously deployed, transparent forwarders moved from being a rare artifact to a large part of the open DNS ecosystem. Our results suggest that longitudinal measurements are necessary to capture the complete evolution of *e.g.,* amplifiers. However, the measurement methodology should be regularly checked and improved to reflect current trends well.

Chapter 8 studied over 1M Web domains with 272k QUIC-enabled services and found that 35% of server Web certificates exceed the QUIC anti-amplification limit. Such large certificates directly slow down the QUIC connection setup as they lead to multi-RTT handshakes. In the case of server implementations which are not standard-compliant, larger certificates lead to higher amplification factors. We detected factors $\geq 30\times$ in IP spoofing scenarios for a major CDN. Our results draw attention to the fact that even though specific limits are given in standards, the real-world with its implementations can be different. This highlights the need for Internet measurements. But they also point out how current inefficiencies of a specific ecosystem can have dramatic cascading effects if combined with a new system.

DDoS Vantage Points and Methodologies in Future Work

Across the chapters of this thesis, we extensively used multiple DDoS vantage points. We observed and classified traffic patterns in the Internet core as well as the Internet edge, utilizing attack measurement methodologies incorporating honeypots and network telescopes. Despite unveiling surprisingly incomplete or local observations, which ultimately can lead to discrepancies across vantage points, we want to emphasize that the current methodologies are not *inherently* flawed.

The Internet is an, although incrementally, ever-changing system. Therefore, longitudinal measurement methodologies require reconsideration over time, including updated configuration and sanitization. In spite of all change, tools like telescopes will always pose the fundamental question of what can be observed using a completely passive methodology. Likewise, honeypots will keep challenging the idea of luring attackers to actively infer knowledge.

To illustrate that network telescopes and honeypots remain a relevant tool, we present two (passive and active) measurement opportunities for future research. First, observing the emerging HTTP3 ecosystem. Since HTTP3 relies on QUIC, telescopes enable non-intrusive measurements of the backend infrastructure, the attack landscape, and the deployment of various mitigation strategies using `RETRY`. Second, random-subdomain attacks via DNS over HTTPS. Since the payload of interest only becomes visible after a successful crypto handshake, interaction such as provided by honeypots is necessary.

Telescopes and honeypots, however, remain fundamentally challenged, in particular with the increased deployment of IPv6. In contrast to IPv4, IPv6 provides a significantly larger address space. Consequently, attackers are less likely to (*i*) randomly spoof the address that is part of a telescope and (*ii*) discover honeypots using Internet-wide scans. Correlating data from the Internet edge and core, as demonstrated in this thesis, could therefore potentially become even more important in the future.

A Comment on the Future of DDoS Attack Research

We have addressed future strategies to overcome and mitigate DDoS in each chapter separately. The overarching, harsh truth, however, is that DDoS attacks are here to stay, either because they remain economically feasible or they follow a completely different motivation. The underlying reasons may include technical limitations such as trade-offs, implementation errors and legacy issues, or non-monetary or non-political reasons such as bolstering reputation in the *scene* or just amusement, *for the lulz*. To cope with attacks we must continue to focus our efforts on three measurement pillars, a clear understanding of coverage, mitigation, and prevention.

It is the duty of Internet measurement research to precisely report in order to ultimately produce advice on how to improve the Internet, hand in hand with network operators. A continuous feedback loop between researchers and network operators helps that measurement methods are aligned with actual deployment scenarios and that results are made aware to those who may change deployment, fostering an ongoing process of improvement.

List of Figures

List of Tables

List of Acronyms

AS Autonomous System

ASN Autonomous System Number

BGP Border Gateway Protocol

CCC Cambridge Cybercrime Centre

CDF Cumulative Distribution Function

CDN Content Delivery Network

DNS Domain Name System

DDoS Distributed Denial-of-Service

DoS Denial-of-Service

HTTP(S) Hypertext Transfer Protocol (Secure)

ICMP Internet Control Message Protocol

ICS Industrial Control System

IETF Internet Engineering Task Force

IP Internet Protocol

ISP Internet Service Provider

IXP Internet Exchange Point

MAC Medium Access Control

QUIC Quick UDP Internet Connections

RBTH Remote Triggered Black Hole

RFC Request for Comments

RTT Round-Trip Time

SoK Systemization of Knowledge

TCP Transmission Control Protocol

TLS Transport Layer Security

UDP User Datagram Protocol

Bibliography

Peer-Reviewed References

[1] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A Multifaceted Approach to Understanding the Botnet Phenomenon," in *Proc. of ACM IMC*, Rio de Janeriro, Brazil: ACM, 2006, pp. 41–52. URL: https://doi.org/10.1145/1177080.1177086.

[2] Y. Afek, A. Bremler-Barr, and L. Shafir, "NXNSAttack: Recursive DNS Inefficiencies and Vulnerabilities," in *Proc. of USENIX Security*, Virtual Event: USENIX, 2020, pp. 631–648. URL: https://www.usenix.org/conference/usenixsecurity20/presentation/afek.

[3] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger, "Anatomy of a Large European IXP," in *Proc. of ACM SIGCOMM*, Helsinki, Finland: ACM, 2012, pp. 163–174. URL: https://doi.org/10.1145/2342356.2342393.

[4] M. Allman, "Putting DNS in Context," in *Proc. of ACM IMC*, Virtual Event: ACM, 2020, pp. 309–316. URL: https://doi.org/10.1145/3419394.3423659.

[5] M. Almeida, A. Finamore, D. Perino, N. Vallina-Rodriguez, and M. Varvello, "Dissecting DNS Stakeholders in Mobile Networks," in *Proc. of ACM CoNEXT*, Incheon, Republic of Korea: ACM, 2017, pp. 28–34. URL: https://doi.org/10.1145/3143361.3143375.

[6] M. Anagnostopoulos, G. Kambourakis, P. Kopanos, G. Louloudakis, and S. Gritzalis, "DNS Amplification Attack Revisited," *Elsevier Comput. Secur.*, vol. 39, no. B, pp. 475–485, 2013. URL: https://doi.org/10.1016/j.cose.2013.10.001.

[7] Anonymous, "Towards a Comprehensive Picture of the Great Firewall's DNS Censorship," in *Proc. of USENIX FOCI*, San Diego, USA: USENIX, 2014. URL: https://www.usenix.org/conference/foci14/workshop-program/presentation/anonymous.

[8] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai Botnet," in *Proc. of USENIX Security*, Vancouver, Canada: USENIX, 2017, pp. 1093–1110. URL: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis.

[9] M. Backes, T. Holz, C. Rossow, T. Rytilahti, M. Simeonovski, and B. Stock, "On the Feasibility of TTL-Based Filtering for DRDoS Mitigation," in *Proc. of Springer RAID*, Paris, France: Springer, 2016, pp. 303–322. URL: https://doi.org/10.1007/978-3-319-45719-2%5C_14.

[10] T. Bajtoš, P. Sokol, and T. Mézešová, "Multi-Stage Cyber-Attacks Detection in the Industrial Control Systems," *Springer Recent Devel. on Ind. Contr. Sys. Resil. and SSDC*, vol. 255, pp. 151–173, 2020. URL: https://doi.org/10.1007/978-3-030-31328-9_8.

[11] E. Balkanli and A. N. Zincir-Heywood, "On the Analysis of Backscatter Traffic," in *Proc. of IEEE LCN*, Edmonton, Canada: IEEE, 2014, pp. 671–678. URL: https://doi.org/10.1109/LCNW.2014.6927719.

[12] R. R. R. Barbosa, R. Sadre, and A. Pras, "A First Look into SCADA Network Traffic," in *Proc. of IEEE NOMS*, Maui, USA: IEEE, 2012, pp. 518–521. URL: https://doi.org/10.1109/NOMS.2012.6211945.

[13] R. R. R. Barbosa, R. Sadre, and A. Pras, "Difficulties in Modeling SCADA Traffic: A Comparative Analysis," in *Proc. of Springer PAM*, Vienna, Austria: Springer, 2012, pp. 126–135. URL: https://doi.org/10.1007/978-3-642-28537-0%5C_13.

[14] G. Bernieri, M. Conti, and F. Pascucci, "MimePot: a Model-based Honeypot for Industrial Control Networks," in *Proc. of IEEE SMC*, Bari, Italy: IEEE, 2019, pp. 433–438. URL: https://doi.org/10.1109/SMC.2019.8913891.

[15] R. Berthier and W. H. Sanders, "Specification-Based Intrusion Detection for Advanced Metering Infrastructures," in *Proc. of IEEE PRDC*, Pasadena, USA: IEEE, 2011, pp. 184–193. URL: https://doi.org/10.1109/PRDC.2011.30.

[16] E. W. Biersack, Q. Jacquemart, F. Fischer, J. Fuchs, O. Thonnard, G. Theodoridis, D. Tzovaras, and P. Vervier, "Visual Analytics for BGP Monitoring and Prefix Hijacking Identification," *IEEE Network*, vol. 26, no. 6, pp. 33–39, 2012. URL: https://doi.org/10.1109/MNET.2012.6375891.

[17] P. Biswal and O. Gnawali, "Does QUIC Make the Web Faster?" In *Proc. of IEEE GLOBECOM*, Washington, USA: IEEE, 2016. URL: https://doi.org/10.1109/GLOCOM.2016.7841749.

[18] N. Blenn, V. Ghiëtte, and C. Doerr, "Quantifying the Spectrum of Denial-of-Service Attacks through Internet Backscatter," in *Proc. of ACM ARES*, Reggio Calabria, Italy: ACM, 2017. URL: https://doi.org/10.1145/3098954.3098985.

[19] C. D. Murta and P. R. Torres Jr. and P. Mohapatra, "QRPp1-4: Characterizing Quality of Time and Topology in a Time Synchronization Network," in *Proc. of IEEE GLOBECOM*, San Francisco, USA: IEEE, 2006. URL: https://doi.org/10.1109/GLOCOM.2006.467.

[20] F. Cangialosi, T. Chung, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson, "Measurement and Analysis of Private Key Sharing in the HTTPS Ecosystem," in *Proc. of ACM SIGSAC CCS*, Vienna, Austria: ACM, 2016, pp. 628–640. URL: https://doi.org/10.1145/2976749.2978301.

[21] G. Carlucci, L. D. Cicco, and S. Mascolo, "HTTP over UDP: An Experimental Investigation of QUIC," in *Proc. of ACM SAC*, Salamanca, Spain: ACM, 2015, pp. 609–614. URL: https://doi.org/10.1145/2695664.2695706.

[22] M. Casado, T. Garfinkel, W. Cui, V. Paxson, and S. Savage, "Opportunistic Measurement: Extracting Insight from Spurious Traffic," in *Proc. of ACM HotNets*, College Park, Maryland: ACM, 2005. URL: https://conferences.sigcomm.org/hotnets/2005/papers/monitor_90.pdf.

[23] J. M. Ceron, C. Scholten, A. Pras, and J. Santanna, "MikroTik Devices Landscape, Realistic Honeypots, and Automated Attack Classification," in *Proc. of IEEE NOMS*, Virtual Event: IEEE, 2020. URL: https://doi.org/10.1109/NOMS47738.2020.9110336.

[24] N. Chatzis, G. Smaragdakis, J. Böttger, T. Krenc, and A. Feldmann, "On the Benefits of Using a Large IXP as an Internet Vantage Point," in *Proc. of ACM IMC*, Barcelona, Spain: ACM, 2013, pp. 333–346. URL: https://doi.org/10.1145/2504730.2504746.

[25] T. Chung, R. van Rijswijk-Deij, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson, "A Longitudinal, End-to-End View of the DNSSEC Ecosystem," in *Proc. of USENIX Security*, Vancouver, BC: USENIX, 2017, pp. 1307–1322. URL: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/chung.

[26] B. Collier, D. R. Thomas, R. Clayton, and A. Hutchings, "Booting the Booters: Evaluating the Effects of Police Interventions in the Market for Denial-of-Service Attacks," in *Proc. of ACM IMC*, Amsterdam, Netherlands: ACM, 2019, pp. 50–64. URL: https://doi.org/10.1145/3355369.3355592.

[27] Q. D. Coninck and O. Bonaventure, "Multipath QUIC: Design and Evaluation," in *Proc. of ACM CoNEXT*, Incheon, Republic of Korea: ACM, 2017, pp. 160–166. URL: https://doi.org/10.1145/3143361.3143370.

[28] S. Cook, B. Mathieu, P. Truong, and I. Hamchaoui, "QUIC: Better for What and for Whom?" In *Proc. of IEEE ICC*, Paris, France: IEEE, 2017. URL: https://doi.org/10.1109/ICC.2017.7997281.

[29] Y. Cui, T. Li, C. Liu, X. Wang, and M. Kühlewind, "Innovating Transport with QUIC: Design Approaches and Research Challenges," *IEEE Internet Computing*, vol. 21, no. 2, pp. 72–76, 2017. URL: https://doi.org/10.1109/MIC.2017.44.

[30] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir, "Taming the 800 Pound Gorilla: The Rise and Decline of NTP DDoS Attacks," in *Proc. of ACM IMC*, Vancouver, Canada: ACM, 2014, pp. 435–448. URL: https://doi.org/10.1145/2663716.2663717.

[31] A. Dainotti, A. King, K. C. Claffy, F. Papale, and A. Pescapè, "Analysis of a "/0" Stealth Scan from a Botnet," *IEEE/ACM Trans. on Netw.*, vol. 23, no. 2, pp. 341–354, 2015. URL: https://doi.org/10.1145/2398776.2398778.

[32] P. De Vaere, T. Bühler, M. Kühlewind, and B. Trammell, "Three Bits Suffice: Explicit Support for Passive Measurement of Internet Latency in QUIC and TCP," in *Proc. of ACM IMC*, Boston, USA: ACM, 2018, pp. 22–28. URL: https://dl.acm.org/citation.cfm?id=3278535.

[33] C. Dietzel, A. Feldmann, and T. King, "Blackholing at IXPs: On the Effectiveness of DDoS Mitigation in the Wild," in *Proc. of Springer PAM*, Heraklion, Greece: Springer, 2016, pp. 319–332. URL: https://doi.org/10.1007/978-3-319-30505-9%5C_24.

[34] C. Dietzel, M. Wichtlhuber, G. Smaragdakis, and A. Feldmann, "Stellar: Network Attack Mitigation Using Advanced Blackholing," in *Proc. of ACM CoNEXT*, Heraklion, Greece: ACM, 2018, pp. 152–164. URL: https://doi.org/10.1145/3281411.3281413.

[35] T. V. Doan, J. Fries, and V. Bajpai, "Evaluating Public DNS Services in the Wake of Increasing Centralization of DNS," in *Proc. of IFIP NETWORKING*, Virtual Event: IEEE, 2021. URL: https://doi.org/10.23919/IFIPNetworking52078.2021.9472831.

[36] N. Duffield, C. Lund, M. Thorup, and M. Thorup, "Estimating Flow Distributions from Sampled Flow Statistics," in *Proc. of ACM SIGCOMM*, Karlsruhe, Germany: ACM, 2003, pp. 325–336. URL: https://doi.org/10.1145/863955.863992.

[37] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A Search Engine Backed by Internet-Wide Scanning," in *Proc. of ACM SIGSAC CCS*, Denver, USA: ACM, 2015, pp. 542–553. URL: https://doi.org/10.1145/2810103.2813703.

[38] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast Internet-wide Scanning and Its Security Applications," in *Proc. of USENIX Security*, Washington, D.C.: USENIX, 2013, pp. 605–620. URL: https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric.

[39] M. Dusi, S. Napolitano, S. Niccolini, and S. Longo, "A Closer Look at Thin-Client Connections: Statistical Application Identification for QoE Detection," *IEEE Commun. Mag.*, vol. 50, no. 11, pp. 195–202, 2012. URL: https://doi.org/10.1109/MCOM.2012.6353701.

[40] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proc. of AAAI KDD*, Portland, USA: AAAI, 1996, pp. 226–231. URL: http://www.aaai.org/Library/KDD/1996/kdd96-037.php.

[41] C. Fachkha, E. Bou-Harb, and M. Debbabi, "Fingerprinting Internet DNS Amplification DDoS Activities," in *Proc. of IEEE NTMS*, Dubai, United Arab Emirates: IEEE, 2014. URL: https://doi.org/10.1109/NTMS.2014.6814019.

[42] C. Fachkha, E. Bou-Harb, and M. Debbabi, "Inferring Distributed Reflection Denial of Service Attacks from Darknet," *Elsevier Comp. Commun.*, vol. 62, pp. 59–71, 2015. URL: https://doi.org/10.1016/j.comcom.2015.01.016.

[43] S. L. Feibish, Y. Afek, A. Bremler-Barr, E. Cohen, and M. Shagam, "Mitigating DNS Random Subdomain DDoS Attacks by Distinct Heavy Hitters Sketches," in *Proc. of ACM/IEEE HotWeb*, San Jose, USA: ACM, 2017. URL: https://doi.org/10.1145/3132465.3132474.

[44] P. Ferretti, M. Pogliani, and S. Zanero, "Characterizing Background Noise in ICS Traffic Through a Set of Low Interaction Honeypots," in *Proc. of ACM CPS-SPC*, London, United Kingdom: ACM, 2019, pp. 51–61. URL: https://doi.org/10.1145/3338499.3357361.

[45] O. Fonseca, Í. Cunha, E. Fazzion, W. Meira, B. Junior, R. A. Ferreira, and E. Katz-Bassett, "Tracking Down Sources of Spoofed IP Packets," in *Proc. of IFIP NETWORKING*, Catania, Italy: IEEE, 2020, pp. 208–216. URL: https://doi.org/10.1145/3360468.3368175.

[46] E. Gagliardi and O. Levillain, "Analysis of QUIC Session Establishment and Its Implementations," in *IFIP WISTP and LNSC volume 12024*, Cham, Switzerland: Springer, 2019, pp. 169–184. URL: https://doi.org/10.1007/978-3-030-41702-4_11.

[47] O. Gasser, Q. Scheitle, B. Rudolph, C. Denis, N. Schricke, and G. Carle, "The Amplification Threat Posed by Publicly Reachable BACnet Devices," *River Publ. J. of Cyber Sec. and Mob.*, vol. 6, no. 1, pp. 77–104, 2017. URL: https://doi.org/10.13052/jcsm2245-1439.614.

[48] V. Ghiëtte, N. Blenn, and C. Doerr, "Remote Identification of Port Scan Toolchains," in *Proc. of IEEE NTMS*, Larnaca, Cyprus: IEEE, 2016. URL: https://doi.org/10.1109/NTMS.2016.7792471.

[49] V. Giotsas, P. Richter, G. Smaragdakis, A. Feldmann, C. Dietzel, and A. W. Berger, "Inferring BGP Blackholing Activity in the Internet," in *Proc. of ACM IMC*, London, United Kingdom: ACM, 2017, pp. 1–14. URL: https://doi.org/10.1145/3131365.3131379.

[50] H. Griffioen, K. Oosthoek, P. van der Knaap, and C. Doerr, "Scan, Test, Execute: Adversarial Tactics in Amplification DDoS Attacks," in *Proc. of ACM SIGSAC CCS*, Virtual Event: ACM, 2021, pp. 940–954. URL: https://doi.org/10.1145/3460120.3484747.

[51] Y. He, M. Faloutsos, S. Krishnamurthy, and B. Huffaker, "On Routing Asymmetry in the Internet," in *Proc. of IEEE GLOBECOM*, St. Louis, USA: IEEE, 2005, pp. 904–909. URL: https://doi.org/10.1109/GLOCOM.2005.1577769.

[52] T. Heinrich, R. R. Obelheiro, and C. A. Maziero, "New Kids on the DRDoS Block: Characterizing Multiprotocol and Carpet Bombing Attacks," in *Proc. of Springer PAM*, Virtual Event: Springer, 2021, pp. 427–443. URL: https://doi.org/10.1007/978-3-030-72582-2%5C_16.

[53] R. Hiesgen, M. Nawrocki, A. King, A. Dainotti, T. C. Schmidt, and M. Wählisch, "Spoki: Unveiling a New Wave of Scanners through a Reactive Network Telescope," in *Proc. of USENIX Security*, Boston, USA: USENIX, 2022, pp. 431–448. URL: https://www.usenix.org/conference/usenixsecurity22/presentation/hiesgen.

[54] N. Hinze, M. Nawrocki, M. Jonker, A. Dainotti, T. C. Schmidt, and M. Wählisch, "On the Potential of BGP Flowspec for DDoS Mitigation at Two Sources: ISP and IXP," in *Proc. of ACM SIGCOMM Posters and Demos*, Budapest, Hungary: ACM, 2018, pp. 57–59. URL: https://doi.org/10.1145/3234200.3234209.

[55] P. Hoffman, G. Grinstein, and D. Pinkney, "Dimensional Anchors: A Graphic Primitive for Multidimensional Multivariate Information Visualizations," in *Proc. of ACM NPIVM*, Kansas City, USA: ACM, 1999, pp. 9–16. URL: https://doi.org/10.1145/331770.331775.

[56] R. Holz, J. Hiller, J. Amann, A. Razaghpanah, T. Jost, N. Vallina-Rodriguez, and O. Hohlfeld, "Tracking the Deployment of TLS 1.3 on the Web: A Story of Experimentation and Centralization," *ACM Sigcomm Comput. Commun. Rev.*, vol. 50, no. 3, pp. 3–15, 2020. URL: https://doi.org/10.1145/3411740.3411742.

[57] T. Holz and F. Raynal, "Detecting Honeypots and other Suspicious Environments," in *Proc. of IEEE IAW*, West Point, USA: IEEE, 2005, pp. 29–36. URL: https://doi.org/10.1109/IAW.2005.1495930.

[58] M. Honda, Y. Nishida, C. Raiciu, A. Greenhalgh, M. Handley, and H. Tokuda, "Is it still possible to extend TCP?" In *Proc. of ACM IMC*, Berlin, Germany: ACM, 2011, pp. 181–194. URL: https://doi.org/10.1145/2068816.2068834.

[59] Y. Hou, J. Such, and A. Rashid, "Understanding Security Requirements for Industrial Control System Supply Chains," in *Proc. of ACM SEsCPS*, Montreal, Canada: IEEE, 2019, pp. 50–53. URL: https://doi.org/10.1109/SEsCPS.2019.00016.

[60] V. M. Igure, S. A. Laughter, and R. D. Williams, "Security Issues in SCADA Networks," *Elsevier Comput. Secur.*, vol. 25, no. 7, pp. 498–506, 2006. URL: https://doi.org/10.1016/j.cose.2006.03.001.

[61] N. Ishikura, D. Kondo, V. Vassiliades, I. Iordanov, and H. Tode, "DNS Tunneling Detection by Cache-Property-Aware Features," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 2, pp. 1203–1217, 2021. URL: https://doi.org/10.1109/TNSM.2021.3078428.

[62] L. Izhikevich, R. Teixeira, and Z. Durumeric, "LZR: Identifying Unexpected Internet Services," in *Proc. of USENIX Security*, Virtual Event: USENIX, 2021, pp. 3111–3128. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/izhikevich.

[63] B. Jones, N. Feamster, V. Paxson, N. Weaver, and M. Allman, "Detecting DNS Root Manipulation," in *Proc. of Springer PAM*, Heraklion, Greece: Springer, 2016, pp. 276–288. URL: https://doi.org/10.1007/978-3-319-30505-9%5C_21.

[64] M. Jonker, A. King, J. Krupp, C. Rossow, A. Sperotto, and A. Dainotti, "Millions of Targets Under Attack: A Macroscopic Characterization of the DoS Ecosystem," in *Proc. of ACM IMC*, London, United Kingdom: ACM, 2017, pp. 100–113. URL: https://doi.org/10.1145/3131365.3131383.

[65] M. Jonker, A. Pras, A. Dainotti, and A. Sperotto, "A First Joint Look at DoS Attacks and BGP Blackholing in the Wild," in *Proc. of ACM IMC*, Boston, USA: ACM, 2018, pp. 457–463. URL: https://dl.acm.org/citation.cfm?id=3278571.

[66] M. Jonker, A. Sperotto, R. van Rijswijk-Deij, R. Sadre, and A. Pras, "Measuring the Adoption of DDoS Protection Services," in *Proc. of ACM IMC*, Santa Monica, USA: ACM, 2016, pp. 279–285. URL: https://doi.org/10.1145/2987443.2987487.

[67] A. M. Kakhki, S. Jero, D. Choffnes, C. Nita-Rotaru, and A. Mislove, "Taking a Long Look at QUIC: An Approach for Rigorous Evaluation of Rapidly Evolving Transport Protocols," in *Proc. of ACM IMC*, London, United Kingdom: ACM, 2017, pp. 290–303. URL: https://doi.org/10.1145/3131365.3131368.

[68] M. Karami, Y. Park, and D. McCoy, "Stress Testing the Booters: Understanding and Undermining the Business of DDoS Services," in *Proc. of ACM WWW*, Montreal, Canada: ACM, 2016, pp. 1033–1043. URL: https://doi.org/10.1145/2872427.2883004.

[69] J. Klick, S. Lau, M. Wählisch, and V. Roth, "Towards Better Internet Citizenship: Reducing the Footprint of Internet-wide Scans by Topology Aware Prefix Selection," in *Proc. of ACM IMC*, Santa Monica, USA: ACM, 2016, pp. 421–427. URL: https://doi.org/10.1145/2987443.2987457.

[70] D. Kopp, C. Dietzel, and O. Hohlfeld, "DDoS Never Dies? An IXP Perspective on DDoS Amplification Attacks," in *Proc. of Springer PAM*, Virtual Event: Springer, 2021, pp. 284–301. URL: https://doi.org/10.1007/978-3-030-72582-2_17.

[71] D. Kopp, M. Wichtlhuber, I. Poese, J. Santanna, O. Hohlfeld, and C. Dietzel, "DDoS Hide & Seek: On the Effectiveness of a Booter Services Takedown," in *Proc. of ACM IMC*, Amsterdam, Netherlands: ACM, 2019, pp. 65–72. URL: https://doi.org/10.1145/3355369.3355590.

[72] M. Korczyński, Y. Nosyk, Q. Lone, M. Skwarek, B. Jonglez, and A. Duda, "Don't Forget to Lock the Front Door! Inferring the Deployment of Source Address Validation of Inbound Traffic," in *Proc. of Springer PAM*, Eugene, USA: Springer, 2020, pp. 107–121. URL: https://doi.org/10.1007/978-3-030-44081-7%5C_7.

[73] M. Korczyński, Y. Nosyk, Q. Lone, M. Skwarek, B. Jonglez, and A. Duda, "Inferring the Deployment of Inbound Source Address Validation Using DNS Resolvers," in *Proc. of ACM ANRW*, Virtual Event: ACM, 2020, pp. 9–11. URL: https://doi.org/10.1145/3404868.3406668.

[74] M. Kosek, T. V. Doan, M. Granderath, and V. Bajpai, "One to Rule Them All? A First Look at DNS over QUIC," in *Proc. of Springer PAM*, Virtual Event: Springer, 2022, pp. 537–551. URL: https://doi.org/10.1007/978-3-030-98785-5%5C_24.

[75] M. Kosek, L. Schumann, R. Marx, T. V. Doan, and V. Bajpai, "DNS Privacy with Speed? Evaluating DNS over QUIC and its Impact on Web Performance," in *Proc. of ACM IMC*, Nice, France: ACM, 2022, pp. 44–50. URL: https://doi.org/10.1145/3517745.3561445.

[76] L. Krämer, J. Krupp, D. Makita, T. Nishizoe, T. Koide, K. Yoshioka, and C. Rossow, "AmpPot: Monitoring and Defending Amplification DDoS Attacks," in *Proc. of Springer RAID*, Kyoto, Japan: Springer, 2015, pp. 615–636. URL: https://doi.org/10.1007/978-3-319-26362-5%5C_28.

[77] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson, "Netalyzr: Illuminating The Edge Network," in *Proc. of ACM IMC*, Melbourne, Australia: ACM, 2010, pp. 246–259. URL: https://doi.org/10.1145/1879141.1879173.

[78] J. Krupp, M. Backes, and C. Rossow, "Identifying the Scan and Attack Infrastructures Behind Amplification DDoS Attacks," in *Proc. of ACM SIGSAC CCS*, Vienna, Austria: ACM, 2016, pp. 1426–1437. URL: https://doi.org/10.1145/2976749.2978293.

[79] J. Krupp, M. Karami, C. Rossow, D. McCoy, and M. Backes, "Linking Amplification DDoS Attacks to Booter Services," in *Proc. of Springer RAID*, Atlanta, USA: Springer, 2017, pp. 427–449. URL: https://doi.org/10.1007/978-3-319-66332-6%5C_19.

[80] M. Kührer, T. Hupperich, J. Bushart, C. Rossow, and T. Holz, "Going Wild: Large-Scale Classification of Open DNS Resolvers," in *Proc. of ACM IMC*, Tokyo, Japan: ACM, 2015, pp. 355–368. URL: https://doi.org/10.1145/2815675.2815683.

[81] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, "Exit from Hell? Reducing the Impact of Amplification DDoS Attacks," in *Proc. of USENIX Security*, San Diego, USA: USENIX, 2014, pp. 111–125. URL: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/kuhrer.

[82] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, "Hell of a Handshake: Abusing TCP for Reflective Amplification DDoS Attacks," in *Proc. of USENIX WOOT*, San Diego, USA: USENIX, 2014. URL: https://www.usenix.org/conference/woot14/workshop-program/presentation/kuhrer.

[83] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, J. Bailey, J. Dorfman, J. Roskind, J. Kulik, P. Westin, R. Tenneti, R. Shade, R. Hamilton, V. Vasiliev, W.-T. Chang, and Z. Shi, "The QUIC Transport Protocol: Design and Internet-Scale Deployment," in *Proc. of ACM SIGCOMM*, Los Angeles, USA: ACM, 2017, pp. 183–196. URL: https://doi.org/10.1145/3098822.3098842.

[84] K. Lee, C. Huveneers, O. Gimenez, V. Peddemors, and R. Harcourt, "To Catch or to Sight? A Comparison of Demographic Parameter Estimates Obtained from Mark-Recapture and Mark-Resight Models," *Springer Biodiv. and Conserv.*, vol. 23, no. 11, pp. 2781–2800, 2014. URL: https://doi.org/10.1007/s10531-014-0748-9.

[85] F. Lichtblau, F. Streibelt, T. Krüger, P. Richter, and A. Feldmann, "Detection, Classification, and Analysis of Inter-Domain Traffic with Spoofed Source IP Addresses," in *Proc. of ACM IMC*, London, United Kingdom: ACM, 2017, pp. 86–99. URL: https://doi.org/10.1145/3131365.3131367.

[86] H. Lin, A. Slagell, C. Di Martino, Z. Kalbarczyk, and R. K. Iyer, "Adapting Bro into SCADA: Building a Specification-Based Intrusion Detection System for the DNP3 Protocol," in *Proc. of ACM CSIIRW*, Oak Ridge, USA: ACM, 2013. URL: https://doi.org/10.1145/2459976.2459982.

[87] B. Liu, C. Lu, H. Duan, Y. Liu, Z. Li, S. Hao, and M. Yang, "Who Is Answering My Queries: Understanding and Characterizing Interception of the DNS Resolution Path," in *Proc. of USENIX Security*, Baltimore, USA: USENIX, 2018, pp. 1113–1128. URL: https://www.usenix.org/conference/usenixsecurity18/presentation/liu-baojun.

[88] J. Liu, Y. Xiao, S. Li, W. Liang, and C. P. Chen, "Cyber Security and Privacy Issues in Smart Grids," *IEEE Commun. Surv. Tutorials*, vol. 14, no. 4, pp. 981–997, 2012. URL: https://doi.org/10.1109/SURV.2011.122111.00145.

[89] K. Lu, T. Chai, H. Xu, S. Prasad, J. Yan, and Z. Zhang, "Research on Unexpected DNS Response from Open DNS Resolvers," *Oxford The Comput. J.*, vol. 65, no. 9, pp. 2276–2298, 2021. URL: https://doi.org/10.1093/comjnl/bxab063.

[90] X. Luo, L. Wang, Z. Xu, K. Chen, J. Yang, and T. Tian, "A Large Scale Analysis of DNS Water Torture Attack," in *Proc. of ACM CSAI*, Shenzhen, China: ACM, 2018, pp. 168–173. URL: https://doi.org/10.1145/3297156.3297272.

[91] R. Lychev, S. Jero, A. Boldyreva, and C. Nita-Rotaru, "How Secure and Quick is QUIC? Provable Security and Performance Analyses," in *Proc. of IEEE S&P*, San Jose, USA: IEEE, 2015, pp. 214–231. URL: https://doi.org/10.1109/SP.2015.21.

[92] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Microtome J. of Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

[93] D. C. MacFarland, C. A. Shue, and A. J. Kalafut, "Characterizing Optimal DNS Amplification Attacks and Effective Mitigation," in *Proc. of Springer PAM*, New York, USA: Springer, 2015, pp. 15–27. URL: https://doi.org/10.1007/978-3-319-15509-8%5C_2.

[94] D. Madariaga, L. Torrealba, J. Madariaga, J. Bermúdez, and J. Bustos-Jiménez, "Analyzing the Adoption of QUIC From a Mobile Development Perspective," in *Proc. of ACM SIGCOMM EPIQ*, Virtual Event: ACM, 2020, pp. 35–41. URL: https://doi.org/10.1145/3405796.3405830.

[95] K. Man, Z. Qian, Z. Wang, X. Zheng, Y. Huang, and H. Duan, "DNS Cache Poisoning Attack Reloaded: Revolutions with Side Channels," in *Proc. of ACM SIGSAC CCS*, Virtual Event: ACM, 2020, pp. 1337–1350. URL: https://doi.org/10.1145/3372297.3417280.

[96] I. L. Meitei, K. J. Singh, and T. De, "Detection of DDoS DNS Amplification Attack Using Classification Algorithm," in *Proc. of ACM ICIA*, Pondicherry, India: ACM, 2016. URL: https://doi.org/10.1145/2980258.2980431.

[97] B. Miller and D. Rowe, "A Survey of SCADA and Critical Infrastructure Incidents," in *Proc. of ACM RIIT*, Calgary, Canada: ACM, 2012, pp. 51–56. URL: https://doi.org/10.1145/2380790.2380805.

[98] A. Mirian, Z. Ma, D. Adrian, M. Tischer, T. Chuenchujit, T. Yardley, R. Berthier, J. Mason, Z. Durumeric, J. Halderman, and M. Bailey, "An Internet-Wide View of ICS Devices," in *Proc. of IEEE PST*, Auckland, New Zealand: IEEE, 2016, pp. 96–103. URL: https://doi.org/10.1109/PST.2016.7906943.

[99] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," *ACM Sigcomm Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, 2004. URL: http://doi.acm.org/10.1145/997150.997156.

[100] A. Mitseva, A. Panchenko, and T. Engel, "The State of Affairs in BGP Security: A Survey of Attacks and Defenses," *Elsevier Comp. Commun.*, vol. 124, pp. 45–60, 2018. URL: https://doi.org/10.1016/j.comcom.2018.04.013.

[101] S.-J. Moon, Y. Yin, R. A. Sharma, Y. Yuan, J. M. Spring, and V. Sekar, "Accurately Measuring Global Risk of Amplification Attacks using AmpMap," in *Proc. of USENIX Security*, Virtual Event: USENIX, 2021, pp. 3881–3898. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/moon.

[102] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," *ACM Trans. Comput. Syst.*, vol. 24, no. 2, pp. 115–139, 2006. URL: https://doi.org/10.1145/1132026.1132027.

[103] D. Moore, C. Shannon, and K. Claffy, "Code-Red: A Case Study on the Spread and Victims of an Internet Worm," in *Proc. of ACM SIGCOMM IMW*, Marseille, France: ACM, 2002, pp. 273–284. URL: https://doi.org/10.1145/637201.637244.

[104] T. Morris, R. Vaughn, and Y. Dandass, "A Retrofit network Intrusion Detection System for MODBUS RTU and ASCII Industrial Control Systems," in *Proc. of IEEE HICSS*, Grand Wailea, USA: IEEE, 2012, pp. 2338–2345. URL: https://doi.org/10.1109/HICSS.2012.78.

[105] G. C. M. Moura, J. Heidemann, M. Müller, R. de O. Schmidt, and M. Davids, "When the Dike Breaks: Dissecting DNS Defenses During DDoS," in *Proc. of ACM IMC*, Boston, USA: ACM, 2018, pp. 8–21. URL: https://doi.org/10.1145/3278532.3278534.

[106] G. C. M. Moura, J. Heidemann, R. d. O. Schmidt, and W. Hardaker, "Cache Me If You Can: Effects of DNS Time-to-Live," in *Proc. of ACM IMC*, Amsterdam, Netherlands: ACM, 2019, pp. 101–115. URL: https://doi.org/10.1145/3355369.3355568.

[107] S. Mukkamala, K. Yendrapalli, R. Basnet, M. K. Shankarapani, and A. H. Sung, "Detection of Virtual Environments and Low Interaction Honeypots," in *Proc. of IEEE IAW*, West Point, USA: IEEE, 2007, pp. 92–98. URL: https://doi.org/10.1109/IAW.2007.381919.

[108] L. Müller, M. Luckie, B. Huffaker, K. Claffy, and M. Barcellos, "Challenges in Inferring Spoofed Traffic at IXPs," in *Proc. of ACM CoNEXT*, Orlando, USA: ACM, 2019, pp. 96–109. URL: https://doi.org/10.1145/3359989.3365422.

[109] M. Nawrocki, J. Blendin, C. Dietzel, T. C. Schmidt, and M. Wählisch, "Down the Black Hole: Dismantling Operational Practices of BGP Blackholing at IXPs," in *Proc. of ACM IMC*, Amsterdam, Netherlands: ACM, 2019, pp. 435–448. URL: https://doi.org/10.1145/3355369.3355593.

[110] M. Nawrocki, R. Hiesgen, T. C. Schmidt, and M. Wählisch, "QUICsand: Quantifying QUIC Reconnaissance Scans and DoS Flooding Events," in *Proc. of ACM IMC*, Virtual Event: ACM, 2021, pp. 283–291. URL: https://doi.org/10.1145/3487552.3487840.

[111] M. Nawrocki, M. Jonker, T. C. Schmidt, and M. Wählisch, "The Far Side of DNS Amplification: Tracing the DDoS Attack Ecosystem from the Internet Core," in *Proc. of ACM IMC*, Virtual Event: ACM, 2021, pp. 419–434. URL: https://doi.org/10.1145/3487552.3487835.

[112] M. Nawrocki, M. Koch, T. C. Schmidt, and M. Wählisch, "Transparent Forwarders: An Unnoticed Component of the Open DNS Infrastructure," in *Proc. of ACM CoNEXT*, Virtual Event: ACM, 2021, pp. 454–462. URL: https://doi.org/10.1145/3485983.3494872.

[113] M. Nawrocki, J. Kristoff, C. Kanich, R. Hiesgen, T. C. Schmidt, and M. Wählisch, "SoK: A Data-driven View on Methods to Detect Reflective Amplification DDoS Attacks Using Honeypots," in *Proc. of IEEE Euro S&P*, Delft, Netherlands: IEEE, 2023, pp. 576–591. URL: https://doi.org/10.1109/EuroSP57164.2023.00041.

[114] M. Nawrocki, T. C. Schmidt, and M. Wählisch, "Uncovering Vulnerable Industrial Control Systems from the Internet Core," in *Proc. of IEEE NOMS*, Virtual Event: IEEE, 2020. URL: https://doi.org/10.1109/NOMS47738.2020.9110256.

[115] M. Nawrocki, T. C. Schmidt, and M. Wählisch, "Industrial Control Protocols in the Internet Core: Dismantling Operational Practices," *Wiley Int. J. Netw. Manag.*, vol. 32, no. 1, 2022. URL: https://doi.org/10.1002/nem.2158.

[116] M. Nawrocki, P. F. Tehrani, R. Hiesgen, J. Mücke, T. C. Schmidt, and M. Wählisch, "On the Interplay between TLS Certificates and QUIC Performance," in *Proc. of ACM CoNEXT*, Rome, Italy: ACM, 2022, pp. 204–213. URL: https://doi.org/10.1145/3555050.3569123.

[117] A. A. Niaki, W. R. Marczak, S. Farhoodi, A. McGregor, P. Gill, and N. Weaver, "Cache Me Outside: A New Look at DNS Cache Probing," in *Proc. of Springer PAM*, Cham, Switzerland: Springer, 2021, pp. 427–443. URL: https://doi.org/10.1007/978-3-030-72582-2%5C_25.

[118] A. Noroozian, M. Korczyński, C. H. Gañan, D. Makita, K. Yoshioka, and M. Van Eeten, "Who Gets the Boot? Analyzing Victimization by DDoS-as-a-Service," in *Proc. of Springer RAID*, Paris, France: Springer, 2016, pp. 368–389. URL: https://doi.org/10.1007/978-3-319-45719-2%5C_17.

[119] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson, "Characteristics of Internet Background Radiation," in *Proc. of ACM IMC*, Taormina, Italy: ACM, 2004, pp. 27–40. URL: https://doi.org/10.1145/1028788.1028794.

[120] J. Park, A. Khormali, M. Mohaisen, and A. Mohaisen, "Where Are You Taking Me? Behavioral Analysis of Open DNS Resolvers," in *Proc. of IEEE/IFIP DSN*, Portland, USA: IEEE, 2019, pp. 493–504. URL: https://doi.org/10.1109/DSN.2019.00057.

[121] M. Piraux, Q. D. Coninck, and O. Bonaventure, "Observing the Evolution of QUIC Implementations," in *Proc. of ACM SIGCOMM EPIQ*, Heraklion, Greece: ACM, 2018, pp. 8–14. URL: https://doi.org/10.1145/3284850.3284852.

[122] V. L. Pochat, T. van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, "Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation," in *Proc. of ISOC NDSS*, San Diego, USA: Internet Society, 2019. URL: https://www.ndss-symposium.org/ndss-paper/tranco-a-research-oriented-top-sites-ranking-hardened-against-manipulation/.

[123] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye, "IP Geolocation Databases: Unreliable?" *ACM Sigcomm Comput. Commun. Rev.*, vol. 41, no. 2, pp. 53–56, 2011. URL: https://doi.org/10.1145/1971162.1971171.

[124] P. A. Ralston, J. H. Graham, and J. L. Hieb, "Cyber Security Risk Assessment for SCADA and DCS Networks," *Elsevier ISA Transactions*, vol. 46, no. 4, pp. 583–594, 2007. URL: https://doi.org/10.1016/j.isatra.2007.04.003.

[125] A. Randall, E. Liu, G. Akiwate, R. Padmanabhan, G. M. Voelker, S. Savage, and A. Schulman, "Trufflehunter: Cache Snooping Rare Domains at Large Public DNS Resolvers," in *Proc. of ACM IMC*, Virtual Event: ACM, 2020, pp. 50–64. URL: https://doi.org/10.1145/3419394.3423640.

[126] A. Randall, E. Liu, R. Padmanabhan, G. Akiwate, G. M. Voelker, S. Savage, and A. Schulman, "Home is Where the Hijacking is: Understanding DNS Interception by Residential Routers," in *Proc. of ACM IMC*, Virtual Event: ACM, 2021, pp. 390–397. URL: https://doi.org/10.1145/3487552.3487817.

[127] A. Reuter, R. Bush, I. Cunha, E. Katz-Bassett, T. C. Schmidt, and M. Wählisch, "Towards a Rigorous Methodology for Measuring Adoption of RPKI Route Validation and Filtering," *ACM Sigcomm Comput. Commun. Rev.*, vol. 48, no. 1, pp. 19–27, 2018. URL: https://doi.org/10.1145/3211852.3211856.

[128] P. Richter and A. Berger, "Scanning the Scanners: Sensing the Internet from a Massively Distributed Network Telescope," in *Proc. of ACM IMC*, Amsterdam, Netherlands: ACM, 2019, pp. 144–157. URL: https://doi.org/10.1145/3355369.3355595.

[129] P. Richter, G. Smaragdakis, A. Feldmann, N. Chatzis, J. Boettger, and W. Willinger, "Peering at Peerings: On the Role of IXP Route Servers," in *Proc. of ACM IMC*, Vancouver, Canada: ACM, 2014, pp. 31–44. URL: https://doi.org/10.1145/2663716.2663757.

[130] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and A. Pras, "A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 6, pp. 1877–1888, 2016. URL: https://doi.org/10.1109/JSAC.2016.2558918.

[131] R. van Rijswijk-Deij, A. Sperotto, and A. Pras, "DNSSEC and Its Potential for DDoS Attacks: A Comprehensive Measurement Study," in *Proc. of ACM IMC*, Vancouver, Canada: ACM, 2014, pp. 449–460. URL: https://doi.org/10.1145/2663716.2663731.

[132] C. Rosborough, C. Gordon, and B. Waldron, "All About Eve: Comparing DNP3 Secure Authentication With Standard Security Technologies for SCADA Communications," in *Proc. of MIPSYCON*, Minnesota, USA: University of Minnesota, 2019. URL: https://assets.ccaps.umn.edu/documents/CPE-Conferences/MIPSYCON-Papers/2019/AllAboutEve.pdf.

[133] C. Rossow, "Amplification Hell: Revisiting Network Protocols for DDoS Abuse," in *Proc. of ISOC NDSS*, San Diego, USA: Internet Society, 2014. URL: https://www.ndss-symposium.org/ndss2014/amplification-hell-revisiting-network-protocols-ddos-abuse.

[134] J. E. Rubio, C. Alcaraz, R. Roman, and J. Lopez, "Current Cyber-Defense Trends in Industrial Control Systems," *Elsevier Comput. Secur.*, vol. 87, 2019. URL: https://doi.org/10.1016/j.cose.2019.06.015%22.

[135] J. Rüth, I. Poese, C. Dietzel, and O. Hohlfeld, "A First Look at QUIC in the Wild," in *Proc. of Springer PAM*, Berlin, Germany: Springer, 2018, pp. 255–268. URL: https://doi.org/10.1007/978-3-319-76481-8_19.

[136] J. J. Santanna, R. O. De Schmidt, D. Tuncer, J. De Vries, L. Z. Granville, and A. Pras, "Booter Blacklist: Unveiling DDoS-for-Hire Websites," in *Proc. of IEEE CNSM*, Montreal, Canada: IEEE, 2016, pp. 144–152. URL: https://doi.org/10.1109/CNSM.2016.7818410.

[137] J. J. Santanna, R. van Rijswijk-Deij, R. Hofstede, A. Sperotto, M. Wierbosch, L. Z. Granville, and A. Pras, "Booters - An Analysis of DDoS-as-a-Service Attacks," in *Proc. IFIP/IEEE IM*, Ottawa, Canada: IEEE, 2015, pp. 243–251. URL: https://doi.org/10.1109/INM.2015.7140298.

[138] Q. Scheitle, O. Hohlfeld, J. Gamba, J. Jelten, T. Zimmermann, S. D. Strowes, and N. Vallina-Rodriguez, "A Long Way to the Top: Significance, Structure, and Stability of Internet Top Lists," in *Proc. of ACM IMC*, Boston, USA: ACM, 2018, pp. 478–493. URL: https://doi.org/10.1145/3278532.3278574.

[139] K. Schomp, T. Callahan, M. Rabinovich, and M. Allman, "On Measuring the Client-Side DNS Infrastructure," in *Proc. of ACM IMC*, Barcelona, Spain: ACM, 2013, pp. 77–90. URL: https://doi.org/10.1145/2504730.2504734.

[140] A. V. Serbanescu, S. Obermeier, and D.-Y. Yu, "ICS Threat Analysis Using a Large-Scale Honeynet," in *Proc. of BCS ICS-CSR*, Ingolstadt, Germany: BCS, 2015, pp. 20–30. URL: https://doi.org/10.14236/ewic/ICS2015.3.

[141] P. Sermpezis, V. Kotronis, P. Gigis, X. Dimitropoulos, D. Cicalese, A. King, and A. Dainotti, "ARTEMIS: Neutralizing BGP Hijacking Within a Minute," *IEEE/ACM Trans. on Netw.*, vol. 26, no. 6, pp. 2471–2486, 2018. URL: https://doi.org/10.1109/TNET.2018.2869798.

[142] R. Shapiro, S. Bratus, E. Rogers, and S. Smith, "Identifying Vulnerabilities in SCADA Systems via Fuzz-Testing," in *Proc. of Springer ICCIP*, Hanover, USA: Springer, 2011, pp. 57–72. URL: https://doi.org/10.1007/978-3-642-24864-1%5C_5.

[143] T. Shreedhar, R. Panda, S. Podanev, and V. Bajpai, "Evaluating QUIC Performance Over Web, Cloud Storage, and Video Workloads," *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 2, pp. 1366–1381, 2022. URL: https://doi.org/10.1109/TNSM.2021.3134562.

[144] J. Smith, P. Mittal, and A. Perrig, "Website Fingerprinting in the Age of QUIC," *Sciendo Proc. Priv. Enhancing Technol.*, vol. 2021, no. 2, pp. 48–69, 2021. URL: https://doi.org/10.2478/popets-2021-0017.

[145] P. F. Tehrani, E. Osterweil, J. H. Schiller, T. C. Schmidt, and M. Wählisch, "Security of Alerting Authorities in the WWW: Measuring Namespaces, DNSSEC, and Web PKI," in *Proc. of ACM WWW*, Virtual Event: ACM, 2021, pp. 2709–2720. URL: https://doi.org/10.1145/3442381.3450033.

[146] D. R. Thomas, R. Clayton, and A. R. Beresford, "1000 Days of UDP Amplification DDoS Attacks," in *Proc. of IEEE/APWG eCrime*, Phoenix, USA: IEEE, 2017, pp. 79–84. URL: https://doi.org/10.1109/ECRIME.2017.7945057.

[147] K. Tilling and J. A. Sterne, "Capture-Recapture Models Including Covariate Effects," *Oxford Amer. J. of Epidem.*, vol. 149, no. 4, pp. 392–400, 1999. URL: https://doi.org/10.1093/oxfordjournals.aje.a009825.

[148] T. J. de Toledo and N. M. Torrisi, "Encrypted DNP3 Traffic Classification Using Supervised Machine Learning Algorithms," *MPDI Mach. Learn. Knowl. Extr.*, vol. 1, no. 1, pp. 384–399, 2019. URL: https://doi.org/10.3390/make1010022.

[149] M. Trevisan, D. Giordano, I. Drago, M. M. Munafò, and M. Mellia, "Five Years at the Edge: Watching Internet From the ISP Network," *IEEE/ACM Trans. on Netw.*, vol. 28, no. 2, pp. 561–574, 2020. URL: https://doi.org/10.1109/TNET.2020.2967588.

[150] A. Valdes and S. Cheung, "Intrusion Monitoring in Process Control Systems," in *Proc. of IEEE HICSS*, Waikoloa, USA: IEEE, 2009, pp. 1–7. URL: https://doi.org/10.1109/HICSS.2009.273.

[151] E. Vasilomanolakis, S. Srinivasa, and M. Mühlhäuser, "Did you Really Hack a Nuclear Power Plant? An Industrial Control Mobile Honeypot," in *Proc. of IEEE CNS*, Florence, Italy: IEEE, 2015, pp. 729–730. URL: https://doi.org/10.1109/CNS.2015.7346907.

[152] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A Survey of Methods for Encrypted Traffic Classification and Analysis," *Wiley Int. J. Netw. Manag.*, vol. 25, no. 5, pp. 355–374, 2015. URL: https://doi.org/10.1002/nem.1901.

[153] P.-A. Vervier, O. Thonnard, and M. Dacier, "Mind Your Blocks: On the Stealthiness of Malicious BGP Hijacks," in *Proc. of ISOC NDSS*, San Diego, USA: Internet Society, 2015. URL: https://www.ndss-symposium.org/ndss2015/mind-your-blocks-stealthiness-malicious-bgp-hijacks.

[154] G. Wan, L. Izhikevich, D. Adrian, K. Yoshioka, R. Holz, C. Rossow, and Z. Durumeric, "On the Origin of Scanning: The Impact of Location on Internet-Wide Scans," in *Proc. of ACM IMC*, Virtual Event: ACM, 2020, pp. 662–679. URL: https://doi.org/10.1145/3419394.3424214.

[155] P. Wang, L. Wu, R. Cunningham, and C. C. Zou, "Honeypot Detection in Advanced Botnet Attacks," *Inderscience Int. J. Inf. Comput. Secur.*, vol. 4, no. 1, pp. 30–51, 2010. URL: https://doi.org/10.1504/IJICS.2010.031858.

[156] N. Weaver, C. Kreibich, and V. Paxson, "Redirecting DNS for Ads and Profit," in *Proc. of USENIX FOCI*, San Francisco, USA: USENIX, 2011. URL: https://www.usenix.org/conference/foci11/redirecting-dns-ads-and-profit.

[157] K. Wolsing, J. Rüth, K. Wehrle, and O. Hohlfeld, "A Performance Perspective on Web Optimized Protocol Stacks: TCP+TLS+HTTP/2 vs. QUIC," in *Proc. of ACM ANRW*, Montreal, Canada: ACM, 2019, pp. 1–7. URL: https://doi.org/10.1145/3340301.3341123.

[158] E. Wustrow, M. Karir, M. Bailey, F. Jahanian, and G. Huston, "Internet Background Radiation Revisited," in *Proc. of ACM IMC*, Melbourne, Australia: ACM, 2010, pp. 62–74. URL: https://doi.org/10.1145/1879141.1879149.

[159] Y. Zhang, L. Wang, W. Sun, R. C. Green II, and M. Alam, "Distributed Intrusion Detection System in a Multi-Layer Network Architecture of Smart Grids," *IEEE Trans. on Smart Grid*, vol. 2, no. 4, pp. 796–808, 2011. URL: https://doi.org/10.1109/TSG.2011.2159818.

[160] X. Zheng, C. Lu, J. Peng, Q. Yang, D. Zhou, B. Liu, K. Man, S. Hao, H. Duan, and Z. Qian, "Poison Over Troubled Forwarders: A Cache Poisoning Attack Targeting DNS Forwarding Devices," in *Proc. of USENIX Security*, Virtual Event: USENIX, 2020, pp. 577–593. URL: https://www.usenix.org/conference/usenixsecurity20/presentation/zheng.

[161] B. Zhu, A. D. Joseph, and S. Sastry, "A Taxonomy of Cyber Attacks on SCADA Systems," in *Proc. of IEEE iThings & CPSCom*, Dalian, China: IEEE, 2011, pp. 380–388. URL: https://doi.org/10.1109/iThings/CPSCom.2011.34.

[162] B. Zhu and S. Sastry, "SCADA-Specific Intrusion Detection/Prevention Systems: A Survey and Taxonomy," in *Proc. of SCS CPSWEEK 2010*, Stockholm, Sweden: Ptolemy TRUST, 2010. URL: https://ptolemy.berkeley.edu/projects/truststc/conferences/10/CPSWeek/papers/scs1_paper_8.pdf.

[163] J. Zirngibl, P. Buschmann, P. Sattler, B. Jaeger, J. Aulbach, and G. Carle, "It's over 9000: Analyzing Early QUIC Deployments with the Standardization on the Horizon," in *Proc. of ACM IMC*, Virtual Event: ACM, 2021, pp. 261–275. URL: https://doi.org/10.1145/3487552.3487826.

Other References

[164] J. Abley, O. Gudmundsson, M. Majkowski, and E. Hunt, "Providing Minimal-Sized Responses to DNS Queries That Have QTYPE=ANY," IETF, RFC 8482, Jan. 2019. URL: https://doi.org/10.17487/RFC8482.

[165] R. Aitchison, *Pro DNS and BIND 10 (Book)*. New York: Apress, 2011. URL: https://www.zytrax.com/books/dns/ch4/#authoritative.

[166] Allen-Bradley, "EtherNet/IP Secure Communication," Rockwell Automation, User Manual 1756-EN2TSC, 2015. URL: https://literature.rockwellautomation.com/idc/groups/literature/documents/um/enet-um003_-en-p.pdf.

[167] Avast Software s.r.o., *The Return of the Mirai Botnet*, Website, 2020. URL: https://blog.avast.com/return-of-mirai-botnet-avast.

[168] J. Baines, *RouterOS Post Exploitation*, Website, 2019. URL: https://medium.com/tenable-techblog/routeros-post-exploitation-784c08044790.

[169] D. J. Bernstein, *SYN Cookies*, Website, 1996. URL: https://cr.yp.to/syncookies.html.

[170] R. C. Bodenheim, "Impact of the Shodan Computer Search Engine on Internet-Facing Industrial Control System Devices," Air Force Institute of Technology, Ohio, Thesis, 2014. URL: https://core.ac.uk/download/pdf/277527601.pdf.

[171] CAIDA, *AS Rank API – Ranking of Autonomous Systems*, Website, 2005. URL: http://as-rank.caida.org/.

[172] CAIDA, *The UCSD Network Telescope*, Website, 2012. URL: http://www.caida.org/projects/network_telescope/.

[173] L. Casanova and A. Miroux, "Emerging Market Multinationals Report: 10 Years that Changed Emerging Markets," Emerging Markets Institute – Cornell University, Tech. Rep., 2020. URL: https://doi.org/10.7298/cvhn-dc87.

[174] Censys, *IO Search 2.0*, Website, 2017. URL: https://search.censys.io/.

[175] J. M. Ceron, J. J. Chromik, J. Santanna, and A. Pras, "Online Discoverability and Vulnerabilities of ICS/SCADA Devices in the Netherlands," arXiv.org, Tech. Rep. arXiv:2011.02019, 2019. URL: https://arxiv.org/abs/2011.02019.

[176] Cloudflare, *What is a QUIC Flood DDoS Attack? QUIC and UDP Floods.* Website, 2020. URL: https://www.cloudflare.com/ko-kr/learning/ddos/what-is-a-quic-flood/.

[177] Cloudflare Radar, *DDoS Attack Trends for Q4 2021*, Website, 2021. URL: https://radar.cloudflare.com/notebooks/ddos-2021-q4#network-layer-ddos-attacks-by-attack-rate.

[178] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," IETF, RFC 5280, May 2008. URL: https://doi.org/10.17487/RFC5280.

[179] D. Eastlake 3rd, "Domain Name System (DNS) IANA Considerations," IETF, RFC 6895, Apr. 2013. URL: https://doi.org/10.17487/RFC6895.

[180] D. Makrushin (Kaspersky Labs), *The Cost of Launching a DDoS Attack*, Website, 2017. URL: https://securelist.com/the-cost-of-launching-a-ddos-attack/77784/.

[181] J. Damas, M. Graff, and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))," IETF, RFC 6891, Apr. 2013. URL: https://doi.org/10.17487/RFC6891.

[182] G. Devarajan, *Unraveling SCADA Protocols: Using Sulley Fuzzer*, DefCon 15 Presentation, 2007. URL: https://infocon.org/cons/DEF%20CON/DEF%20CON%2015/DEF%20CON%2015%20presentations/DEF%20CON%2015%20-%20devarajan.pdf.

[183] W. Eddy, "TCP SYN Flooding Attacks and Common Mitigations," IETF, RFC 4987, Aug. 2007. URL: https://doi.org/10.17487/RFC4987.

[184] J. Eumann, R. Hiesgen, T. C. Schmidt, and M. Wählisch, "A Reproducibility Study of "IP Spoofing Detection in Inter-Domain Traffic"," arXiv.org, Tech. Rep. arXiv:1911.05164, 2019. URL: https://arxiv.org/abs/1911.05164.

[185] F5, Inc., *Our Roadmap for QUIC and HTTP/3 Support in NGINX*, Website, 2021. URL: https://www.nginx.com/blog/our-roadmap-quic-http-3-support-nginx/.

[186] Facebook, *How Facebook is Bringing QUIC to Billions*, Website, 2020. URL: https://engineering.fb.com/2020/10/21/networking-traffic/how-facebook-is-bringing-quic-to-billions/.

[187] Facebook Incubator, *mvfst*, Code Repository, 2019. URL: https://github.com/facebookincubator/mvfst/.

[188] Facebook Incubator, *mvfst – Commits*, Code Repository, 2019. URL: https://github.com/facebookincubator/mvfst/search?p=2&q=retry&type=commits.

[189] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," IETF, RFC 2267, Jan. 1998. URL: https://doi.org/10.17487/RFC2267.

[190] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," IETF, RFC 2827, May 2000. URL: https://doi.org/10.17487/RFC2827.

[191] D. Freedman, B. Foust, B. Greene, B. Maddison, A. Robachevsky, J. Snijders, and S. Steffann, "Mutually Agreed Norms for Routing Security (MANRS) Implementation Guide," RIPE, Document ripe-706, 2018. URL: https://www.ripe.net/publications/docs/ripe-706.

[192] A. Ghedini and V. Vasiliev, "TLS Certificate Compression," IETF, RFC 8879, Dec. 2020. URL: https://doi.org/10.17487/RFC8879.

[193] Google, *QUICHE*, Code Repository, 2018. URL: https://github.com/google/quiche/search?p=2&q=retry&type=commits.

[194] Google, *QUICHE – Commit: Internal change*, Code Repository, 2021. URL: https://github.com/google/quiche/commit/36d9a1fbff6e0f8665a1c60c09e19aa38380ae85.

[195] B. R. Greene, *Remote Triggered Black Hole (RTBH) Filtering*, Website, 2019. URL: http://www.senki.org/operators-security-toolkit/remote-triggered-black-hole-rtbh-filtering/.

[196] HAW, *Quiche Fork with Compression*, Code Repository, 2022. URL: https://github.com/josephnoir/quiche.

[197] P. Hoffman and P. McManus, "DNS Queries over HTTPS (DoH)," IETF, RFC 8484, Oct. 2018. URL: https://doi.org/10.17487/RFC8484.

[198] P. Hoffman, A. Sullivan, and K. Fujiwara, "DNS Terminology," IETF, RFC 8499, Jan. 2019. URL: https://doi.org/10.17487/RFC8499.

[199] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)," IETF, RFC 7858, May 2016. URL: https://doi.org/10.17487/RFC7858.

[200] C. Huitema, *A Simple Test of DDOS Attacks on QUIC*, Website, 2020. URL: https://huitema.wordpress.com/2020/09/22/a-simple-test-of-ddos-attacks-on-quic/.

[201] G. Huston and J. Damas, *Measuring Recursive Resolver Centrality*, RIPE Meeting 82 Presentation, 2021. URL: https://ripe82.ripe.net/wp-content/uploads/presentations/41-2021-05-19-resolver-centrality.pdf.

[202] J. Iyengar and I. Swett, "QUIC Loss Detection and Congestion Control," IETF, RFC 9002, May 2021. URL: https://doi.org/10.17487/RFC9002.

[203] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," IETF, RFC 9000, May 2021. URL: https://doi.org/10.17487/RFC9000.

[204] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," IETF, Internet-Draft – work in progress 09, Jan. 2018. URL: https://datatracker.ietf.org/doc/draft-ietf-quic-transport/09.

[205] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," IETF, Internet-Draft – work in progress 10, Mar. 2018. URL: https://datatracker.ietf.org/doc/draft-ietf-quic-transport/10.

[206] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," IETF, Internet-Draft – work in progress 15, Oct. 2018. URL: https://datatracker.ietf.org/doc/draft-ietf-quic-transport/15.

[207] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," IETF, Internet-Draft – work in progress 33, Jan. 2021. URL: https://datatracker.ietf.org/doc/draft-ietf-quic-transport/33.

[208] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," IETF, Internet-Draft – work in progress 01, Jan. 2017. URL: https://datatracker.ietf.org/doc/draft-ietf-quic-transport/01.

[209] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," IETF, Internet-Draft – work in progress 02, Mar. 2017. URL: https://datatracker.ietf.org/doc/draft-ietf-quic-transport/03.

[210] D. Jackson, *DNS Amplification Variation Used in Recent DDoS Attacks*, Website, 2009. URL: https://www.secureworks.com/research/dns-amplification.

[211] M. Jonker, *A First Joint Look at DoS Attacks and BGP Blackholing in the Wild*, RIPE 79 Presentation, 2019. URL: https://ripe78.ripe.net/archives/video/22/.

[212] T. King, C. Dietzel, J. Snijders, G. Doering, and G. Hankins, "BLACKHOLE Community," IETF, RFC 7999, Oct. 2016. URL: https://doi.org/10.17487/RFC7999.

[213] A. Kirkham, "Issues with Private IP Addressing in the Internet," IETF, RFC 6752, Sep. 2012. URL: https://doi.org/10.17487/RFC6752.

[214] J. Klick, S. Lau, D. Marzin, J.-O. Malchow, and V. Roth, "Internet-Facing PLCs – A New Back Orifice," Black Hat USA, White Paper, 2015. URL: https://www.blackhat.com/docs/us-15/materials/us-15-Klick-Internet-Facing-PLCs-A-New-Back-Orifice-wp.pdf.

[215] O. Kolkman, W. Mekking, and R. Gieben, "DNSSEC Operational Practices, Version 2," IETF, RFC 6781, Dec. 2012. URL: https://doi.org/10.17487/RFC6781.

[216] B. Krebs, *KrebsOnSecurity Hit With Record DDoS*, Website, 2016. URL: https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos.

[217] W. Kumari and D. McPherson, "Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF)," IETF, RFC 5635, Aug. 2009. URL: https://doi.org/10.17487/RFC5635.

[218] O. Kupreev, E. Badovskaya, and A. Gutnikov, *DDoS Attacks in Q4 2018*, Website, 2019. URL: https://securelist.com/ddos-attacks-in-q4-2018/89565/.

[219] C. Labovitz, *Tracing Volumetric DDoS to its Booter / IPHM Origins*, NANOG 82 Presentation, 2021. URL: https://storage.googleapis.com/site-media-prod/meetings/NANOG82Virtual/2368/20210611_Labovitz_Tracing_Ddos_End-To-End_v1.pdf.

[220] Let's Encrypt, *Let's Encrypt's Hierarchy as of August 2021*, Website, 2021. URL: https://letsencrypt.org/certificates/.

[221] M. McKeay (Akamai), *Summer SOTI - DDoS by the Numbers*, Website, 2018. URL: https://blogs.akamai.com/2018/06/summer-soti---ddos-by-the-numbers.html.

[222] M. Prince (Cloudflare), *The DDoS That Almost Broke the Internet*, Website, 2013. URL: https://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet/.

[223] D. Madory and T. Dua, *RE: Prefix Hijacking, How to Prevent and Fix Currently*, NANOG Mail Archive, 2014. URL: https://seclists.org/nanog/2014/Aug/513.

[224] Marcin Nawrocki, *Dismantling Operational Practices of BGP Blackholing at IXPs*, Talk at RIPE 79, 2019. URL: https://ripe79.ripe.net/archives/video/212/.

[225] Marcin Nawrocki, *Down the Black Hole: Dismantling Operational Practices of BGP Blackholing at IXPs*, Website, 2019. URL: https://labs.ripe.net/author/marcin_nawrocki/down-the-black-hole-dismantling-operational-practices-of-bgp-blackholing-at-ixps/.

[226] Marcin Nawrocki, *Down the Black Hole: Dismantling Operational Practices of BGP Blackholing at IXPs*, Website, 2019. URL: https://blog.apnic.net/2019/12/18/down-the-black-hole-dismantling-operational-practices-of-bgp-blackholing-at-ixps/.

[227] Marcin Nawrocki, *Down the Black Hole: Dismantling Operational Practices of BGP Blackholing at IXPs*, Talk at MIX Salottino, 2020. URL: https://mix-it.net/.

[228] Marcin Nawrocki, *QUICsand: Quantifying QUIC Reconnaissance Scans and DoS Flooding Events*, Talk at DUST, the International Workshop on Darkspace and UnSolicited Traffic Analysis, 2021. URL: https://www.caida.org/workshops/dust/2107/.

[229] Marcin Nawrocki, *Tracing the DDoS Attack Ecosystem from the Internet Core*, Website, 2022. URL: https://blog.apnic.net/2022/04/28/tracing-the-ddos-attack-ecosystem-from-the-internet-core/.

[230] Marcin Nawrocki, *Transparent Forwarders: An Unnoticed Component of the Open DNS Infrastructure*, Talk at APRICOT, 2022. URL: https://2022.apricot.net/program/schedule-conference/#/day/11/security-operations.

[231] Marcin Nawrocki, *On the Interplay between TLS Certificates and QUIC Performance*, Talk at IETF 116 MAPRG, 2023. URL: https://datatracker.ietf.org/meeting/116/session/maprg/.

[232] Marcin Nawrocki, *On the Interplay between TLS Certificates and QUIC Performance*, Website, 2023. URL: https://blog.apnic.net/2023/01/16/on-the-interplay-between-tls-certificates-and-quic-performance/.

[233] Marcin Nawrocki, *Revisiting QUIC Handshakes and TLS Deployment: About Three Challenges*, Talk at IETF 116 QUICWG, 2023. URL: https://datatracker.ietf.org/meeting/116/session/quic/.

[234] Marcin Nawrocki, *SoK: A Data-Driven View on Methods to Detect Reflective Amplification DDoS Attacks Using Honeypots*, Talk at Sixth Annual Cambridge Cybercrime Centre Conference, 2023. URL: https://www.cambridgecybercrime.uk/conference2023.html.

[235] P. Marques, N. Sheth, R. Raszuk, B. Greene, J. Mauch, and D. McPherson, "Dissemination of Flow Specification Rules," IETF, RFC 5575, Aug. 2009. URL: https://doi.org/10.17487/RFC5575.

[236] Matthias Wählisch and Marcin Nawrocki, *QUICsand: Quantifying QUIC Reconnaissance Scans and DoS Flooding Events*, Talk at IETF 113, MAPRG, 2022. URL: https://datatracker.ietf.org/meeting/113/session/maprg.

[237] J. Mattsson, *Background on the 3x Anti-Amplification Limit*, IETF Mail Archive, 2021. URL: https://mailarchive.ietf.org/arch/msg/quic/RdeQ_y4dHLzufgtXYccFPBiqrUQ/.

[238] J. Mauch, *Spoofing ASNs*, NANOG Mail Archive, 2013. URL: http://seclists.org/nanog/2013/Aug/132.

[239] MAWI Working Group, *Packet Traces from WIDE Backbone*, Traffic Archive, 1999. URL: http://mawi.wide.ad.jp/mawi/.

[240] B. Meixell and E. Forner, "Out of Control: Demonstrating SCADA Exploitation," Black Hat USA, White Paper, 2013. URL: https://media.blackhat.com/us-13/US-13-Forner-Out-of-Control-Demonstrating-SCADA-WP.pdf.

[241] Microsoft, *quicreach*, Code Repository, 2022. URL: https://github.com/microsoft/quicreach/.

[242] J. S. (Microworks), *Deploying Secure DNP3 (IEEE 1815) - What You Need to Know*, DistribuTech Presentation, 2016. URL: https://trianglemicroworks.com/docs/default-source/referenced-documents/deploying-secure-dnp3-dtech-2016.pdf.

[243] Miek Gieben, *DNS Library in Go*, Code Repository, 2010. URL: https://github.com/miekg/dns.

[244] MIT Technology Review, *Industrial Control Systems are Still Vulnerable to Malicious Cyberattacks*, Website, 2019. URL: https://www.technologyreview.com/2019/01/28/137704/industrial-control-systems-are-still-vulnerable-to-malicious-cyberattacks/.

[245] P. Mockapetris, "Domain Names - Implementation and Specification," IETF, RFC 1035, Nov. 1987. URL: https://doi.org/10.17487/RFC1035.

[246] C. Morales, *NETSCOUT Arbor Confirms 1.7 Tbps DDoS Attack; The Terabit Attack Era Is Upon Us*, Website, 2018. URL: https://asert.arbornetworks.com/netscout-arbor-confirms-1-7-tbps-ddos-attack-terabit-attack-era-upon-us/.

[247] J. Mücke, M. Nawrocki, R. Hiesgen, P. Sattler, J. Zirngibl, G. Carle, T. C. Schmidt, and M. Wählisch, "Waiting for QUIC: On the Opportunities of Passive Measurements to Understand QUIC Deployments," arXiv.org, Tech. Rep. arXiv:2209.00965, 2022. URL: https://arxiv.org/abs/2209.00965.

[248] Mushorg (Honeynet Project), *CONPOT ICS/SCADA Honeypot*, Code Repository, 2013. URL: https://github.com/mushorg/conpot.

[249] M. Nawrocki, M. Wählisch, T. C. Schmidt, C. Keil, and J. Schönfelder, "A Survey on Honeypot Software and Data Analysis," arXiv.org, Tech. Rep. arXiv:1608.06249, 2016. URL: http://arxiv.org/abs/1608.06249.

[250] NETSCOUT, *13th Annual Worldwide Infrastructure Security Report – Insight into the Global Threat Landscape*, Website, 2019. URL: https://www.netscout.com/resources/threat-report-archives/13th-worldwide-infrastructure-security-report.

[251] NETSCOUT, *14th Annual Worldwide Infrastructure Security Report – Cloud in the Crosshairs*, Website, 2019. URL: https://www.netscout.com/sites/default/files/2019-03/SECR_005_EN-1901%E2%80%93WISR.pdf.

[252] Nexusguard Reasearch, *Could QUIC Turn into the Next Most Prevalent Amplification Attack Vector?* Website, 2020. URL: https://blog.nexusguard.com/could-quic-turn-into-the-next-most-prevalent-amplification-attack-vector.

[253] NTOP, *nDPI – Open and Extensible LGPLv3 Deep Packet Inspection Library*, Website, 1998. URL: https://www.ntop.org/products/deep-packet-inspection/ndpi/.

[254] E. Osterweil, P. F. Tehrani, T. C. Schmidt, and M. Wählisch, "From the Beginning: Key Transitions in the First 15 Years of DNSSEC," arXiv.org, Tech. Rep. arXiv:2109.08783, 2021. URL: https://arxiv.org/abs/2109.08783.

[255] P. McManus (Fastly), *Does the QUIC Handshake Require Compression to be Fast?* Website, 2020. URL: https://www.fastly.com/blog/quic-handshake-tls-compression-certificates-extension-study.

[256] Pandas, *Computational Tools*, Website, 2019. URL: https://pandas.pydata.org/pandas-docs/stable/user_guide/computation.html#exponentially-weighted-windows.

[257] PeeringDB, *A Freely Available, User-Maintained, Database of Networks, and the go-to Location for Interconnection Data*, Website, 2004. URL: https://www.peeringdb.com/.

[258] D. Piscitello, *IP Prefix Squatting Attacks*, Website, 2011. URL: https://securityskeptic.typepad.com/the-security-skeptic/2011/06/ip-prefix-squatting-attacks.html.

[259] PJM Interconnection, "DNP SCADA over Internet with TLS Security," PJM Interconnection, Jetstream Guide, 2017. URL: https://www.pjm.com/-/media/etools/jetstream/jetstream-guide.ashx?la=en.

[260] N. Provos and T. Holz, *Virtual Honeypots – From Botnet Tracking to Intrusion Detection (Book)*, 2nd. Boston, USA: Addison–Wesley, 2008. URL: https://dl.acm.org/doi/book/10.5555/1408351.

[261] J. Quittek, S. Bryant, B. Claise, P. Aitken, and J. Meyer, "Information Model for IP Flow Information Export," IETF, RFC 5102, Jan. 2008. URL: https://doi.org/10.17487/RFC5102.

[262] Rauchgeist, *DDoS Angriff*, Social Media Post, 2018. URL: https://www.facebook.com/324164061331608/posts/liebe-leute-leider-sind-wir-opfer-einer-ddos-attacke-bitte-habt-keine-sorge-wege/605379806543364/.

[263] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," IETF, RFC 8446, Aug. 2018. URL: https://doi.org/10.17487/RFC8446.

[264] RIPE NCC, *RIPE Routing Information Service (RIS)*, Website, 1999. URL: http://www.ripe.net/projects/ris/rawdata.html.

[265] Russian Embassy UK, *Cyberattacks on the Russian Embassy Website*, Social Media Post, 2019. URL: https://twitter.com/russianembassy/status/1163803718342475776.

[266] F. J. Ryba, M. Orlinski, M. Wählisch, C. Rossow, and T. C. Schmidt, "Amplification and DRDoS Attack Defense – A Survey and New Perspectives," arXiv.org, Tech. Rep. arXiv:1505.07892, 2015. URL: http://arxiv.org/abs/1505.07892.

[267] J. Ryburn, *DDoS Mitigation Using BGP Flowspec*, NANOG 63 Presentation, 2015. URL: https://archive.nanog.org/sites/default/files/tuesday_general_ddos_ryburn_63.16.pdf.

[268] Schneider Electric USA, "MODBUS/TCP Security," Schneider Electric, Protocol Specification MB-TCP-Security-v21, 2018. URL: https://modbus.org/docs/MB-TCP-Security-v21_2018-07-24.pdf.

[269] Sectigo Limited, *Certificate Search, ID 3958242236*, Website, 2015. URL: https://crt.sh/?id=3958242236.

[270] Sectigo Limited, *Certificate Search, ID 9314791*, Website, 2015. URL: https://crt.sh/?id=9314791.

[271] M. Seeman, *QUIC Interop Runner*, Website, 2019. URL: https://interop.seemann.io/?test=s.

[272] Y. Sheffer, R. Holz, and P. Saint-Andre, "Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS)," IETF, RFC 7457, Feb. 2015. URL: https://doi.org/10.17487/RFC7457.

[273] Shodan, *The World's First Search Engine for Internet-Connected Devices*, Website, 2014. URL: https://www.shodan.io/.

[274] S. Strowes, *Visibility of IPv4 and IPv6 Prefix Lengths in 2019*, Website, 2019. URL: https://labs.ripe.net/Members/stephen_strowes/visibility-of-prefix-lengths-in-ipv4-and-ipv6.

[275] T. Shani (Imperva), *Updated: This DDoS Attack Unleashed the Most Packets Per Second Ever. Here's Why That's Important*, Website, 2019. URL: https://www.imperva.com/blog/this-ddos-attack-unleashed-the-most-packets-per-second-ever-heres-why-thats-important/.

[276] T. Werner (Honeynet Project), *Honeytrap – Transport Layer Honeypot*, Code Repository, 2006. URL: https://github.com/tillmannw/honeytrap/.

[277] Talos Intelligence, *New VPNFilter Malware Targets at least 500K Networking Devices Worldwide*, Website, 2018. URL: https://blog.talosintelligence.com/2018/05/VPNFilter.html.

[278] The Shadowserver Foundation, *Open Resolver Scanning Project*, Website, 2013. URL: https://scan.shadowserver.org/dns/.

[279] The Shadowserver Foundation, *Scanning Project*, Website, 2022. URL: https://scan.shadowserver.org/.

[280] The Verge, *The Man Behind a Spree of Gaming Network Cyberattacks has Pleaded Guilty*, Website, 2018. URL: https://www.theverge.com/2018/11/7/18071764/austin-thompson-derptrolling-sony-blizzard-game-ddos-arrest-guilty.

[281] M. Thomson and S. Turner, "Using TLS to Secure QUIC," IETF, RFC 9001, May 2021. URL: https://doi.org/10.17487/RFC9001.

[282] M. Thomson, *Space for Packet Metadata*, IETF Mail Archive, 2018. URL: https://mailarchive.ietf.org/arch/msg/quic/wzrNfxRCwwgfw499Dh0YsHYTSoI/.

[283] Troy Mursch (Badpackets), *Quasi Networks Responds as We Witness the Death of The Master Needler*, Website, 2017. URL: https://badpackets.net/quasi-networks-responds-as-we-witness-the-death-of-the-master-needler-80-82-65-66-for-now/.

[284] TUM, *QScanner*, Code Repository, 2021. URL: https://github.com/tumi8/QScanner.

[285] D. Turk, "Configuring BGP to Block Denial-of-Service Attacks," IETF, RFC 3882, Sep. 2004. URL: https://doi.org/10.17487/RFC3882.

[286] Q. Vohra and E. Chen, "BGP Support for Four-octet AS Number Space," IETF, RFC 4893, May 2007. URL: https://doi.org/10.17487/RFC4893.

[287] M. Wählisch, A. Vorbach, C. Keil, J. Schönfelder, T. C. Schmidt, and J. H. Schiller, "Design, Implementation, and Operation of a Mobile Honeypot," arXiv.org, Tech. Rep. arXiv:1301.7257, 2013. URL: http://arxiv.org/abs/1301.7257.

[288] L. Wei and J. S. Heidemann, "Whac-A-Mole: Six Years of DNS Spoofing," arXiv.org, Tech. Rep. arXiv:2011.12978, 2020. URL: https://arxiv.org/abs/2011.12978.

[289] K. Wilhoit, "Who's Really Attacking Your ICS Equipment?" Trend Micro, White Paper, 2013. URL: https://paper.bobylive.com/Meeting_Papers/BlackHat/Europe-2013/bh-eu-13-whose-really-attacking-wilhoit-wp.pdf.

[290] M. M. Winn, "Constructing Cost-Effective and Targetable ICS Honeypots Suited for Production Networks," Air Force Institute of Technology, Ohio, Thesis, 2015. URL: https://core.ac.uk/download/pdf/277528034.pdf.

[291] E. Winstead, *Response Rate Limiting with BIND*, APRICOT 37 Presentation, 2014. URL: https://conference.apnic.net/data/37/apricot-2014-rrl_1393309768.pdf.

[292] T. Yardley, *ICS Security Tools, Tips, and Trade*, Code Repository, 2015. URL: https://github.com/ITI/ICS-Security-Tools.