Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Evaluating Network Security Using Internet-wide Measurements

Oliver Gasser

Dissertation

Network Architectures and Services

NET-2019-05-1

TECHNISCHE UNIVERSITÄT MÜNCHEN

FAKULTÄT FÜR INFORMATIK

Evaluating Network Security Using Internet-wide Measurements

Oliver Gasser

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

DOKTORS DER NATURWISSENSCHAFTEN

genehmigten Dissertation.

Vorsitzende/-r: Prof. Dr.-Ing. Jörg Ott

Prüfende/-r der Dissertation:

1. Prof. Dr.-Ing. Georg Carle
2. Prof. Anja Feldmann, Ph. D.

Die Dissertation wurde am 4. März 2019 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 10. Mai 2019 angenommen.

EVALUATING NETWORK SECURITY USING INTERNET-WIDE MEASUREMENTS

OLIVER GASSER

Acknowledgements

This work would not have been possible without the continuous support and help by many people. I would like to thank them. First of all I would like to thank my supervisor Prof. Georg Carle who supported me in my research efforts and provided guidance through all these years. I would also like to thank Prof. Anja Feldmann for being my second examiner and Prof. Jörg Ott for chairing the examination committee.

I would also like to thank my Bachelor's and Master's theses advisors Lothar and Ralph. You two got me interested in network measurement research and which lit the initial flame for this Ph. D. Additionally, I would like to thank my paper co-authors Quirin, Ralph, Paweł, Qasim, Maciej, Stephen, Luuk, Sebastian, Theodor, Johanna, Lexi, Thomas, Matthias, Benjamin, Max, Benedikt, Carl, Nadja, Felix, Taejoong, Jens, Johannes, Roland, Oliver, Dave, Alan, Patricia, Connor, Narseo, Rubén, Christian, Florian, Ángel, Minoo, Patrick, Timm, Lothar, Felix, Helmut, Johann, Andreas, Quentin, Ernst, and Georg for the always interesting discussions and fun we had when writing papers. I would also like to thank Vaibhav Bajpai for providing feedback on this dissertation. And of course, thanks to Marc-Oliver for bringing joy, snacks, and blinking lights into our office.

Finally, I would like to thank my loved ones. My love Doris, thank you for supporting me through these years and encouraging me to not wait until the last minute to write it all up. My parents Barbara and Richard and my siblings Juliane and Markus for providing the support I needed and being there for me.

Abstract

With the Internet being ubiquitous in many aspects of our daily lives, it is paramount to ensure the security of Internet services as well as the devices connected to the network. With security issues being reported in the news more frequently than ever, we as researchers must play an active role in making the Internet more secure. One suitable way to evaluate the security of Internet services, devices, and protocols are active network measurements.

In this thesis we present methods and tools to tackle current challenges in Internet measurements, conduct security analyses of protocols covering three different areas, and provide public measurement services for others to use.

First, we enhance the fast port scanning tool ZMap to make it IPv6-capable and provide several probe modules for it. This enhanced version—ZMapv6—allows for the first time to conduct Internet-scale measurements in the IPv6 Internet. We use this tool in several of our measurements studies to identify responsive hosts. We also develop goscanner, a tool designed to perform more complex protocol exchanges in large-scale measurements. goscanner's implementation leverages multi-core processing architectures to e. g. efficiently download TLS certificates.

Next, we perform three security measurement studies covering three different types of devices: web servers, building automation devices, and out-of-band management devices.

In our web server study we perform Internet-wide HTTPS measurements using ZMap, ZMapv6, and goscanner covering more than 190 M domain names. We evaluate the deployment of different HTTPS security techniques and find contrasting rates of deployment: HTTP Strict Transport Security sees much higher deployment compared to HTTP Public Key Pinning. We show that higher deployment numbers correlate to lower deployment effort and lower risk to availability. In addition, we find that there are still several insecure certificates in Certificate Transparency logs.

By conducting measurements with ZMap and ZMapv6 for the building automation protocol BACnet we find 16 k publicly reachable BACnet devices. These devices not only endanger the privacy of their owners but they also pose a threat to other Internet users. We demonstrate that the BACnet protocol is vulnerable to amplification attacks—reaching a similar amplification factor as open DNS resolvers.

Moreover, we scan the out-of-band management protocol IPMI. IPMI provides remote access to systems and e. g. allows to boot powered-off machines, change their boot order, or access their console. Using a new scanning technique we uncover IPMI devices even if they have IPMI-over-IP disabled. Also for IPMI devices we uncover security shortcomings such as weak keys or factory-default certificates.

Finally, as part of our IPv6 measurement studies we provide the IPv6 Hitlist Service where researchers can access up-to-date measurement data. Providing access to this data allows fellow researchers to conduct additional security measurements and analyses which in turn strengthens the security of the Internet.

Zusammenfassung

Da das Internet immer präsenter in vielen Aspekten unseres täglichen Lebens ist, ist es von höchster Wichtigkeit, die Sicherheit von Internetdiensten und mit dem Internet verbundenen Geräten zu gewährleisten. Nachdem immer häufiger von Sicherheitsproblemen berichtet wird, müssen wir als Wissenschaftler eine aktive Rolle spielen, indem wir das Internet sicherer machen. Ein geeigneter Weg um die Sicherheit von Internetdiensten, -geräten und -protokollen auszuwerten sind aktive Messungen.

In dieser Arbeit werden Methoden und Werkzeuge präsentiert, um aktuelle Herausforderungen im Gebiet Internetmessungen zu meistern, Sicherheitsanalysen von Protokollen aus drei verschiedenen Gebieten durchgeführt und öffentlich verfügbare Messdienste für andere zugänglich gemacht.

Im ersten Teil der Arbeit erweitern wir den Port-Scanner ZMap um IPv6-Fähigkeiten und fügen mehrere IPv6-Messmodule hinzu. Diese erweiterte Version – ZMapv6 – erlaubt es erstmals Internet-weite Messungen im IPv6-Internet durchzuführen. Dieses Werkzeug wird in mehreren unserer Messstudien verwendet, um antwortende Hosts zu finden. Außerdem entwickeln wir mit goscanner ein Werkzeug, das es erlaubt, komplexe Protokollaustausche in großflächigen Messungen durchzuführen. Die Implementierung von goscanner nutzt Multicore-Prozessor-Architekturen geschickt aus, um z.B. TLS-Zertifikate effizient herunterzuladen.

Im zweiten Teil werden drei Sicherheitsmessungen durchgeführt, die drei verschiedene Gerätetypen abdecken: Webserver, Gebäudeautomatisierungs- und Out-of-band-Managemenet-Geräte.

In der Webserver-Studie werden Internet-weite HTTPS-Messungen mit ZMap, ZMapv6 und goscanner durchgeführt, die mehr als 190 M Domainnamen abdecken. Durch die Auswertung der Verbreitung von verschiedenen HTTPS-Sicherheitsmechanismen werden kontrastierte Verbreitungsraten sichtbar: HTTP Strict Transport Security ist weit häufiger verbreitet als HTTP Public Key Pinning. Wir zeigen, dass hohe Verbreitung mit niedrigem Bereitstellungsaufwand und niedrigem Erreichbarkeitsrisiko einhergeht. Zusätzlich zeigen Certificate-Transparency-Messungen, dass es immer noch mehrere unsichere gültige TLS-Zertifikate gibt.

Als Ergebnis von Messungen mit ZMap und ZMapv6, die das Gebäudeautomatisierungsprotokoll BACnet zum Ziel hatten, finden wir 16 k öffentlich erreichbare BACnet-Geräte. Diese Geräte sind nicht nur eine Gefahr für die Privatsphäre der Inhaber, sondern auch für andere Teilnehmer des Internets. Wir zeigen nämlich, dass das BACnet-Protokoll verwundbar für Verstärkungsangriffe ist – es werden ähnliche Verstärkungsfaktoren erreicht wie mit offenen DNS-Resolvern.

Zudem vermessen wir das Out-of-band-Management-Protokoll IPMI. IPMI stellt entfernten Zugriff auf Systeme bereit und erlaubt z.B. ausgeschaltete Systeme zu starten, deren Boot-Reihenfolge zu verändern, oder auf deren Konsole zuzugreifen. Mit einer neuen Messtechnik können IPMI-Geräte entdeckt werden, obwohl sie IPMI-over-IP deaktiviert haben. Auch für IPMI-Geräte decken wir Sicherheitsmängel wie schwache Schlüssel und die Verwendung von Standardzertifikaten auf.

Abschließend stellen wir als Teil unserer IPv6-Messungen den “IPv6 Hitlist Service” mit aktuellen Messdaten für Forscher zur Verfügung. Durch Zugriff auf diese Daten können andere Forscher weitere Sicherheitsmessungen und -analysen durchführen, was wiederum die Sicherheit des Internets stärkt.

Contents

Part I

Introduction

Chapter 1

Introduction

In this chapter we introduce the reader to the dissertation. First, we give a motivation for the research topic's importance by explaining the research goal, providing background on the state of the art, and highlighting important research aspects. Second, we present the research questions that will be answered in this thesis and provide an in-depth description for each research question. Third, we describe our main contributions and link them to research questions and thesis chapters. Fourth, we provide an outline of this dissertation's structure.

1.1 Motivation

The Internet has become more and more ubiquitous in the past years in our everyday life. The increasing reliance of our society on IT and the Internet specifically, lead to it being classified as a critical infrastructure [35]. Additionally, Internet devices and networks are both (mostly unintentionally) attackers and victims of Denial-of-Service type attacks [149, 150, 193, 194]. These attacks are getting more and more sophisticated and increase their bandwidth continuously.

To counter these attacks, we can take reactive and proactive measures. Reactive measures include deploying DDoS protection services [6, 54, 113, 137] or blackholing traffic to certain target IP addresses [64, 65] to mitigate currently running attacks. Proactive measures can consist of conducting penetration tests and performing security audits to find weak points in the infrastructure or hardening own networks e. g. by deploying firewalls with strict rule sets.

From a research perspective we can contribute to proactive security by finding vulnerable devices, services, or networks in the Internet, *before* an attack occurs.

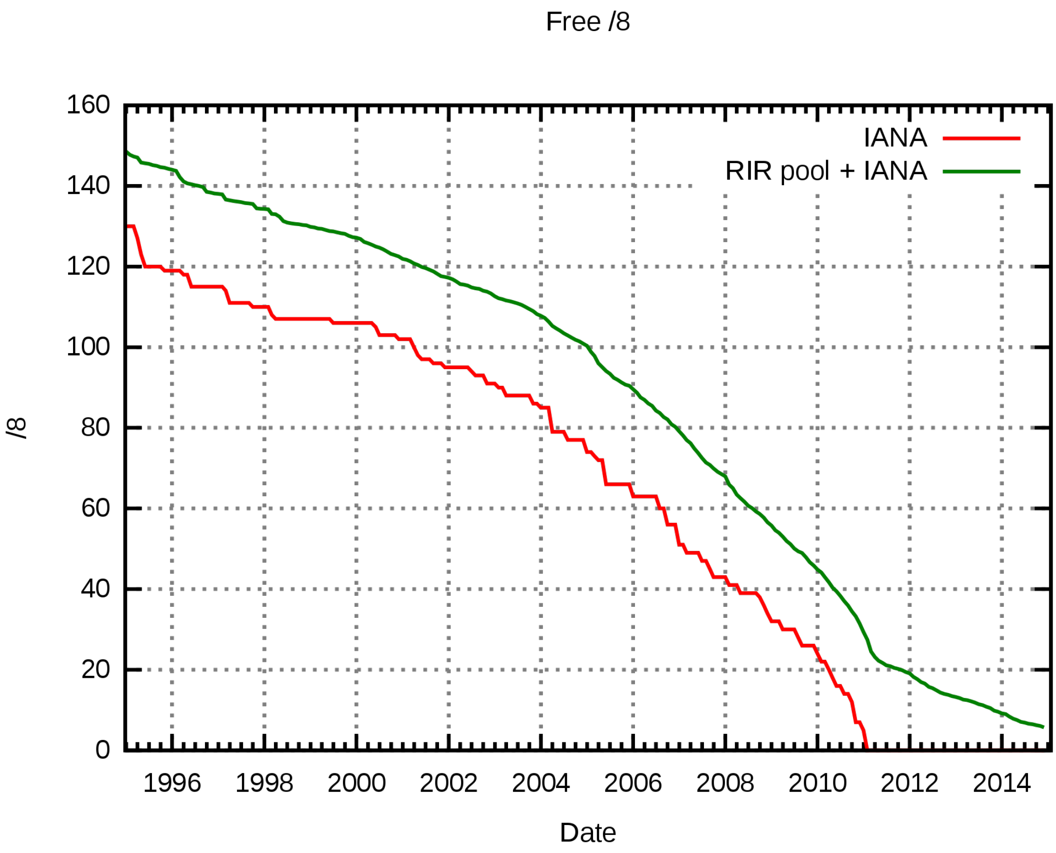


Figure 1.1: Exhaustion of IPv4 /8 networks since 1995. Created by Mro [175], CC-BY-SA 4.0 [58].

To identify these vulnerable devices, services, or networks we can perform active measurements, i. e., measurements where we actively send packets and inspect the responses to determine the security status of probed entities. To assess the actual scale of security issues we perform Internet-wide measurements, i. e., measurements covering the complete Internet.

The original version 4 of the Internet Protocol [191] has long had issues such as the exhaustion of available IP addresses shown in Figure 1.1. To remediate this limitation a new Internet Protocol was introduced, IPv6 [63]. Since its initial standardization more than 20 years ago [62], IPv6 deployment has been slow to gain traction. In recent years, however, pushed by initiatives such as the “World IPv6 Launch” [139] the IPv6 deployment has grown steadily, reaching more than 25 % in January 2019 (see Figure 1.2).

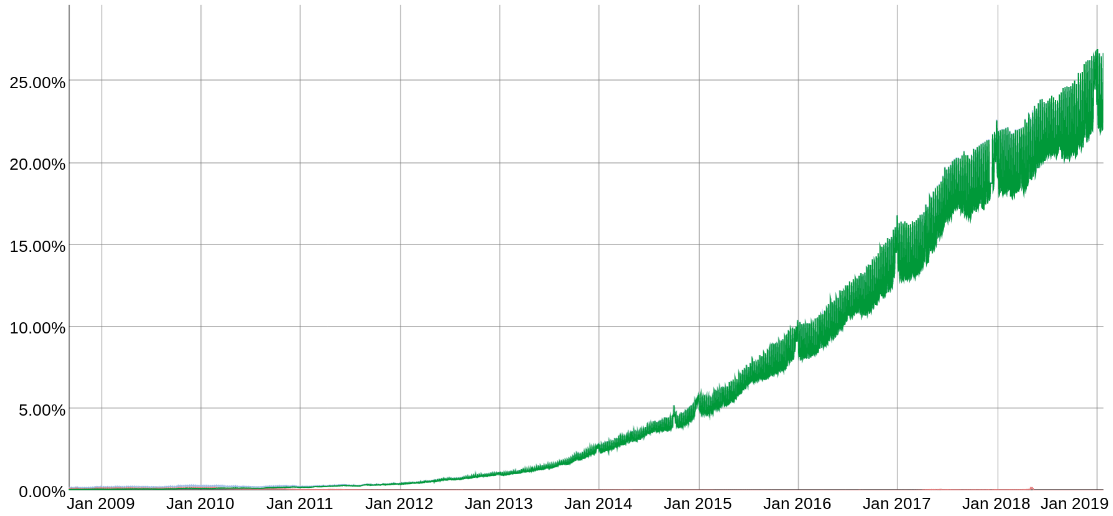


Figure 1.2: Global IPv6 adoption over time as measured by Google [112].

This increase in IPv6 deployment and the potential differences in the IPv4 and IPv6 Internet [21, 59, 68] needs to be reflected in how we conduct Internet measurements as well. Researchers can no longer focus solely on the IPv4 Internet and ignore the IPv6 counterpart. They need to consider the IPv6 Internet in their measurement design and execution, especially considering the large geographical differences in IPv6 deployment as shown in Figure 1.3. To paint an accurate picture of the IPv4 and IPv6 Internet, we therefore want to conduct measurements of both protocols whenever possible and feasible.

In addition to improving measurement methodology in this thesis, we strive to conduct measurements for three important protocols in the Internet: HTTPS, BACnet, and IPMI.

HTTPS and its certificate ecosystem is an important cornerstone of the Web. It provides security, privacy, and integrity through the use of cryptographic primitives and the certificate ecosystem. This ecosystem, however, has been under siege for several years by security issues and compromises [15, 73, 129, 183, 195]. In response to these security threats several HTTPS security techniques have been developed. In this thesis we want to investigate the deployment and security gains attained through the use of these techniques.

Internet-of-Things (IoT) devices most of the time provide a direct interface from the virtual world to the real world. Consequently, changes in these devices can directly affect physical assets and people. When deploying connected IoT devices in the Internet in addition to taking care of security issues, we need to take safety issues seriously as well.

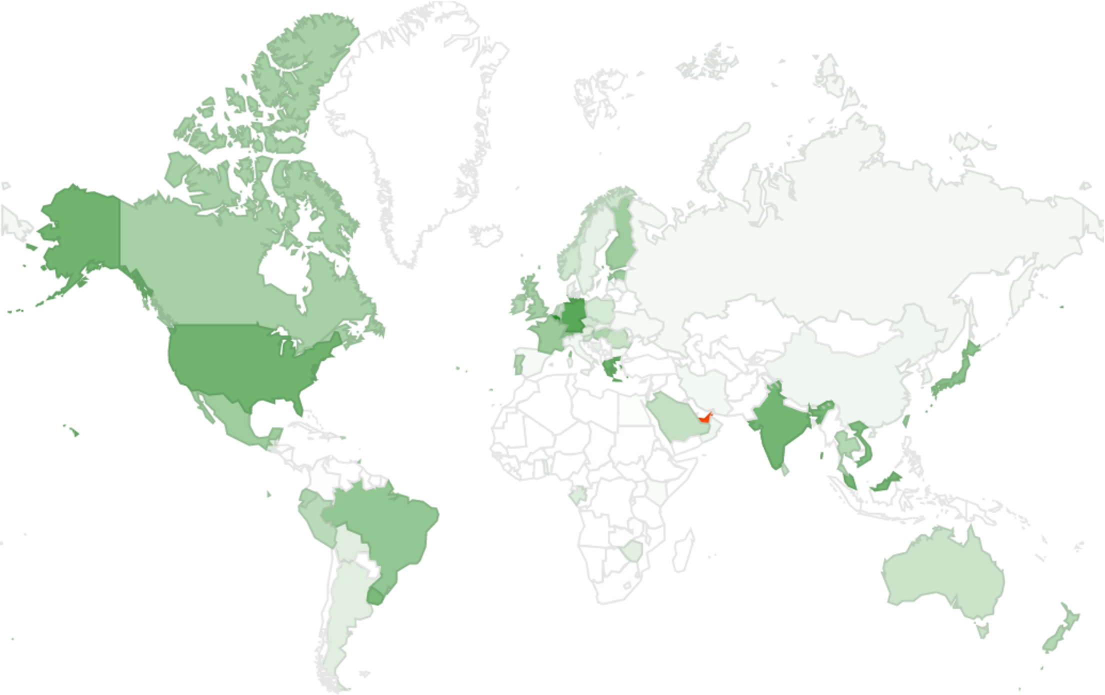


Figure 1.3: Per country IPv6 adoption as measured by Google [112]. Green countries experience no IPv6-related issues (darker green means higher IPv6 deployment); orange (high deployment) and red (low deployment) countries experience IPv6-related issues.

Unfettered access to these devices could lead to harming material assets (e. g. facilitating break-ins through presence detection by monitoring the heating within people's homes) or even people (e. g. remotely manipulating the heating). As a result, the security of IoT devices and protocols is critical to ensure people's well-being. As an exponent of the ever-growing Internet-of-Things (IoT) ecosystem we select the building automation protocol BACnet [19], which we analyze in-depth.

Remote management of systems such as rack-mounted servers is an important job for system administrators. Protocols such as IPMI [138] facilitate these managements tasks and allow to remotely reboot a machine, change boot devices, or access a system's console. The IPMI-over-IP extension makes it possible to use IPMI over the Internet. As IPMI has seen security vulnerabilities in the past [118], we choose to survey the IPMI population and conduct a rigorous security analysis of found devices.

By improving Internet measurement methodology in the IPv4 and IPv6 world and conducting in-depth analysis of three security critical protocols, we strive to push the state of the art of Internet security measurements in this dissertation.

1.2 Research Questions

In this section we present the research questions (RQs) which this dissertation tries to answer.

The main research questions are as follows:

I. How can we perform Internet-scale measurements in the IPv6 Internet?

II. How biased are address sources for IPv6 hitlists?

III. Are HTTPS servers still vulnerable to Man-in-the-Middle attacks?

IV. Are BACnet devices vulnerable to amplification attacks?

V. Are IPMI devices vulnerable to Man-in-the-Middle attacks?

In the following we elaborate on each research question in detail. For each research question we identify challenges which need to be tackled in order to provide adequate conditions for answering the research question.

RQ I: How can we perform Internet-scale measurements in the IPv6 Internet?

To assess the security of deployed Internet services using active measurements various methodologies and tools are available. In this research question we evaluate different methodologies regarding their aptitude to evaluate Internet service security in the scope of applying Internet measurements on a global scale and their usefulness to perform measurements in the IPv6 Internet. These methodologies can be split into two groups: (1) port-based measurement and (2) protocol-based measurement methods. In the first category we find tools which perform port scans. These tools are built with high-bandwidth measurements in mind and use “fire and forget” methodologies. In the second category we find tools which perform more complex scans consisting of protocol exchanges. Due to the need of interaction between sender and receiver in a protocol exchange, these tools need to keep state and at the same time be efficient enough to perform Internet-scale measurements.

Therefore, we identify challenge 1 (C 1) which needs to be tackled:

Challenge 1: Develop a tool to conduct Internet-scale port-based measurements in IPv6

With tools such as ZMap we can perform port-based measurements on the complete IPv4 address space within hours [71] or even minutes [4]. When looking at the IPv6 Internet and the vast address space that comes with it, this brute-force approach is infeasible. We therefore propose to investigate the feasibility of IPv6 Internet-scale measurements. To tackle this challenge we survey currently available tools, discuss their limitations, and present tools suitable to face this challenge.

To answer the second part of RQ I, we need to tackle C 2:

Challenge 2: Develop a tool to conduct Internet-scale protocol-based measurements

In addition to performing stateless port-based measurements, establishing stateful connections to facilitate protocol-based exchanges is a useful methodology to gain more insight into deployed services and devices, e. g. by downloading and evaluating TLS certificates from HTTPS servers. In tackling this challenge we explore abstract methods of designing fast protocol-based measurement tools and present concrete implementations to perform protocol-based measurements on an Internet-wide scale.

By tackling these two challenges we build the foundation to answer RQ I, which provides the methodological foundation for the following in-depth analyses.

RQ II: How biased are address sources for IPv6 hitlists?

The IPv6 Internet brings new opportunities for clients as well as service providers. With its increasing adoption, however, new challenges arise for the Internet measurement community. As IPv6 becomes more prevalent in today's Internet, the Internet measurement community needs to investigate security issues in the IPv4 as well as the IPv6 Internet. Due to the expansive IPv6 address space, brute-force scanning as done in IPv4 is not feasible in the IPv6 Internet. Therefore, we use lists of IPv6 addresses, so-called hitlists to specify targets for measurements in the IPv6 Internet. These hitlists come with own challenges (e. g. which sources do we use for hitlist addresses).

To answer this research question we need to tackle the following four challenges:

Challenge 3: Extract targets for IPv6 hitlists from passive address sources

We leverage addresses found in passive measurements such as flow data from an Internet Exchange Point to extract IPv6 addresses. Extracting addresses from passive measurement data comes with a few peculiarities. To tackle this challenge we investigate the distribution of IPv6 addresses extracted from passive sources, classify them according to their communication behavior, and investigate address responsiveness. As a result we provide valuable insights for the research community on how to best use passive address sources for IPv6 hitlists.

Challenge 4: Extract targets for IPv6 hitlists from active address sources

In addition to addresses from passive sources, we perform active measurements to extract IPv6 addresses from active sources. These active sources pose challenges themselves, as e. g. addresses might exhibit different response behavior. After tackling this challenge researchers know how to best use active sources to gather targets for IPv6 measurements.

Challenge 5: Evaluate balancedness and biases of address sources for IPv6 hitlists

Identifying potential IPv6 hitlist sources is only part of the equation. Another crucial piece of the puzzle is to analyze potential hitlist sources to identify imbalanced sources (i. e., sources with most addresses from just a few networks). Moreover, biases due to address assignment strategies (e. g. assigning all possible addresses within a prefix) can significantly influence measurements based on IPv6 hitlists. By tackling this challenge we can assess the balancedness of hitlists sources and can reduce biases in IPv6 hitlists, which leads to more accurate results in IPv6 measurement studies.

Challenge 6: Provide IPv6 Hitlist Service

In addition to conducting regular measurements, we want to explore the possibility of providing a measurement service to fellow researchers in this challenge. This measurement service could provide researchers with up-to-date measurement results which they can then use in their own security research.

By tackling these four challenges we can provide an answer to RQ II in conducting an evaluation of biases in IPv6 hitlists.

RQ III: Are HTTPS servers still vulnerable to Man-in-the-Middle attacks?

HTTPS is an important cornerstone in the Internet ecosystem. It provides security for web-based protocols such as HTTP as well as email protocols such as SMTP or IMAP. In this research question we evaluate the security of the HTTPS ecosystem against Man-in-the-Middle attacks by (1) analyzing HTTPS security techniques and (2) leveraging Certificate Transparency to inspect the security and quality of TLS certificates.

While working on these two parts of the research question we tackle the following two challenges:

Challenge 7: Evaluate deployment of HTTPS security extensions

To counter the misissuance of certificates by a small group of certificate authorities and the resulting risk of Man-in-the-Middle attacks to the HTTPS ecosystem, a number of

TLS and HTTPS security extensions have been proposed in the last years. To find out whether today's Internet is secure from these attacks, we perform a large-scale analysis of the deployment and correct usage of these security extensions. To tackle this challenge we evaluate the success of two header-based HTTPS security techniques in-depth and compare them to other security extensions.

Challenge 8: Evaluate security of certificates in Certificate Transparency logs

Another important security technique in the HTTPS ecosystem is Certificate Transparency (CT), which has been heavily pushed in the last few years. Since CT provides a repository of issued certificates, we want to use this in order to assess the security and quality of issued certificates to tackle this challenge. In addition, we shed light on the rise of CT over time and evaluate some of its lesser known aspects such as gossiping. By tackling this challenge we can assess the risk of Man-in-the-Middle attacks stemming from insecure certificates.

After tackling these two challenges we are able to provide an answer to RQ III by analyzing the security of HTTPS extensions and certificates.

RQ IV: Are BACnet devices vulnerable to amplification attacks?

In the age of pervasive Internet-of-Things devices, also building automation systems are getting more and more connected. When connecting building automation devices to the Internet, two immediate challenges arise: (1) can these devices be accessed by unwanted parties and (2) can these devices be misused (e. g. by taking part in amplification attacks). In this research question we investigate whether BACnet building automation devices can be misued in amplification attacks by malicious actors.

During the work on this research question we tackle the following three challenges:

Challenge 9: Evaluate deployment of publicly reachable BACnet devices

As building automation systems can contain sensitive information about buildings and their inhabitants their security is crucial. In addition to leaking sensitive information, these devices could be remotely manipulated to e. g. changing the room temperature or opening security access doors. Due to the critical nature of these devices we want to understand if BACnet building automation devices can be reached from the public Internet. In addition, to tackle this challenge we categorize the BACnet deployment typologically, geographically, and topologically.

Challenge 10: Evaluate amplification attack potential of BACnet devices

After we have evaluated whether BACnet devices are reachable from the public Internet, we want to evaluate the danger posed by the misuse of these devices. As has been shown by the Mirai botnet in 2016, connected Internet devices pose a threat by unknowingly being part of Denial-of-Service attacks [149]. In this challenge we want to analyze if BACnet devices can be misused in these types of attacks as well, more specifically in amplification attacks. Moreover, we want to assess the impact of these attacks when using BACnet devices.

Challenge 11: Evaluate impact of notification campaign targeting BACnet devices

Notifying affected parties after discovering security faults has the potential to ameliorate the security situation. By tackling this challenge we explore the effect of notification campaigns on the number of publicly reachable BACnet devices.

After tackling these three challenges we can provide an answer to RQ IV by analyzing the vulnerability of BACnet for amplification attacks.

RQ V: Are IPMI devices vulnerable to Man-in-the-Middle attacks?

IPMI is a protocol for out-of-band management and allows for full remote access of a machine. Using this protocol we can e. g. reboot a system, monitor input on the terminal, and install a new system image. Due to these extensive possibilities offered by IPMI, it is important that IPMI devices are segregated in separate networks. In this research question we want to assess the security of IPMI devices with regard to Man-in-the-Middle attacks.

During the work on this research question we tackle the following two challenges:

Challenge 12: Evaluate deployment of publicly reachable IPMI devices

Most rack-mounted servers have some form of integrated IPMI out-of-band management device. Thanks to extensions of the IPMI standard, IPMI devices can be contacted over IP. This feature, however, can be disabled by system administrators. In tackling this challenge we evaluate the possibility to detect IPMI devices more effectively using different measurement techniques and evaluate the deployment of publicly reachable IPMI devices.

Challenge 13: Evaluate TLS security of IPMI devices

After the evaluation of found public IPMI devices, we want to investigate the TLS security of these found devices to tackle this challenge. This allows to better assess the danger of these devices being taken over and misused by attackers in Man-in-the-Middle attacks.

After tackling these two challenges we are able to provide an answer to RQ V by analyzing the vulnerability of IPMI devices to Man-in-the-Middle attacks.

1.3 Key Contributions of This Thesis

In this section we highlight one key contributions for each research question of this thesis. In Table 1.1 we show a mapping between each highlighted key contribution to the respective research question and thesis chapter.

Table 1.1: Linking highlighted key contributions to research questions and thesis chapters.

Key contribution	Research question	Thesis chapter
ZMapv6	RQ I	Chapter 3
IPv6 Hitlist Service	RQ II	Chapter 4
HTTPS security techniques	RQ III	Chapter 5
BACnet amplification	RQ IV	Chapter 6
IPMI security	RQ V	Chapter 7

In the following we briefly describe each highlighted key contribution:

ZMapv6 We create the first Internet-scale scanning tool for the IPv6 Internet and make it publicly available. ZMapv6 lays the foundation for our IPv6 measurements as it brings ZMap's fire-and-forget technique to the IPv6 Internet. By combining ZMapv6 with IPv6 hitlists we can efficiently conduct security studies in the IPv6 Internet.

IPv6 Hitlist Service In addition to our in-depth analysis of IPv6 hitlists, we provide a service with responsive IPv6 addresses and aliased prefixes for researchers to use. The published data is based on daily IPv6 measurements and is therefore updated regularly. Dozens of researchers have access to the IPv6 Hitlist Service and use it to further evaluate and enhance the Internet's security.

HTTPS security techniques Using data from Internet-wide scans we analyze the deployment of various HTTPS security techniques. We find that most deployments are configured correctly, although there are significant differences in the number of deployed systems depending on the security technique. We finally correlate risk and effort to deployment and find that techniques with low risk and low effort such as HSTS find higher deployment compared to high risk and high effort techniques (e. g. HPKP).

BACnet amplification We analyze the building automation protocol BACnet and uncover its vulnerability to amplification attacks. We show that publicly reachable BACnet devices not only pose a threat to their own privacy, security, and safety but that they can be misused as amplifiers by malicious actors. Our analysis shows that BACnet devices can reach a similar amplification factor as open DNS resolvers.

IPMI security We evaluate the deployment of the out-of-band management protocol IPMI by scanning for publicly reachable IPMI devices. We uncovered a new scanning method using RMCP Ping requests to identify IPMI devices with IPMI-over-IP disabled. Additionally, we find that the security of these IPMI devices is quite lacking—we find many instances with devices offering short keys and factory-default certificates, in addition to IPMI's general security vulnerabilities.

In addition to these key contributions, we would like to highlight 3 of the 13 challenges listed in Section 1.2, as tackling these 3 challenges was especially difficult or impactful.

Tackling challenge 5 allowed us to better understand balancedness and biases of address sources for IPv6 hitlists. This laid the necessary ground work for providing daily measurement data in our IPv6 Hitlist Service to the measurement community.

In solving challenge 7 we showed the correlation between the successful rollout of a security technique and its risk to availability and deployment effort. By showing this correlation we highlight that successful introduction of a new security technique depends heavily on its ease-of-use and associated risk.

With tackling challenge 10 we added another protocol to the list of protocols vulnerable to amplification attacks—BACnet. This shows that active measurements can play a positive role by pointing out protocol weaknesses before they are discovered by malicious actors.

1.4 Structure of This Thesis

The thesis is structured analogous to the outline of the research questions.

We present previous works related to this dissertation in the upcoming Chapter 2. In this chapter we compare this dissertation to important research in this field. Third-party publications which are related to a specific part of this dissertation only, are discussed in that specific dissertation chapter. Together with this introductory chapter, the related work survey completes the first part of this thesis.

In the second part of the dissertation we focus on Internet measurement methodology. Chapter 3 presents methodology and tools to assess the security of deployed Internet services using active measurements. In doing so this chapter gives answers to RQ I. In Chapter 4 we discuss challenges and solutions to Internet-scale measurements in the IPv6 Internet. It addresses RQ II by evaluating address sources for IPv6 hitlists and analyzing their biases. This concludes the second part of this dissertation.

The three subsequent chapters in part three of this dissertation detail results from concrete active measurement campaigns undertaken during this work. First, we evaluate the security of the HTTPS ecosystem in Chapter 5. By investigating HTTPS security techniques and certificates in Certificate Transparency logs we give answers to RQ III. Second, Chapter 6 surveys the building automation protocol BACnet and its potential misuse in amplification attacks. With the BACnet analysis research we answer RQ IV and its challenges. Third, Chapter 7 addresses RQ V by evaluating the reachability of the out-of-band management protocol IPMI. These three chapters serve as case studies of how active network measurements can be used to understand the security state of deployed Internet services.

Finally, this thesis concludes in Chapter 8 where we provide a summary and give pointers for future work.

Chapter 2

Related Work

In this chapter we discuss work related to the general topic of this dissertation. Publications which are related to specific sub-topics of this dissertation are examined in the respective topic's chapter, i. e., Sections 3.1.1 and 3.2.5 for work related to network measurement methodology (Chapter 3), Section 4.1 for work related to measurements in the IPv6 Internet (Chapter 4), Sections 5.1.2 and 5.2.2 for work related to HTTPS ecosystem security (Chapter 5), Section 6.9 for work related to BACnet security evaluations (Chapter 6), and Section 7.2 for work related to our IPMI security analyses (Chapter 7).

In the following sections we discuss four publications related to the general topic of this dissertation and compare them to the scope of this dissertation. Table 2.1 gives an overview of these related works and compares them to this dissertation.

Table 2.1: Comparison of this dissertation to related work regarding fulfillment of the posed research questions (see Section 1.2). ✓ signifies that the research question has been fulfilled, ❍ means that it has been partially fulfilled, and ✗ means that the research question has not been fulfilled or was out-of-scope for this work.

	RQ I	RQ II	RQ III	RQ IV	RQ V
Holz [131]	✗	✗	✓	✗	✗
Durumeric [70]	✓	✗	✓	❍	✗
Fiebig [91]	✓	✓	✗	✗	✗
Hendriks [122]	✓	✓	✗	✗	✗
This dissertation	✓	✓	✓	✓	✓

2.1 Analysis of TLS Public Key Infrastructure

In 2014, Holz published a thorough analysis of three Public Key Infrastructures (PKIs) [131]: the X.509 PKI for TLS most commonly used in the Web's HTTPS ecosystem, the OpenPGP Web of Trust which is used in PGP-encrypted email, and the SSH key distribution mechanism. In the following we focus on the author's findings regarding the X.509 PKI as it is most relevant to this dissertation.

To analyze the empirical state of the X.509 PKI the author conducted multiple active and passive measurements. Contrary to this dissertation, Holz's HTTPS measurements did not cover the whole Internet, i. e., they were not Internet-wide. Additionally, the measurements were conducted over IPv4 only, probably also due to the lower deployment of IPv6 at the time.

Holz found that the security of the HTTPS PKI was in a dismal state: Many certificates had too short keys or insecure signature algorithms, were self-signed, or did not provide a correct chain to a trusted root certificate. In this dissertation we confirm that many of Holz's findings are still valid today, although we see some improvements in the HTTPS ecosystem due to higher awareness of involved parties and stricter enforcement of rules.

Building automation protocols and out-of-band management protocols which are analyzed for their security in this dissertation, were out-of-scope of Holz's dissertation.

2.2 Fast IPv4-wide Security Measurements

Durumeric's seminal work on fast Internet-wide Measurements was published in 2017 [70]. With his work the author pushed the envelope on fast network measurements applied to security measurements.

He created the tool ZMap which allowed for fast port scans covering the complete IPv4 address space in under an hour [71]. The author of this dissertation built ZMapv6 [244], an IPv6-enabled version of Durumeric's ZMap tool. Durumeric himself did not conduct Internet-scale measurements on the IPv6 Internet.

Durumeric performed multiple IPv4-wide measurements and conducted several analyses of the TLS ecosystem: Similar to Holz [131] he uncovered issues with the HTTPS ecosystem, in addition to key generation flaws in TLS certificates, security weaknesses in email protocol deployments, and surveying the Heartbleed vulnerability [75].

In his dissertation Durumeric briefly touched on the insecurity of industrial control system devices including BACnet, but did not provide a thorough analysis. The out-of-band management protocol IPMI was out-of-scope of Durumeric's dissertation.

Finally, Durumeric presented Censys [72] which provides researchers with the possibility of getting up-to-date measurement results, similar to the IPv6 Hitlist Service presented in this dissertation.

2.3 Finding Security Misconfigurations with DNS Measurements

In 2017 Fiebig published his dissertation on security misconfigurations through the use of DNS-based measurements [91]. One of his main contributions was his work on using IPv6 reverse DNS (rDNS) to identify IPv6 addresses.

In his work he contributed tools to the systematic walking of the rDNS tree in IPv6. Additionally, he evaluated the rDNS dataset for misconfigurations and evaluated its security. He identified IPv6 misconfigurations which can not be mitigated by operators. He provides concrete suggestions on how to improve the DNS deployment and avoid security misconfigurations. Generally, Fiebig finds that rDNS zones are commonly well maintained. In this dissertation we use part of Fiebig's IPv6 results to analyze the usefulness of this data source for IPv6 hitlists.

Fiebig neither investigated the security of the HTTPS protocol nor BACnet or IPMI devices, as these were out-of-scope of his dissertation.

2.4 IPv6 Security Measurements

In his dissertation published in 2019, Hendriks investigated several IPv6-specific issues regarding resilience and security [122]. He performed empirical measurements of IPv6-specific threats to assess the actual severity of these issues.

He proposed a novel techniques to find IPv6 hosts vulnerable to DDoS attacks without conducting active measurements. In addition, he developed the zesplot tool which helps to visualize the expansive IPv6 address space. zesplot was also used in the IPv6 hitlist analysis conducted by the author of this dissertation [100].

Moreover, Hendriks analyzed the threats of inadequate handling of IPv6 Extension Headers. He found faulty IPv6 implementations which made it possible to (1) hide traffic from network operators and (2) evade firewall rules. Hendriks also found IPv6-

specific misconfigurations in the DNS by evaluating two years of DNS data from several large DNS zones.

To counter these threats, Hendriks proposed actionable ways to identify and prevent such mistakes in the future. In addition, he provided an online service to provision measurements inspecting firewall configurations in your own network.

The analysis of HTTPS, BACnet, and IPMI were out-of-scope of Hendriks's dissertation.

Part II

Internet Measurement Methodology

Chapter 3

Network Measurement Methodology

In this chapter we present and evaluate methodologies to conduct Internet-wide network measurements to assess the security of deployed Internet services. First, we evaluate tools and methodologies addressing new challenges arising from the increasing deployment of IPv6. Second, we develop and evaluate tools to conduct large-scale active measurements with protocol exchanges. Third, we summarize the chapter with our key contributions to this dissertation.

3.1 IPv6 Measurement Methods and Tools

Since the IPv6 address space [63] is much more expansive than the IPv4 address space [191], measurement methodologies differ significantly between the two.

Probing the complete IPv4 Internet can nowadays be done in under one hour [4]. This is made possible by tools such as ZMap [71] and masscan [114] who can scan the Internet at a line rate of 10 Gbit/s. In contrast to older tools such as nmap [162], these tools leverage a stateless architecture and circumvent the kernel space by implementing their custom IP stack. This allows them to achieve these high packet rates and makes the brute-force scanning of all IPv4 address space possible.

For IPv6, however, this approach is not feasible, since the IPv6 address space is more than 10^{28} times larger than the IPv4 address space. Therefore it was clear from the beginning that the measurements in the IPv6 Internet needed a different approach.

This section is based on two papers [100, 104] which were written with co-authors Quirin Scheitle, Sebastian Gebhard, Georg Carle, Pawel Foremski, Qasim Lone, Maciej Korczynski, Stephen D. Strowes, and Luuk Hendriks.

3.1.1 Related Work

We review related work on IPv6 measurements, which can be categorized into deployment analysis, analysis of IPv6 address structures, DNS-based methods to gain addresses, security measurements focusing on IPv6, and measurement software.

Deployment Analysis

The deployment of IPv6 clients and services as well as the IPv6 topology was analyzed by several researchers since 2008.

Malone [164] used data from server logs and performed active traceroutes to obtain IPv6 addresses. By analyzing the interface IDs of these addresses he finds that the majority of addresses in server logs are EUI-64 addresses and manually configured “low” addresses (addresses with only some of the last bits set to one). For routers found in traceroute results he finds about 90 % of “low” addresses.

Czyz *et al.* [60] measured IPv6 adoption in 2014 with active measurements sourced from datasets such as the Alexa Top 1M list and passive traffic traces. They find that 95 % percent of IPv6 traffic is HTTP and HTTPS and therefore user traffic.

Plonka and Berger [189] used one year of server logs from Akamai to classify client IPv6 addresses. They find that only 4 % of addresses are stable for more than four days. Additionally, only 1 % of addresses were EUI-64 addresses, of which a majority was moving between /64 prefixes.

In 2018, Beverly *et al.* [27] conducted a thorough analysis of strategies for discovering topologies in IPv6. They presented their tool Yarrp6 [44] which performed stateless randomized traceroute to obtain good performance and circumvent ICMP rate-limiting by routers. The authors then used several sources, including TUM’s IPv6 hitlist [242], as Yarrp6 input in order to discover IPv6 topologies. They discover more than 1.3 M IPv6 router interfaces and try to identify subnet boundaries.

Address Structure Analysis

To find new IPv6 addresses researchers leveraged structural properties to find “adjacent” addresses and exploit similar addressing schemes.

Ullrich *et al.* [249] used rule mining, but could find only a few hundred IPv6 addresses.

Foremski *et al.* [96] presented Entropy/IP, an approach which uses machine learning based on the entropy of collected IPv6 addresses to generated a model and output new potential addresses.

Similarly, Murdock *et al.* [176] presented 6Gen to find dense IPv6 address regions and leverage this information to generate neighboring addresses.

Plonka and Berger [188] proposed to use kIP for sharing IPv6 addresses in a privacy-preserving way. To achieve that they harness the structure from IPv6 addressing schemes.

DNS-Based Methods

The information found in the DNS was long known to be a valuable source for IPv6 research [111].

Strowes [232] discovered almost 1 M IPv6 addresses by leveraging the IPv4 rDNS tree to gather domains which he then resolved for IPv6. The majority of the found addresses were responsive to probing.

Fiebig *et al.* [92, 93] on the other hand, walked the IPv6 rDNS tree directly to learn 2.8 M IPv6 addresses. They did, however, not answer the question, whether the found addresses are responsive.

This question was later only partially answered by Borgolte *et al.* [31], who additionally used NSEC walking in DNSSEC-signed zones and evaluated differences in security configurations between IPv4 and IPv6 hosts.

IPv6 Security Measurements

Czyz *et al.* [59] and Borgolte *et al.* [31] both evaluated security of dual-stack devices and found that IPv6 performs worse compared to IPv4. They argued that this might be due to a lack of deployed IPv6 firewall rules, which results in more services being reachable over IPv6 unknowingly.

Scanning Software

Similar to general purpose software, also scanning software went trough a process of adopting IPv6 in addition to IPv4.

The versatile scanning tool nmap [162] supports IPv6 since 2002. Scanning software which focuses more on speed such as ZMap [71] and masscan [114] are not yet IPv6-ready. Therefore, we extend ZMap to make it IPv6-ready, which we publish as ZMapv6 [244].

3.1.2 ZMapv6

In order to make the IPv4-only ZMap ready for measurements in the IPv6 Internet we had to introduce several changes [245] in the source code:

Sending We had to adapt the sending procedure in order to being able to send IPv6 packets instead of IPv4 ones. In addition, we had to implement reading IPv6 addresses from a target file, as the default brute-force scanning for IPv4 was not feasible for IPv6. Finally, we also had to adapt the generation of the validation bytes from IPv4 to IPv6. This function is used to set bytes in the IPv6 header of sent packets and will then be checked in the received packets to ensure that they belong to a running ZMapv6 scan.

Receiving On the receiving side we added the possibility to decode packets with IPv6 and ICMPv6 headers. Again, the modified validation function was applied to received IPv6 packets to ensure that they indeed belong to this IPv6 scan. Additional validation and output generation was implemented in the respective probe modules.

Probe modules We added several probe modules which can be used to perform IPv6 measurements. See below for more details on the IPv6 probe modules.

Configuration We added configuration options specific for running IPv6 measurements.

- The parameter `ipv6-target-file` allows to specify a text file which contains a list of IPv6 addresses to be scanned, separated by newlines. This parameter activates the IPv6 mode of ZMap and signals that IPv6 packets will be sent and received. If this parameter is not present, ZMap will run in the regular IPv4 mode.
- The parameter `ipv6-source-ip` is used to specify the IPv6 source address which will be used when sending packets. This obligatory option is necessary as interfaces can have many associated addresses in IPv6.

IPv6 Probe Modules

After we adapted the ZMap core to being IPv6-ready, we implemented four probe modules for IPv6 scanning:

IPv6 TCP SYN module This module sends an IPv6 packet containing a TCP segment with the specified destination port and the SYN bit set. This is interpreted by receivers listening on this destination port as the first packet of the TCP three-way handshake upon which they respond with an IPv6 packet containing a TCP segment with both the SYN and ACK bits set. If the target is not listening on this port it will respond with a TCP packet with the RST flag set. Alternatively, if the last-hop router does not find the target it responds with an ICMPv6 target unreachable packet. All these responses can be decoded by this module and written to the output file. The sent packet includes bytes in the header which can be used to attribute responses to this scan: Four validation bytes are stored in the TCP initial sequence number and four more bytes are used in the selection of the source port.

IPv6 UDP module This module sends an IPv6 packet containing a UDP datagram with the specified destination port and payload. The payload can be either specified as hex values or by reading a binary file. If the target is listening on the destination port and can interpret the sent payload correctly it may respond with a protocol dependent UDP payload itself. If the target is not listening on this port it will respond with an ICMPv6 port unreachable packet. If the last-hop router does not find the target it responds with an ICMPv6 target unreachable packet. All these responses can be decoded by this module and written to the output file. The sent packet includes bytes in the header which can be used to attribute responses to this scan: Four validation bytes are used in the selection of the source port.

IPv6 UDP DNS module This module is a specialized DNS module specifically for finding recursive resolvers. If you want to send regular DNS probes, please use the regular IPv6 UDP module with an appropriate DNS payload. The sent packet includes bytes in the header which can be used to attribute responses to this scan: Four validation bytes are used in the selection of the source port.

ICMPv6 echo request module This module sends an IPv6 packet containing an ICMPv6 packet with type 128 and code 0, i. e., an echo request packet. If a target is listening for ICMPv6 echo request packets and it is configured to answer them, it will send an ICMPv6 packet with type 129 and code 0, i. e., an echo reply packet. It might also respond with various ICMPv6 error messages such as destination unreachable. All these responses can be decoded by this module and written to the output file. The sent packet includes bytes in the header and payload which can be used to attribute responses to this scan: Eight validation bytes are stored in the ICMPv6 payload and two more bytes are used in the ICMPv6 ID number.

Users of ZMapv6 can choose the appropriate probing module according to the measurement that they want to perform. For a measurement of the QUIC protocol in the IPv6 Internet for example, the user would choose the IPv6 UDP module with measurements conducted on port 443 and a QUIC payload.

Additional ZMapv6 Features

Recently, we added two more features to ZMapv6: reading from stdin [246] and adding the scanning address to the pcap filter [247].

The former was a feature requested by Austin Murdock who already used ZMapv6 in a study on finding unknown IPv6 addresses [176]. It allows to read target IPv6 address from standard input by specifying `-` as parameter for the `ipv6-target-file` option.

The latter feature automatically adds the IPv6 scanning address specified through `ipv6-source-ip` to the pcap filter. This ensures that only packets with this IPv6 address would be processed by the ZMapv6 receive code. This feature is especially useful if multiple ZMapv6 scans are run concurrently each with its own IPv6 address. This improves performance as each ZMapv6 instance only needs to evaluate packets destined to itself and can ignore packets belonging to other ZMapv6 instances.

Code Availability

The latest version of ZMapv6 is available on GitHub [244].

3.1.3 IPv6 Hitlist

With the goal of better understanding the IPv6 Internet, we conduct two large measurement studies of the IPv6 Internet using the concept of IPv6 hitlists [100, 104]. These studies will be detailed in-depth in Chapter 4.

3.2 Large-Scale Protocol-Based Measurements

We also investigate methodologies to perform large-scale protocol-based measurements.

The main difference between conducting port scans (e. g. testing whether port 80 is open using ZMap) and protocol-based scans (e. g. downloading a file from a web server using curl) is the architecture and thus performance of these two categories of tools.

Since port scans do not involve a complicated handshake no state needs to be stored by the measurement tool. This allowed to development of fast scanning tools which bypass the operating system kernel, e. g. ZMap or masscan.

Protocol-based measurements on the other side involve more complex interactions such as the TCP three way handshake or the exchange of mutually interdependent protocol messages such as in the TLS handshake. Therefore, protocol-based measurement tools mostly make use of the operating system kernel to handle connection establishment and teardown and use libraries to facilitate the exchange of protocol messages.

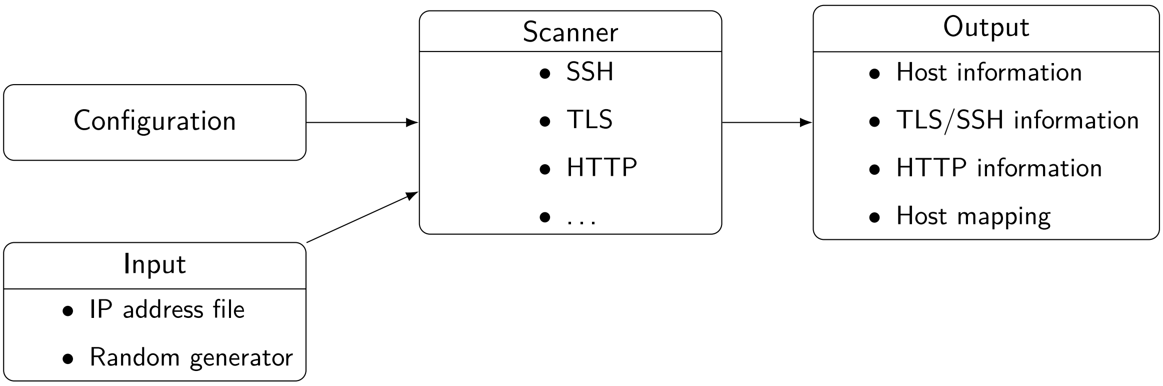


Figure 3.1: Architecture overview of goscanner.

In this section we present goscanner, a protocol-based measurement tool which can perform TLS, HTTPS, and SSH measurements.

3.2.1 Requirements

goscanner's main goal is to be able to conduct protocol-based measurements on a large-scale. It should support measuring SSH and TLS (including exchange of HTTP information), and its design should be extensible to allow for additional protocol support in the future. goscanner should also support connecting to hosts over IPv4 as well as IPv6. By "large-scale" we mean to conduct measurements regularly (e. g. weekly) for all known hosts (i. e., more than 160 M connections in IPv4, and about 25 M connections in IPv6, as of December 2018). As goscanner's main intended purpose is to perform security measurements, the output should contain security relevant data. These include cryptographic host keys for SSH servers, TLS certificates for TLS servers, and security-related HTTP headers (e. g. HSTS [127], HPKP [83]).

3.2.2 Architecture

The architecture of goscanner is modular in order to facilitate the addition of more measurement protocols in the future.

Figure 3.1 shows an overview of goscanner's architecture. We now elaborate on each part of the architecture in more detail:

Configuration The configuration module allows users of goscanner to customize the tool's behavior and adapt it to your system. Most importantly, you can specify the type of measurement, configure input and output, and control goscanner's performance.

Input The input module provides an interface for the scanner to know which hosts to scan. The input module can be an IP address file where addresses and optionally domain information are read from a file and fed to the scanner module. Alternatively, the input module also supports a random IP address generator which pseudo-randomly generates addresses using a linear congruential generator [1].

Scanner The scanner module takes the host information provided by the input module and executes the scan. Currently, scans for the SSH, TLS, and HTTP protocols are implemented, although other protocols can also be added easily. It also respects the configuration as specified through the configuration module. Finally, it sends the scan results to the output module.

Output The output module receives the scan results from the scanner module and saves them to output files or alternatively in a database. The stored information includes host information (IP address, domain name, host configuration), TLS certificate or SSH host key information respectively, HTTP information (includes header values; only present for TLS-based measurements), and information on linking the hosts to the respective TLS, SSH, and HTTP results.

3.2.3 Implementation

We implement goscanner in the programming language Go [110].

In Figure 3.2 we show the important interdependencies of the goscanner tool. See Appendix A for the visualization of the complete interdependencies of the goscanner project.

In the following we elaborate on the important parts of the goscanner tool. Note that we group the description of the implementation by functionality instead of strictly following the file structure. We reference single files to lay out the linking of functionality to files.

Overview

In the implementation of goscanner we follow Go's "Don't communicate by sharing memory; share memory by communicating" paradigm [109] and use goroutines to make use of CPU multiprocessing and we leverage channels to exchange data between these goroutines. Since the duration of network interactions is difficult to predict, the goroutines model allows us to adapt to this "known unknowns" situation by distributing CPU time over several tasks.

[1] LCG configuration: modulus $m = 2^{32}$, multiplier $a = 2147483655$, increment $c = 7$.

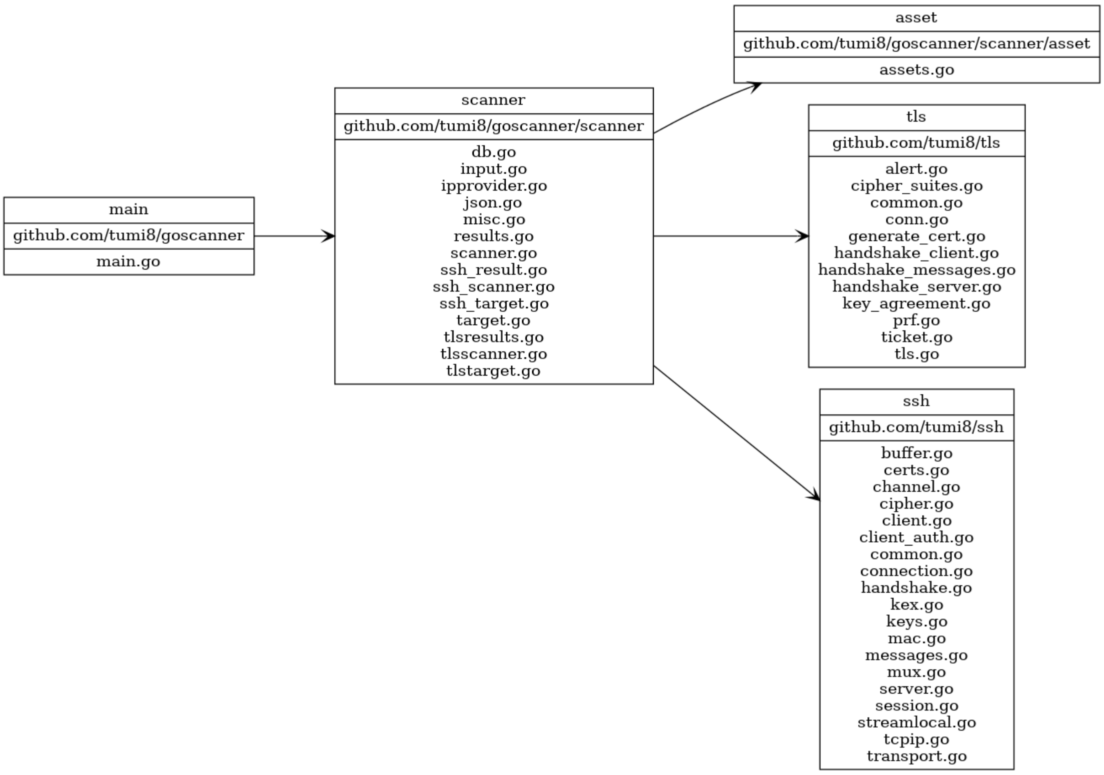


Figure 3.2: Visualization of main goscanner interdependencies, created using goviz [205].

In Figure 3.3 we show goscanner's channel-based information passing between different goroutines. This channel-based communication facilitates the one-to-man and many-to-one information passing, i. e., one target generation goroutine feeding many scanning goroutines which in turn feed information to a single output thread. The multiple scanning goroutines allow to scale CPU intensive tasks such as cryptographic calculations on multiple CPU cores and also facilitate the offset of delays due to packets traveling through the network. The number of goroutines can be customized based on the system where goscanner is run using a command line option.

We also provide a Makefile to ease the building of the goscanner tool with its dependencies and assets.

We split up the implementation into several packages and files within these packages to separate functionality. We will thoroughly explain the different packages and their functionality in the following sections.

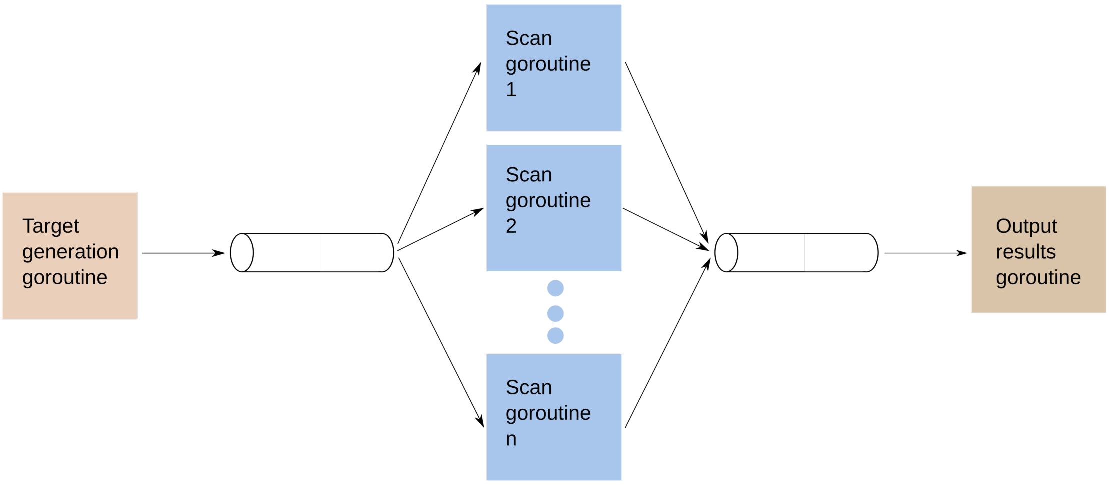


Figure 3.3: goscanner's channel-based information passing implementation.

Main

The main package contains only a single file (`main.go`). Its main purpose is to parse and validate command line arguments and setting up the goscanner. It then spawns different goroutines depending on the configuration and hands over control to the main scanner package.

The file also contains `go:generate` directives which allow to download latest asset files (see below).

Target Generation

goscanner implements two different ways of feeding targets to the scanner component: Reading targets from a file and generating targets using a pseudo-random LCG. The former allows to read targets from a text file or a json file, while the targets consist of IP addresses and optionally domains names. Domain names are important when the scanner is uses SNI for TLS scans. The latter method generates a pseudo-random IPv4 address based on an LCG and a seed value. The LCG ensures that each IP address is targeted only once.

The LCG implementation can be found in the `ipprovider.go` file, the remaining target generation functionality is stored in `input.go`. Both files are part of the `scanner` package.

SSH Probing

goscanner allows to connect to SSH hosts and download their SSH host key. The SSH host key is a cryptographic fingerprint and can therefore serve as a way to identify SSH hosts. Studies have shown that this fingerprint is not as unique as it should be [99, 123]. goscanner's SSH probing functionality can be used to efficiently perform these types of measurements.

We implement SSH functionality in the `scanner` package within the files `ssh_scanner.go`, `ssh_target.go`, and `ssh_result.go`. When establishing an SSH connection we send a special text in the client version field to make our research intentions clear [1]. Importantly, we do not try to log into systems and abort the connection after exchange of the SSH host key. To achieve this we always reject the host key presented by the server which consequently tears down the connection gracefully. In order to get additional information about the SSH connection (i. e., key-exchange information, server version, SSH host key), we modify the original Go implementation of the `crypto/ssh` package by propagating this information back to the calling SSH scanner goroutine. To this end we amend the `client.go`, `connection.go`, and `handshake.go` files. We then import and use the modified version of the `ssh` package [239] in goscanner. The resulting information from each SSH connection is sent to the goscanner's result output goroutine via a channel.

Contributions to the SSH implementation were made by Hendrik Eichner during his Bachelor's Thesis [78].

TLS and HTTPS Probing

In addition to SSH probing goscanner also allows to scan TLS servers to retrieve their TLS certificates. This allows us to evaluate the Internet's use of TLS certificates and identify security misconfigurations [16, 101].

We implement TLS functionality in the `scanner` package within the files `tlsscanner.go`, `tlstarget.go`, and `tlsresults.go`. In contrast to SSH, domain names are important when establishing a TLS connection. We therefore extract domain names from targets provided by the target generation goroutine and signal it to the TLS server using the Server Name Indication extension [76]. To learn which cipher suite is preferred by a TLS server, we list all supported cipher suites by Go's standard library. In addition, we also send the TLS Fallback SCSV [169]. This signaling cipher suite allows us to check if TLS servers implement this TLS downgrade prevention mechanism correctly. In order to avoid filtering out this cipher suite we fork the original `crypto/tls` package and modify the `handshake_client.go` file [240].

[1] `SSH-2.0-OpenSSH_Research_SSH_scanner`

In addition to a raw TLS handshake, goscanner also supports HTTPS scanning. In this mode goscanner sends an HTTP HEAD command after a successful TLS connection establishment. The HTTPS server should then send its HTTP headers but not any content. We then filter for security relevant headers [1] and add them to the results. goscanner can also be configured to conduct HTTP GET or any other HTTP-based requests on a path, by specifying the relevant command line option. When issuing an HTTP request goscanner sends a user agent header which mimics a regular browser [2]. This ensures that websites do not treat our scanning tool the same way as a regular browser.

To speed up the connection process we use TLS's session resumption feature and we use a client session cache for HTTP connections.

The resulting information about TLS and HTTPS servers is subsequently sent to goscanner's result output goroutine.

In order to keep TLS cipher suites parameters always updated, we download them from IANA's website [134] and store them as assets in the `scanner/asset` package within the `assets.go` file. This allows us goscanner to react fast to new cipher suites and it ensures the use of official naming in the result data.

Outputting Results

After a measurement is complete, it is sent to the output goroutine. By default the output goroutine writes the results to multiple files as follows:

`hosts.csv` This CSV file contains general information about scanned hosts, whether a connection was successfully established, at what time, supported cipher suites, and errors. This file is used by the SSH and the TLS component.

`host_keys.csv` This CSV file is used by the SSH component only and store the SSH host keys together with their fingerprint.

`relations.csv` This CSV file is used by the SSH component only and it contains a mapping table to connect hosts in the `hosts.csv` file with host keys in the `host_keys.csv` file. The mapping is done via the IP address and the host key's

[1] Example HTTP headers: Public-Key-Pins, Public-Key-Pins-Report-Only, Strict-Transport-Security, Expect-CT

[2] goscanner HTTP user-agent: `Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.85 Safari/537.36`

fingerprint. Splitting up hosts and host keys into separate files saves space, as each host key only has to be stored once even if found on multiple hosts.

`certs.csv` This CSV file is used by the TLS component only and stores TLS certificates in PEM format [22, 143, 144, 159] together with their fingerprint. It is the counterpart to SSH's `host_keys.csv`.

`cert_host_rel.csv` This CSV file is used by the TLS component only and it contains a mapping table to connect hosts in the `hosts.csv` file with certificates in the `certs.csv` file. The mapping is done via the IP address, domain name, and certificate's fingerprint. Splitting up hosts and certificates into separate files saves space, as each certificate only has to be stored once even if found on multiple hosts.

`scsv.csv` This CSV file is used by the TLS component only and it contains SCSV results for the scanned hosts.

`http.csv` This CSV file is used by the TLS component only and it contains HTTP results for the scanned hosts. Selected headers from every host responding to the HTTP request are stored therein.

In order to ensure uniqueness of output host keys and certificates, we store already written out host keys and certificates in a cache in RAM. If goscanner is run on a machine with low memory, this functionality can be disabled using a command line flag.

Alternatively to writing results to a CSV file, they can also be stored in a database. The result outputting is done in a single goroutine as this avoids contention over file handles. The outputting result functionality is implemented in the `main.go` file within the `main` package and the `results.go`, `ssh_result.go`, `tlsresults.go`, and `db.go` files within the `scanner` package.

3.2.4 Code Availability

We publish goscanner's source code together with setup instructions on GitHub [241]. goscanner has been used in several scientific studies [16, 101] and is used in weekly Internet-wide TLS and HTTPS measurements at TUM.

3.2.5 Similar Tools

There are similar tools to goscanner, which were developed during the inception of goscanner. One of these tools is ZGrab [260] which is part of the ZMap project.

3.3 Key Contributions of This Chapter

This chapter addressed research question RQ I which included challenges C 1 and C 2. We designed and implemented the ZMapv6 [244] and goscanner [241] tools, which are suitable to efficiently assess the security of Internet services and devices through Internet-scale measurements.

In the following we list the key contributions of this chapter:

ZMapv6 We extended the original IPv4-only ZMap core and enhanced it with IPv6 capabilities. We created the TCP SYN, UDP, UDP DNS, and ICMPv6 modules for IPv6 measurements. With the development of ZMapv6 we tackled challenge 1.

goscanner We designed and implemented the multi-processing enhanced protocol-based measurement tool goscanner. Due to its modular design and its goroutine-based implementation in Go it allows for protocol-based large-scale measurements and thus tackles challenge 2.

With overcoming these two challenges we can positively answer research question RQ I, as we are able to perform Internet-scale measurements in the IPv6 Internet using ZMapv6 and goscanner.

Chapter 4

Measurements in the IPv6 Internet

In this chapter we present evaluations of the IPv6 Internet using network measurements. The chapter builds on the IPv6 measurement methodology described in Section 3.1.

First, we present related work in the domain of IPv6 network analysis. Second, we analyze passive address sources (i. e., addresses obtained from passive traffic monitoring) for IPv6 addresses. Third, we evaluate active address sources (i. e., addresses obtained from active measurements) for IPv6 addresses. Fourth, we present our IPv6 hitlist service which provides fellow researchers with ready-to-go IPv6 addresses. Fifth, we discuss implications of the performed analyses on future IPv6 measurements. Sixth, we summarize the chapter with its contributions to this dissertation. Seventh, we conclude with a statement on the author's contributions.

4.1 Related Work

In this section we compare our IPv6 measurement to related work in this field. First, we detail prominent IPv4 measurement work and explain why different approaches are needed for IPv6. Second, we discuss related work in DNS-based approaches for IPv6 measurements. Third, we present previous work which leverages the structure of the IPv6 address space to learn new IPv6 addresses. Fourth, we discuss work in the field of IPv6 topology measurements.

4.1.1 IPv4 Measurement Approaches

In the IPv4 Internet, recent studies find hundreds of million responsive IPv4 addresses [23, 43, 199, 259]. Most of these measurement studies apply a brute-force approach of conducting 0/0 measurements, i. e., enumerating the complete IPv4 Internet [4, 71].

This is possible in IPv4, as the address space is densely populated and relatively small, when compared to IPv6. The IPv6 Internet on the other side is extremely sparsely populated, which leads to different approaches for IPv6 measurements.

4.1.2 DNS-based IPv6 Measurements

Due to its very nature, the Domain Name System has long been a valuable source for IPv6 addresses [66, 111]. Recently, Fiebig *et al.* walked through the rDNS `ip6.arpa` tree and found 2.8 M IPv6 addresses [92, 93]. Although this is an impressive number of addresses, the authors did not probe these addresses. Therefore, it remains unclear, how many of these addresses are actually responsive and run services such as HTTPS or DNS servers. Borgolte *et al.* also leveraged the DNS as a source of IPv6 addresses by NSEC-walking DNSSEC signed reverse zones [31].

4.1.3 Leveraging IPv6's Structural Properties

Although the IPv6 address space is sparsely populated, addresses assigned within it tend to follow a certain structure, as addressing plans are mostly drawn up by human operators. Related work in this areas leverages the structure of IPv6 addressing to find new addresses which can be used as targets for further measurements.

Ullrich *et al.* use rule mining to find just a few hundred IPv6 addresses [249].

Foremski *et al.* present a machine learning approach to learn new addresses, called Entropy/IP [96]. In this approach they train a model based on collected IPv6 addresses and subsequently build an addressing scheme model exposing the entropy of IPv6 address nybbles. These entropy profiles are then leveraged to generate new addresses.

Murdock *et al.* present 6Gen to find regions in the IPv6 address space [176]. For each region they generate possible neighboring addresses, which they deem likely also assigned. Murdock *et al.* also performed a basic version of aliased prefix detection (APD) by probing three random addresses of each /96 network.

Plonka and Berger harness the structure from IPv6 address plans to allow large datasets to be shared [188].

4.1.4 IPv6 Topology Measurements

Beverly *et al.* analyze the IPv6 topology with their tool yarrp [44]. This stateless tool randomizes probes which results in fewer dropped tracerouting probes due to rate limiting [27].

4.2 Passive Sources for IPv6 Addresses

This section is based on the publication "Scanning the IPv6 Internet: Towards a Comprehensive Hitlist" by Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle, which was published at TMA 2016 [104]. In Section 4.7 details on the author's contributions and differences between the thesis text in relation to the published papers are given.

To better understand the value of passive sources for IPv6 addresses, we analyze flow data containing addresses obtained from two sources: A large European Internet Exchange Point (IXP) and the Internet uplink of the Munich Scientific Network (MWN) operated by the Leibniz Supercomputing Centre.

At the IXP, we obtain flow data at a sampling rate of 1:10 000 (systematic count-based sampling on packets). At the MWN, we received flow data for all packets. In the following we analyze IPv6 addresses acquired between September 3 and September 16, 2015. We filter addresses to exclude invalids, bogons, and unroutables and then perform active measurements to each seen address to assess its responsiveness.

4.2.1 Analyzing Addresses from Passive Sources

In total we see 79.0 M unique routable addresses in the IXP source and 1.3 M from the MWN source.

In Figure 4.1 we depict the runup of unique IPv6 addresses, Autonomous Systems and prefixes over the two week period. We see that the percentage of new IP addresses is close to linear for the IXP as well as the MWN. This hints at the usage of privacy extensions which will be evaluated in Section 4.2.3. After a couple of days we already have seen addresses from more than 90 % of all observed ASes and prefixes.

In Figure 4.2 we plot the percentage of unique IPv6 addresses that are new. The dips in the figure are weekend days and a result of the reduced presence and activity of researchers and students.

Table 4.1: Top 5 port-protocol combinations for IXP and MWN (based on count of flows) over two week period (Sep 03 – Sep 16).

Rank	IXP	MWN
1	TCP/443 (34.3 %)	TCP/443 (19.8 %)
2	TCP/80 (10.6 %)	UDP/53 (12.2 %)
3	UDP/53 (1.2 %)	TCP/80 (10.7 %)
4	UDP/443 (0.7 %)	ICMPv6 (1.7 %)
5	ICMPv6 (0.4 %)	UDP/443 (1.4 %)

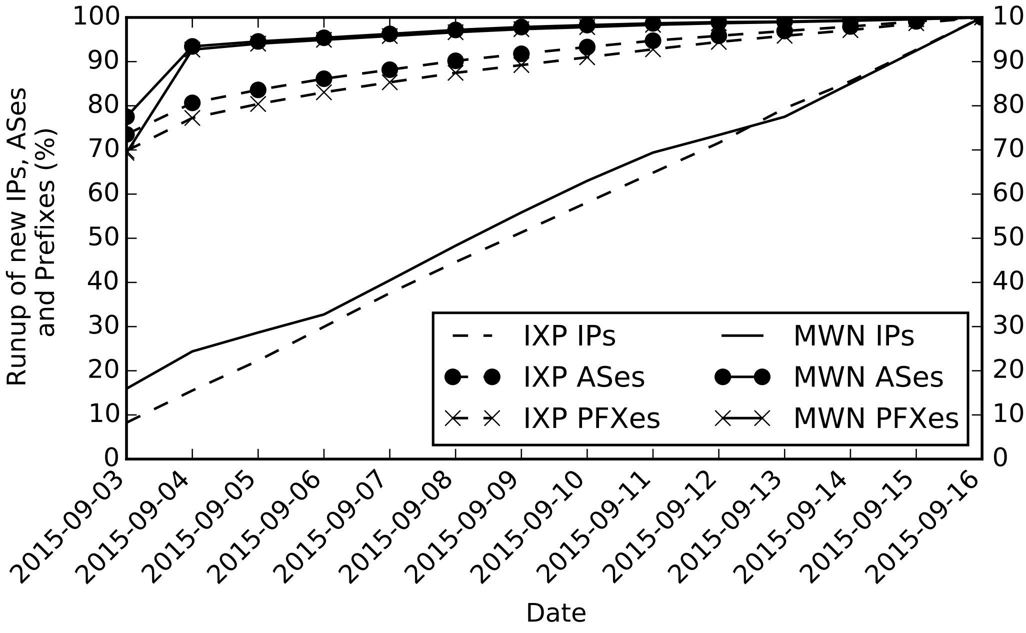


Figure 4.1: Runup of unique IPv6 addresses, ASes, and prefixes over two weeks.

When looking at a port and protocol breakdown of the observed flows, several characteristics stand out (cf. Table 4.1): First, the majority of flows stems from user-generated traffic (TCP/80, TCP/443), with TCP/443 showing clear dominance over TCP/80. Second, UDP/443, likely traffic caused by Google's QUIC protocol [206], sees a relevant share during the observation period. As a difference between the sources, a significantly higher share of DNS flows at the MWN stands out. This is to be expected, as all up- and downstream recursive DNS is processed through this link, whereas most DNS requests at an IXP are likely routed through the peers' regular Internet uplinks. We feed the results of this protocol analysis back to active measurements, by probing the top 5 protocols (cf. Section 4.3).

We next analyze AS and prefix coverage of IPv6 addresses obtained from the IXP and MWN sources. Table 4.2 compares the covered ASes and prefixes of the two passive sources. Even though we see magnitudes more addresses at the IXP, the coverage of ASes as well as prefixes is larger for the MWN source. Overall, we find that the combination of both sources covers more than 4 out of 5 ASes and provides a good prefix coverage of more than 68 %.

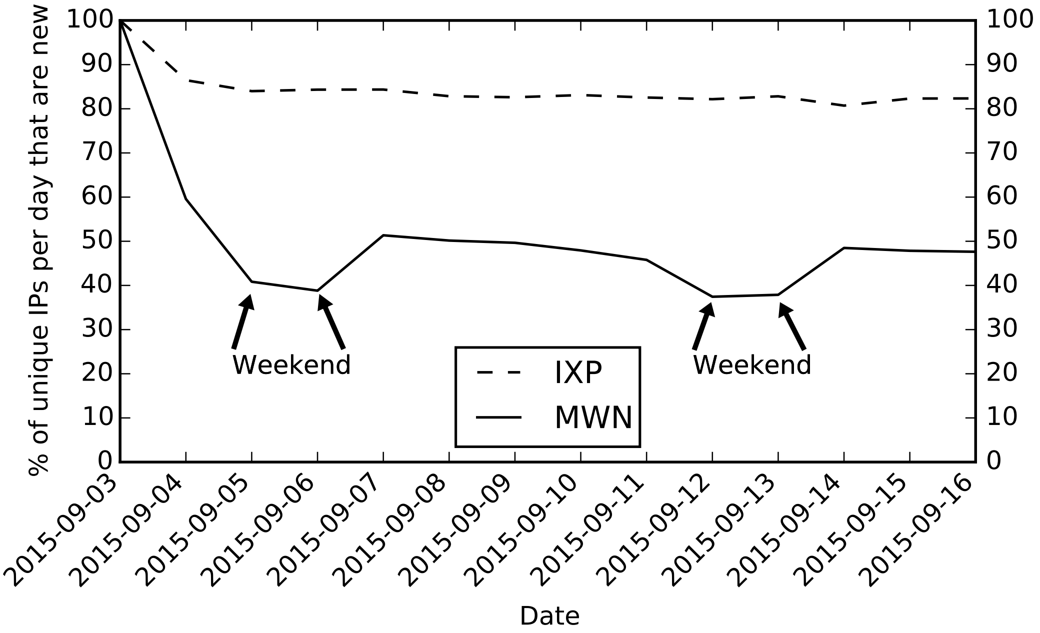


Figure 4.2: New IPv6 addresses per day.

4.2.2 Responsiveness of Addresses from Passive Sources

For both passive sources we perform active scans towards each observed IPv6 address both using ICMP and the protocol it was seen on. We conduct these measurements after 1 second, 1 minute, 1 hour, 1 day, and 1 week time intervals after the IPv6 address was initially seen.

Tables 4.3 and 4.4 show the response rates over time for IXP and MWN, respectively. The addresses learned at MWN show a more time-stable behavior. Server-type ports (TCP/80, TCP/443, UDP/443) show a very high and stable response rate. BitTorrent (UDP/49001) shows a strong decrease after an initially high response rate, while Mainline DHT (UDP/51413) exhibits a more stable response rate. Furthermore, ICMPv6 is quite stable at MWN, whereas the response rate at the IXP is much lower and decreases quickly. Possible explanations are (1) rate limiting of ICMPv6 packets at routers in target networks due to the two magnitudes higher packet volume at the IXP compared to MWN and (2) a larger presence of clients with privacy extensions in the IXP data source compared to the MWN, resulting in a decreased response rate. We explore the presence of privacy extension IPv6 addresses in the following Section 4.2.3.

Table 4.2: AS and prefix statistics for IXP and MWN.

Characteristic	IXP	MWN
Addresses	78,970,545	1,290,242
ASes	6,783	7,398
Prefixes	12,858	15,478
AS coverage	66.61%	72.65%
ASes unique to source	821	1,436
Prefix coverage	49.87%	60.04%
Prefixes unique to source	2,076	4,696
Combined AS coverage	8,219 (80.71%)	
Combined prefix coverage	25,781 (68.09%)	

Table 4.3: IXP response rates after first address observation.

Scan Type	# Targets	1 minute	1 hour	1 day	1 week
ICMPv6	66,079,853	13.28%	4.07%	1.37%	.94%
TCP/80	392,913	70.48%	66.95%	62.02%	60.09%
UDP/443	2,839	81.61%	66.07%	63.61%	55.79%
UDP/49001	25,145	57.66%	31.38%	4.83%	2.22%
UDP/51413	32,732	12.60%	9.10%	5.34%	4.25%

Table 4.4: MWN response rates after first address observation.

Scan Type	# Targets	1 minute	1 hour	1 day	1 week
ICMPv6	828,142	43.32%	40.06%	34.85%	33.43%
TCP/80	82,015	95.44%	95.47%	95.13%	94.44%
TCP/443	82,015	72.12%	72.08%	71.74%	71.03%
UDP/443	5,292	60.67%	60.99%	60.65%	59.73%
UDP/49001	7,314	51.24%	36.88%	10.63%	7.47%
UDP/51413	12,875	42.62%	39.10%	32.94%	29.91%

Table 4.5: Top 5 vendors for EUI-64 IPv6 addresses.

	IXP		Scamper	
Position	Vendor	Percentage	Vendor	Percentage
1	Samsung	30.7%	Arcadyan	28.4%
2	Apple	11.6%	Huawei	24.4%
3	Sony	5.8%	AVM	16.0%
4	Murata	5.1%	Sercomm	10.5%
5	Huawei	5.1%	Cisco	4.4%

4.2.3 Interface Identifiers in Passive IPv6 Address Sources

To better understand the acquired addresses we analyze the interface identifier (IID) of these addresses, i.e. the last 64 bit. The IID can reveal information about the type of device or the longevity of the address. In our analysis we compare the IXP and Scamper traceroute sources, to more clearly show the difference between passive and active sources.

Modified EUI-64

IPv6 has a mechanism to automatically assign addresses without a DHCPv6 server. This *Stateless Address Autoconfiguration* mechanism usually takes an interface's MAC address and modifies it by inserting `ff:fe` in the middle and flipping the 6th bit [126]. This modified EUI-64 ID is then appended to the announced prefix.

We analyze the number of EUI-64 addresses by extracting the IID and filtering for potential EUI-64 IDs. We find 2.3 M and 103.5 k EUI-64 addresses in the IXP and MWN source, respectively.

We aggregate the EUI-64 IDs by MAC vendors [136]. Additionally, we merge vendors with similar names. Table 4.5 shows the top 5 vendors for the IXP and Scamper sources, which are representative for passive and active sources, respectively. Not surprisingly, the IXP's EUI-64 addresses are mainly from end user devices, while the top 5 Scamper EUI-64 vendors are networking equipment manufacturers.

Privacy Extensions

To avoid unique traceability through MAC addresses encoded within the IID, IPv6 introduces a mechanism called *Privacy Extensions* [179]. These reduce traceability by randomizing the IID. Since the 6th bit (leftmost bit is 0th bit) of the IID is always set to 0 to indicate local scope, 63 uniformly distributed bits remain for the IID. By applying the *central limit theorem*, the sum of these single bit distributions approximates the normal distribution $\mathcal{N}(31.5, 15.75)$. Therefore, we count the number of bits set to one in each IID and plot this distribution against the normal distribution.

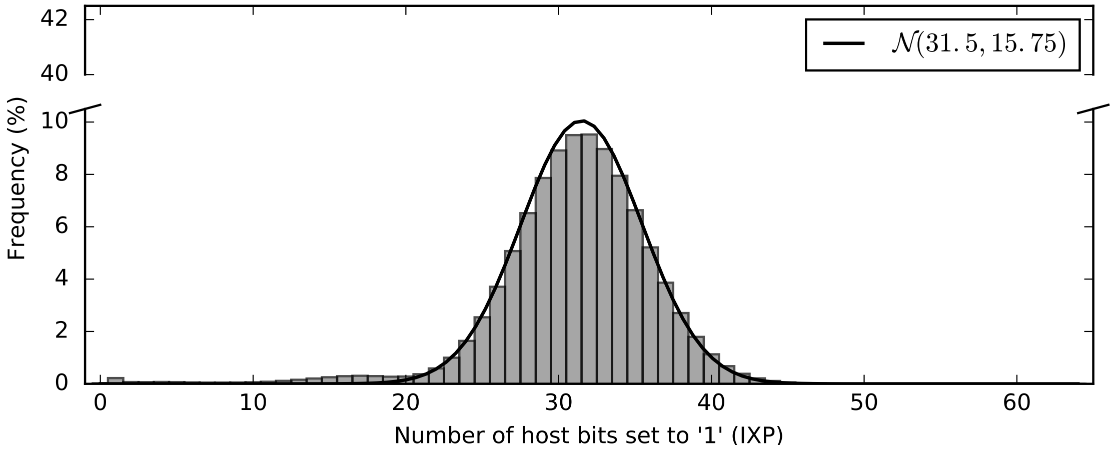


Figure 4.3: IXP Hamming weight distribution of interface ID.

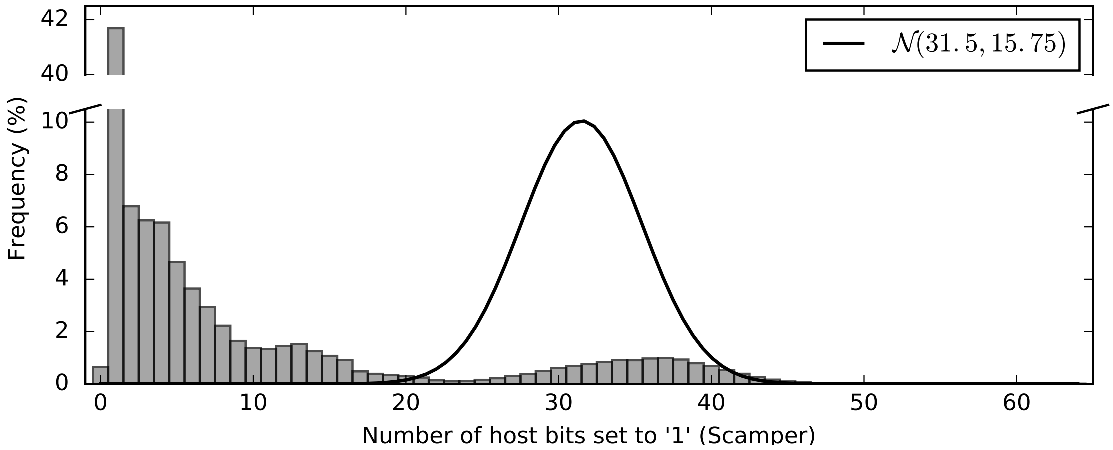


Figure 4.4: Scamper Hamming weight distribution of interface ID.

Figures 4.3 and 4.4 show the Hamming weight distribution for the IXP and Scamper sources. The Hamming weight distribution at the IXP approximates the aforementioned normal distribution. This clearly indicates that the vast majority of the IXP's addresses contains a random IID. A similar phenomenon was observed for the MWN source. Addresses obtained from the active Scamper source on the other hand differ drastically: Two thirds of IIDs have less than six bits set to one, with more than 40% only having one bit set. This hints at a large number of statically assigned addresses which seems reasonable for a source primarily consisting of routers.

In conclusion, the interface identifier analysis shows that passive sources consist mostly of client addresses, with a very high percentage of privacy extension addresses. This explains the stark decrease in responsiveness for these passive sources.

4.3 Active Sources for IPv6 Addresses

This section is based on the publications "Scanning the IPv6 Internet: Towards a Comprehensive Hitlist" by Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle, which was published at TMA 2016 [104]; "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists" by Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczyński, Stephen D. Strowes, Luuk Hendriks, and Georg Carle, which was published at IMC 2018 [100]. In Section 4.7 details on the author's contributions and differences between the thesis text in relation to the published papers are given.

We now analyze IPv6 addresses for hitlists obtained from active sources. We leverage a variety of sources, for which we provide an overview in Table 4.6. Our guiding principle in selecting sources was that they should be *public*, i. e., accessible to anyone for free, in order to make our work reproducible and to allow fellow researchers to deploy variations of our IPv6 hitlist. We consider data with an open and usually positive access decision process as public, such as Verisign's process to access zone files. We also aim to have balanced sources, which include servers, routers, and a share of clients. The sources we leverage are as follows:

Domain Lists This source contains a total of 212 M domains from various large DNS zones, resolved for AAAA records on a daily basis, yielding about 9.8 M unique IPv6 addresses. This source also includes domains extracted from blacklists provided by Spamhaus [229], APWG [18], and Phishtank [187], which leverage 8.5 M, 376 k, and 170 k domains, respectively.

FDNS A comprehensive set of forward DNS (FDNS) ANY lookups performed by Rapid7 [197], yielding 2.5 M unique addresses.

CT DNS domains extracted from TLS certificates logged in Certificate Transparency (CT), and not already part of domain lists, which yields another 16.2 M addresses.

AXFR and TLDR IPv6 addresses obtained from DNS zone transfers (AXFR) from the TLDR project [166] and our own AXFR transfers. Obtained domain names are also resolved for AAAA records daily. This source yields 0.5 M unique IPv6 addresses.

Bitnodes To gather client IPv6 addresses, we use the Bitnodes API [257], that provides current peers of the Bitcoin network. Although this is the smallest source,

Table 4.6: Overview of hitlist sources, as of May 11, 2018.

Name	Public	Nature	IPs	new IPs	#ASes	#PFXes	Top AS1	Top AS2	Top AS3
DL: Domain Lists[1]	Yes	Servers	9.8 M	9.8 M	6.1 k	10.3 k	89.7 %★	2.0 %●	1.5 %■
FDNS: Rapid7 FDNS	Yes	Servers	3.3 M	2.5 M	7.7 k	13.6 k	16.7 %★	8.9 %▲	6.7 %✣
CT: Domains from CT logs[2]	Yes	Servers	18.5 M	16.2 M	5.3 k	8.7 k	92.3 %★	1.6 %✣	0.8 %★
AXFR: AXFR&TLDR	Yes	Mixed	0.7 M	0.5 M	3.2 k	4.7 k	57.0 %★	14.0 %●	8.3 %■
BIT: Bitnodes	Yes	Mixed	31 k	27 k	695	1.4 k	8.0 %★	6.0 %■	6.0 %▲
RA: RIPE Atlas[3]	Yes	Routers	0.2 M	0.2 M	8.4 k	19.1 k	6.6 %✣	3.5 %★	3.1 %✣
Scamper	–	Routers	26.0 M	25.9 M	6.3 k	9.8 k	38.9 %★	23.8 %●	12.0 %■
Total			58.5 M	55.1 M	10.9 k	25.5 k	45.4 %★	18.4 %★	11.5 %●

1: Zone Files, Toplists, Blacklists (partially with NDA); 2: Excluding DNS names already included in Domain Lists; 3: Traceroute and ipmap data

★Amazon, ●Host Europe, ■Cloudflare, ▲Linode, ✣DTAG, ★ProXad, ●Hetzner, ■Comcast, ▲Swisscom, ✣Google, ★Antel, ●Versatel, ■BIHNET

contributing 27 k unique IPv6 addresses, we still find it valuable as it also adds client addresses.

RIPE Atlas We extract all IPv6 addresses found in RIPE Atlas traceroutes, as well as all IPv6 addresses from RIPE's ipmap project [201], which adds another 0.2 M addresses. These are highly disjoint from previous sources, likely due to their nature as routers.

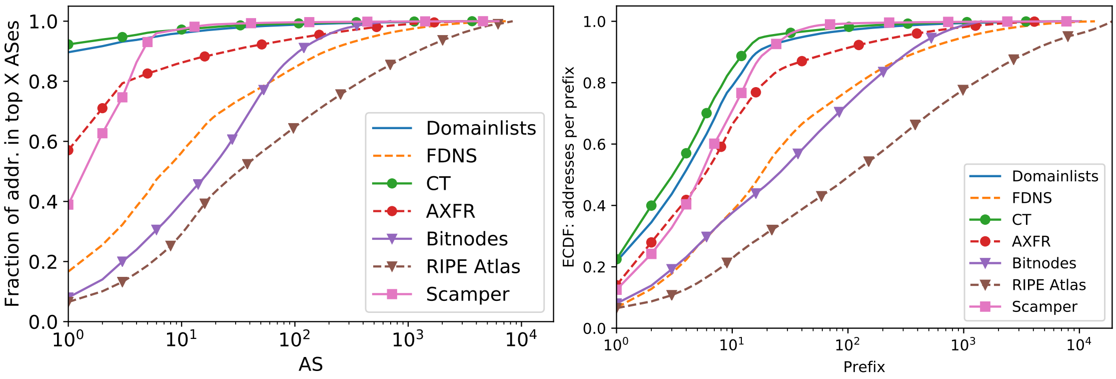
Scamper Finally, we run traceroute measurements using scamper [161] on all addresses from other sources, and extract router IP addresses learned from these measurements. This source shows a very strong growth characteristic, with 25.9 M unique IP addresses.

4.3.1 Input Distribution and Evolution

We also evaluate the distribution of input sources based on announced BGP prefixes and Autonomous Systems, as well as looking at the evolution of inputs over time.

When evaluating the AS distribution for each source in Figure 4.5a, we see stark differences, e. g. for domainlists and CT only a handful of ASes make up a large fraction of addresses, compared to the more balanced RIPE Atlas source. To counter these unbalanced sources we develop Aliased Prefix Detection, which is able to detect dummy addresses, and present it in Section 4.3.3.

Additionally, we analyze the distribution of announced BGP prefixes per hitlist source as shown in Figure 4.5b. Although we see similarities in the prefix distribution when comparing it to the AS distribution, all sources are more balanced when evaluating prefixes. Most of the highly unbalanced ASes in sources in the AS distribution contribute addresses from larger number of prefixes, resulting in a more balanced prefix distribution.



(a) AS distribution for hitlist sources.

(b) Prefix distribution for hitlist sources.

Figure 4.5: Distribution of input sources for IPv6 hitlist.

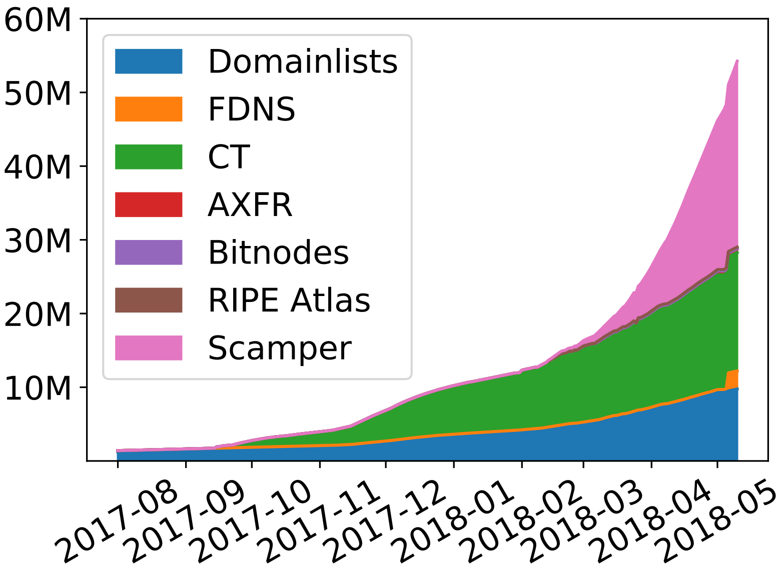


Figure 4.6: Cumulative runup of IPv6 addresses per hitlist source.

Table 4.7: Contribution of active sources from 2016 dataset [104].

Name	IP addresses	ASes	Prefixes
Alexa Top 1M	43.8 k	1.4 k	1.7 k
rDNS	462.2 k	6.7 k	4.8 k
FDNS	1.4 M	8.5 k	5.7 k
Zone files	424.7 k	3.0 k	2.4 k
Total	2.7 M	7.3 k	12.9 k

Next, we evaluate the evolution of input sources by analyzing the contribution of new IPv6 addresses to the hitlist for each source. In Figure 4.6 we depict the runup of hitlist addresses per source over a period of ten months. We see that most hitlist addresses stem from three sources: Domain lists, domains extracted from Certificate Transparency, and router addresses obtained from running traceroutes with scamper. As we find scamper's explosive growth peculiar, we conduct a closer investigation, which reveals that 90.7 % of those IPv6 addresses are SLAAC addresses, i. e., marked by `ff:fe`. The vendor codes in MAC addresses gained from those routers indicate that they are typically home routers: 47.9 % ZTE and 47.7 % AVM (Fritzbox), followed by 1.2 % Huawei with a long tail of 240 other vendors. This shows that our source includes mainly home routers and CPE equipment. Depending on the type of study, it may be desirable to include or exclude these CPE devices. Related work by Beverly *et al.* also discovers these CPE router addresses [27].

We also compare our results from 2018 [100] to our previous work from 2016 [104]. Table 4.7 shows the contribution of the sources Alexa Top 1M, rDNS, FDNS, and Zone files from the 2016 dataset. When comparing the aggregate of these active sources to the aggregate of the 2018 active sources, the increase in IPv6 addresses becomes quite apparent. We find that the number of IPv6 addresses from public sources has increases by a factor of more than 20. The rise of covered ASes and prefixes is less pronounced: The number of ASes with hitlist addresses increased by about 50 %, whereas the number of covered prefixes doubled during the two year period.

4.3.2 rDNS as a Data Source

We also investigate the usefulness of IPv6 rDNS entries for active measurements. As shown by previous work, rDNS walking can be a source for IPv6 addresses [92, 93].

While IPv6 rDNS addresses were used to, e. g. find misconfigured IPv6 networks [31], we are not aware of studies evaluating overall responsiveness. Since walking the rDNS tree to harvest IPv6 addresses is a large effort and puts strain on important Internet

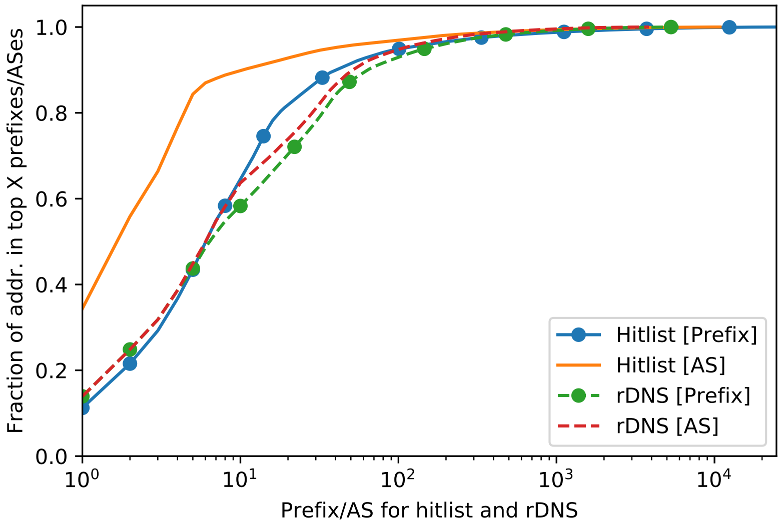


Figure 4.7: Prefix and AS distribution comparing rDNS to other hitlist sources.

infrastructure, we classify this source as "semi-public", compared with sources such as the Alexa Top 1 M list, which is available for download.

We use IPv6 rDNS data provided by Fiebig *et al.* [92] to perform active measurements and compare the results against other hitlist sources. Analyzing the overlap and structure of IPv6 addresses obtained from rDNS, we find a very small intersection with our hitlist. Of the 11.7 M addresses from rDNS, 11.1 M are new. The prefix distribution of rDNS and hitlist addresses is quite similar, as shown in Figure 4.7.

The AS distribution is even more balanced for rDNS addresses compared to the hitlist. Therefore, the addition of rDNS data to the hitlist input would not introduce a bias at the prefix or AS level.

Next, we perform active measurements to compare the response rate of the rDNS population to the hitlist population. Before the active measurement, we filter 2.1 M unrouted addresses and 13.1 k addresses residing in aliased prefixes (see Section 4.3.3) from the rDNS addresses. The response rate for the hitlist with only non-aliased prefixes is generally similar to the response rate of rDNS. The rDNS ICMPv6 response rate is higher: 10 % compared to the hitlist's 6 %. On the other hand, we receive slightly fewer HTTP(S) responses for rDNS, at 2 % (1 %) against the hitlist's 3 % (2 %).

To ensure that responding rDNS addresses are not mostly client addresses, we first analyze the top ASes. As can be seen in Table 4.8, the top responsive ASes in the rDNS data are hosting and service providers, i. e., mostly servers (especially in the TCP/80 measurement).

Table 4.8: Top 5 rDNS ASes in input data, responsive to ICMPv6 probes, and responsive to TCP/80 probes.

#	Input		ICMPv6		TCP/80	
1	Comcast	12.5 %	Online S.A.S.	19.6 %	Google	12.8 %
2	AWeber	10.2 %	Sunokman	17.8 %	Hetzner	10.1 %
3	Yandex	9.8 %	Latnet Serviss	8.7 %	Freebit	6.8 %
4	Belpak	6.2 %	Yandex	7.9 %	Sakura	6.5 %
5	Sunokman	6.1 %	Salesforce	5.3 %	TransIP	5.0 %

Next, we look for IPv6 SLAAC's distinct `ff:fe` sequence and evaluate the hamming weight of IIDs for responsive rDNS addresses, as an additional indicator for clients. We find between 6 % and 9 % SLAAC addresses with `ff:fe`. The IID hamming weight (i. e., number of bits set to 1) can be used to infer the presence of clients with privacy extensions enabled [104]. The rDNS IID hamming weight does not suggest that the rDNS set contains a large client population, especially for TCP/80, where 60 % of addresses have a hamming weight of six or smaller.

To conclude, the responsive part of the rDNS data source adds a balanced set of IPv6 addresses. We therefore suggest adding rDNS data as input to the IPv6 hitlist. Due to technical reasons, the following subsections are evaluated on all input sources except rDNS data.

4.3.3 Aliased Prefixes

Aliased network prefixes, i. e., prefixes under which each possible IP address replies to queries, were already found when conducting IPv4 measurements [14]. For IPv6 measurements, however, aliased prefixes pose a much more significant challenge as they can easily contribute vast numbers of addresses that map to the same server, e. g. through the `IP_FREEBIND` option in Linux. This feature is already in use by CDNs [163], and was identified as a challenge in previous works for rDNS walking [93] and active measurements [176]. Aliased prefixes can artificially inflate the number of IP addresses within a hitlist(e. g. enumerating a /96 prefix can add 2^{32} addresses), and introduce significant bias into any studies using these hitlists. Given this, we want to populate our hitlistonly with valuable addresses, i. e., addresses belonging to different hosts and having balanced prefix and AS distributions. This requires reliable detection and removal of aliased prefixes, for which we introduce a rigorous method in the following.

Similar to previous work [93, 176], our method has its roots in the concept that a randomly selected IP address in the vast IPv6 space is unlikely to respond. Thus, when probing randomly selected addresses, a prefix can be classified as aliased after a certain

Table 4.9: Example of IPv6 fan-out for multi-level aliased prefix detection. We generate one pseudo-random address in `2001:0db8:407:8000::/64` for each of the 16 subprefixes, i. e., `2001:0db8:407:8000:[0-f]/68`.

2001:0db8:0407:8000::/64
2001:0db8:0407:8000:0151:2900:77e9:03a8
2001:0db8:0407:8000:181c:4fcb:8ca8:7c64
2001:0db8:0407:8000:23d1:5e8e:3453:8268
⋮
2001:0db8:0407:8000:f693:2443:915e:1d2e

number of replies have been received. Murdock *et al.* [176] send three probes each to three random addresses in every /96 prefix. Upon receipt of replies from all three random addresses, the prefix is determined as aliased. In the following, we describe how we improve efficiency and effectiveness of this approach in several ways.

Alias detection needs to fulfill two criteria to scale: (1) detection must be low-bandwidth, with a small number of packets required per network, (2) detection must function for end hosts, not only routers, which excludes many alias detection techniques.

Multi-Level Aliased Prefix Detection

For our daily scans, we perform multi-level aliased prefix detection (APD), i. e., detection at different prefix lengths. This is in contrast to previous works that use static prefix lengths, e. g. /96.

To determine whether a prefix is aliased, we send 16 packets to pseudo-random addresses within the prefix, using TCP/80 and ICMPv6. For each packet we enforce traversal of a subprefix with a different nybble. As an example, to check if `2001:db8:407:8000::/64` is aliased, we generate one pseudo-random address for each four-bit subprefix, `2001:db8:407:8000:[0-f]000::/68`. See Table 4.9 for a visual explanation. Using this technique we ensure that (1) probes are distributed evenly over more specific subprefixes and (2) pseudo-random IP addresses, which are unlikely to respond, are targeted.

For each probed prefix we count the number of responsive addresses. If we obtain responses from all 16 probed addresses, we label the prefix as aliased.

We run the aliased prefix detection on IPv6 addresses that are either BGP-announced or in our hitlist. The former source allows us to understand the aliased prefix phenomenon on a global scale, even for prefixes where we do not have any targets. The latter source allows us to inspect our target prefixes more in-depth.

For BGP-based probing, we use each prefix as announced, without enumerating additional prefixes. For our hitlist, we map the contained addresses to all prefixes from 64 to 124, in 4-bit steps. We limit APD probing to prefixes with more than 100 targets for two reasons: First, efficiency, as APD probing requires 32 probes (16 for ICMPv6 and TCP/80, respectively). Second, impact, as prefixes with less than 100 probes can only distort our hitlist in a minor way. We exempt /64 prefixes from this limitation so as to allow full analysis of all known /64 prefixes. We use 47.4 M probes to guarantee complete coverage of all /64 prefixes and 49.2 M probes in total.

As we perform target-based APD at several prefix lengths, the following four cases may occur:

1. Both more and less specific are aliased
2. Both more and less specific are non-aliased
3. More specific aliased, less specific non-aliased
4. More specific non-aliased, less specific aliased

The first two cases depict the “regular” aliased and non-aliased behaviors, respectively. The third case is more interesting as we observe divergent results based on the prefix length that we query. One example is a /96 prefix which is determined as being non-aliased, with only 9 out of 16 /100 subprefixes determined as aliased. This case underlines the need for our fan-out pseudo-random aliased prefix detection. Using purely random addresses, all 16 could by chance fall into the 9 aliased subprefixes, which would then lead to incorrectly labeling the entire /96 prefix as aliased. The fourth case is an anomaly, since an aliased prefix should not have more specific non-aliased subprefixes. One reason for this anomaly is packet loss for subprefix probes, incorrectly labeling the subprefix as non-aliased.

We analyze how common the fourth case is in our results and investigate the reasons. On May 4, 2018, we detect only eight such cases at the prefix lengths /80, /116, and /120:

The /80 prefix shows 3 to 5 out of the 16 possible responses over time. The branches of responding probes differ between days, with no discernible pattern. We suspect this prefix is behind a SYN proxy [77], which is activated only after a certain threshold of connection attempts is reached. Once active, the SYN proxy responds to every incoming TCP SYN, no matter the destination.

The /116 prefix consistently shows 15 out of 16 probes being answered on consecutive days, even though a less specific prefix was classified as aliased. Moreover, the 15 probes answer with the same TCP options on consecutive days. The non-responding probe is always on the `0x0` branch, so we believe the subprefix is handled differently and not by an aliased system. In fact, comparing the paths of the different branches reveals that the `0x0` branch is answered by an address in a different prefix. The DNS reverse pointer of this address hints at a peering router at DE-CIX in Frankfurt, Germany. This /116 anomaly underlines the importance of the multi-level aliased prefix detection, since there are in fact small non-aliased subprefixes within aliased less specific prefixes.

The case of six neighboring /120 prefixes manifests less consistently than the previously described phenomenon. The branches that lack responses change from day to day, as well as from prefix to prefix. Subsequent manual measurements show that previously unresponsive branches become responsive. The root cause is most likely ICMP rate limiting, which explains the seemingly random responding branches. We try to counter packet and ICMPv6-rate limiting loss as explained in the following subsection.

After the APD probing, we perform longest-prefix matching to determine whether a specific IPv6 address falls into an aliased prefix or not. This ensures we use the result of the most closely covering prefix for each IPv6 address, which creates an accurate filter for aliased prefixes. If a target IP address falls into an aliased prefix, we remove it from that day's ZMapv6 and Scamper scans.

Loss Resilience

Packet loss might cause a false negative, i. e., an aliased prefix being incorrectly labeled as non-aliased. To increase resilience against packet loss, we apply (1) cross-protocol response merging and (2) a multi-day sliding window.

As we are probing all 16 target addresses on ICMPv6 and TCP/80, IP addresses may respond inconsistently. Our technique hinges on the fact that it is unlikely for a randomly chosen IP address to respond at all, so we treat an address as responsive even if it replies to only the ICMPv6 or the TCP/80 probe. While this greatly stabilizes our results, we still see high-loss networks, which would fail automatic detection, but could be manually confirmed as aliased.

To further tackle these, we introduce a sliding window over several past days, and require each IP address to have responded to any protocol in the past days. As prefixes may change their nature, we perform this step very carefully, and aim for a very short sliding window to react to such changes as quickly as possible.

To find an optimum, we compare the number of days in the sliding window to the number of prefixes that are unstable, i. e., change the nature of stable and unstable over several days. We show the data in Table 4.10, which confirms that with a sliding

Table 4.10: Impact of sliding window duration on number of unstable prefixes.

Sliding window	0	1	2	3	4	5
Unstable prefixes	65	26	22	14	14	13

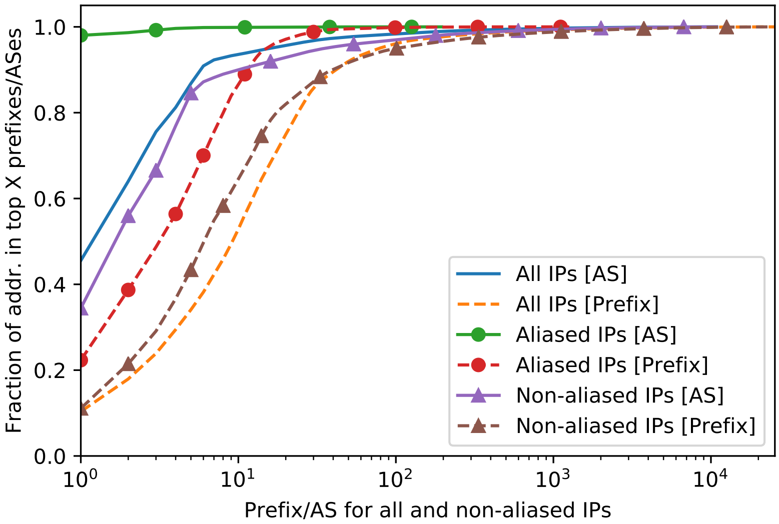


Figure 4.8: Prefix and AS distribution for aliased, non-aliased, and all hitlist addresses.

window of just 3 days, we can reduce the number of unstable prefixes by almost 80 %, while only adding a small delay for prefixes that change their nature. With the final sliding window of 3 days, only 14 of the 909 aliased prefixes as of May 1, 2018 show an unstable nature.

Impact of De-Aliasing

We apply the aliased prefix detection (APD) filtering daily and analyze the impact on our hitlistfor May 11, 2018. Before filtering, there are a total of 55.1 M IPv6 addresses on our hitlist. After identifying aliased prefixes, 29.4 M targets (53.4 %) remain. With the unfiltered hitlist we cover 10 866 ASes and 25 465 announced prefixes. Removing aliased prefixes reduces our AS coverage by only 13 ASes, and lowers prefix coverage by 3.2 % to 24 648 prefixes.

We show the AS and prefix distribution for aliased, non-aliased, and all IPv6 addresses in Figure 4.8. By comparing AS distributions we find aliased prefixes as heavily centered on a single AS (Amazon). In consequence, the AS distribution for non-aliased prefixes is flatter than the population as a whole. The picture changes for prefix distributions: targets in non-aliased prefixes are now slightly more top-heavy compared to the general population. One of the reasons is that the vast majority of aliased IP addresses are

Table 4.11: Results for validating aliased and non-aliased prefixes.

Scan type	Inconsistent	Consistent	Indecisive
Non-aliased prefixes	50.4 %	23.8 %	25.8 %
Aliased prefixes	5.1 %	63.8 %	31.1 %

within 189 /48 prefixes announced by Amazon, which are the shortest detected aliased prefixes. This results in shifting down the prefix distribution of the general population.

We find that aliasing barely occurs in the shortest prefixes. We find some groups of aliased /32 prefixes, but the majority of aliased IPv6 addresses belong to /48 prefixes, with the majority being announced by Amazon and Incapsula. We also see that filtering these prefixes is effective: the Amazon aliased prefixes comprise a large share of the input set.

Validating Aliased Prefixes

We validate whether our detected aliased prefixes in fact belong to single hosts each responding for a complete prefix. Since these each address belonging to one host should exhibit similar behavior, we employ fingerprinting techniques to validate our aliased prefixes.

We select a subset of replies from 20 692 /64 prefixes classified as aliased and compare the results to non-aliased prefixes.

Table 4.11 shows the results of this analysis. We find that aliased prefixes exhibit very consistent behavior compared to non-aliased prefixes.

Comparison to Murdock *et al.* 's Approach

In order to assess our APD approach we quantitatively compare it to Murdock *et al.* 's [176]. As Murdock *et al.* perform alias detection on a best-effort basis by probing addresses in prefixes with a static prefix length of /96, we expect our approach to find more aliased prefixes. This is in fact the case as we find 992.6 k hitlistaddresses residing in aliased prefixes which are not detected by Murdock *et al.* On the other hand, Murdock *et al.* 's approach classifies only 1.4 k hitlistaddresses as aliased which we deem non-aliased.

Additionally, we compare the bandwidth requirements of our APD approach to Murdock *et al.* 's. As our approach works on multiple prefix levels where at least 100 targets are present, we focus on the most likely aliased prefixes. Consequently, we send probes to a total of 50.1 M IPv6 addresses to determine aliased prefixes in our hitlist. Using

Murdock *et al.* 's static /96 prefixes, more than twice as many IPv6 addresses (113.8 M) are probed.

To summarize, our approach finds 992.6 k more hitlist addresses in aliased prefixes compared to Murdock *et al.* 's approach and at the same time probes less than half the number of IP addresses.

4.3.4 Address Probing

To assess how our hitlist addresses perform in terms of responsiveness, we run measurements on multiple ports and over multiple days.

We generate IPv6 targets and probe these targets' responsiveness each day. First, we collect addresses from our hitlist sources. Second, we preprocess, merge, and shuffle these addresses in order to prepare them as input for scanning. Third, we perform aliased prefix detection to eliminate targets in aliased prefixes. Fourth, we traceroute all known addresses using Scamper [161] to learn additional router addresses. Fifth, we use ZMapv6 [244] to conduct responsiveness measurements on all targets. We send probes on ICMP, TCP/80, TCP/443, UDP/53, and UDP/443 to cover the most common services [104]. We repeat this process each day to allow for longitudinal responsiveness analysis.

Responsive Addresses

We first evaluate responsive addresses based on their corresponding BGP prefix.

Overall, our hitlist contains 1.9 M responsive IPv6 addresses, spread over 21 647 BGP prefixes covering 9968 different ASes.

We find that most announced BGP prefixes are covered with dozens to hundreds of responsive targets, whereas a few prefixes contribute 12 k or more responsive addresses.

Cross-protocol Responsiveness

We analyze the cross-protocol responsiveness of our probes, to understand what kind of IPv6 hosts are responding to our probes. In Figure 4.9 we show the conditional probability of responsiveness between protocols, i. e., if protocol X is responding, how likely is it that protocol Y will respond. We compare our findings for IPv6 to Bano *et al.* [23] who performed a similar analysis for IPv4.

We find that if an IPv6 address responds to any of the probes, there is at least a 89 % chance of the same IP address also responding to ICMPv6. The ICMP correlation in IPv6 is higher compared to IPv4, where we see values as low as 73 % [23]. Since ICMPv6 is an integral part of IPv6, it should not be simply blocked in firewalls [61], which makes

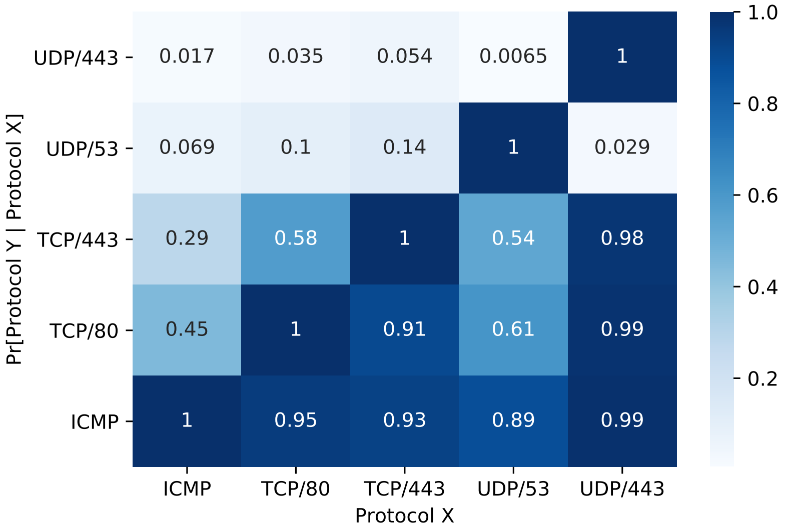


Figure 4.9: Conditional probability of responsiveness between services.

it more likely that hosts are responding to ICMPv6 compared to its IPv4 counterpart. One reason for this could be the important role that ICMPv6 plays in IPv6 connectivity, e. g. providing neighbor discovery using NDP [180].

Additionally, we see correlations between UDP/443, TCP/443, and TCP/80. More specifically, if an address is responsive to QUIC (UDP/443), it has a likelihood of 98 % to be also providing HTTPS and HTTP services. HTTPS servers are 91 % likely to provide an HTTP service as well, e. g. to offer a forwarding service to the secure version of a web page. Note that the reverse correlation (HTTP → HTTPS → QUIC) is far less pronounced. This is to be expected, as not all HTTP(S) servers also offer their websites via QUIC. Compared to the HTTPS → HTTP correlation of 91 % in IPv6, we see only a 72 % correlation in IPv4 [23].

Analyzing DNS (UDP/53) correlation shows mostly similar results in IPv6 as in IPv4. One exception is the lower correlation to HTTPS in IPv6 (54 %) compared to IPv4 (78 %) [23].

4.3.5 Longitudinal Responsiveness

To analyze address responsiveness over time, we probe an address continuously even if it disappears from our hitlist's daily input sources. We evaluate longitudinal responsiveness over two weeks as depicted in Figure 4.10. As a baseline for each source we take all responsive addresses on the first day.

We find that IPv6 addresses from domain lists (DL), FDNS, and RIPE Atlas answer quite consistently over the 14 day period, with all three sources losing only a few per-

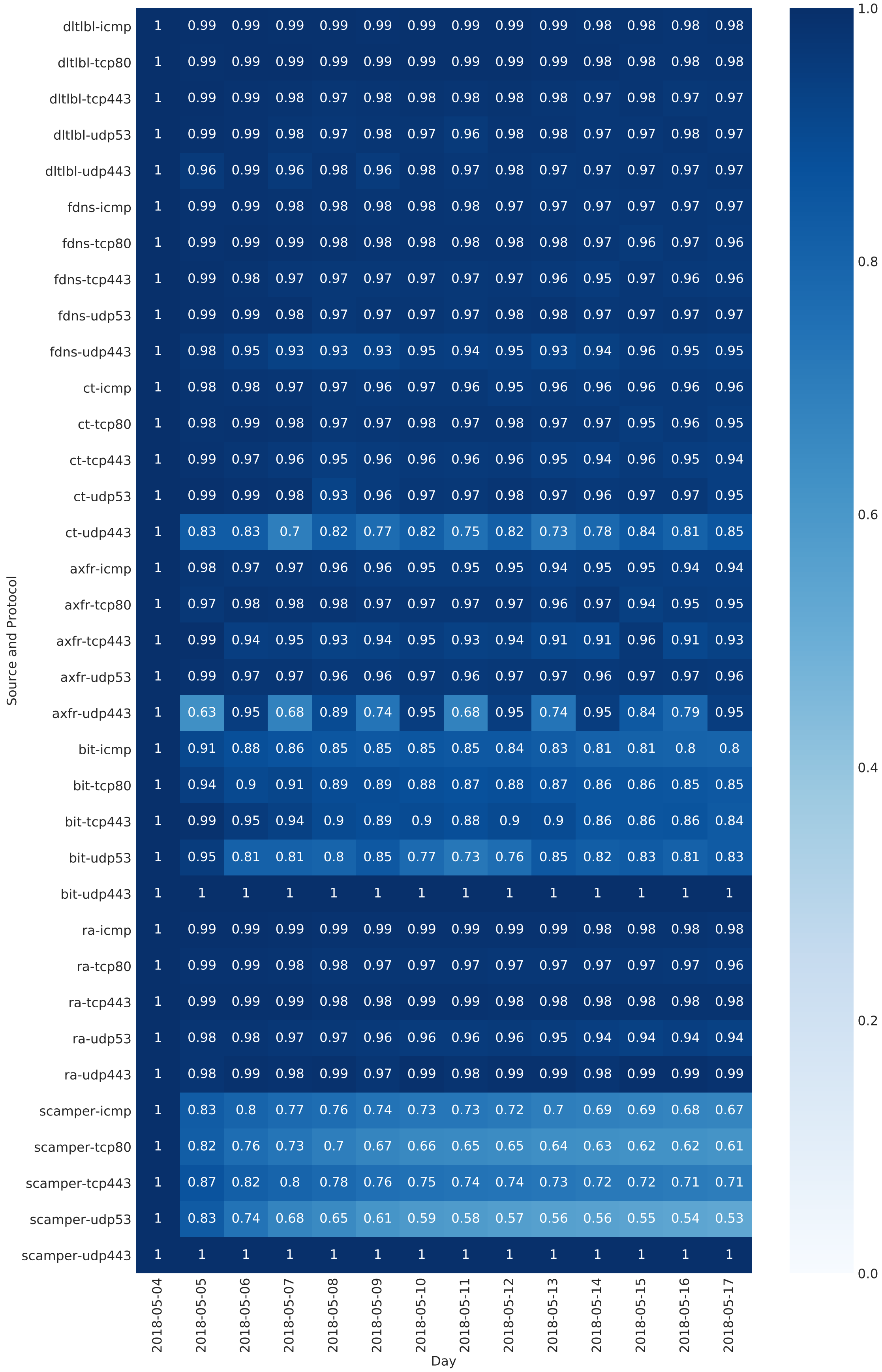


Figure 4.10: Responsiveness over time, split up by hitlist source and probed protocol.

centages of addresses. CT and AXFR sources overall reach a similarly stable response rate; their QUIC response rates, however, fluctuate more heavily. We investigate this phenomenon and find that more than 80 % of fluctuating addresses are located in two prefixes: Akamai and HDNet. We suspect that these companies are testing the deployment of QUIC on some of their systems or that our measurements are caught by a rate limiting mechanism, resulting in flaky response behavior. Moreover, sources which include clients or CPE devices such as Bitnodes and Scamper lose 20 % and 32 % of the responding hosts, respectively. The responsive QUIC addresses for Bitnodes and Scamper sources remain seemingly stable, which is due to the relatively small number of addresses from these sources.

4.3.6 Address Learning

In addition to acquiring IPv6 addresses through domain names and other sources, we can also detect addressing schemes, and leverage those patterns to learn previously unknown addresses.

Methodology

To generate previously unknown addresses, we feed our hitlist into a re-implementation of Entropy/IP [7, 95, 96], and a pre-release version of 6Gen [176]. For this work, we improve the address generator of Entropy/IP by walking the Bayesian network model exhaustively instead of randomly. The improved generator lets us focus on more probable IPv6 addresses, under a constrained scanning budget.

First, we use all addresses in non-aliased prefixes to build a seed address list. Excluding aliased prefixes avoids generating addresses in prefixes where all addresses are responsive, and thus artificially distorting the response rate. Second, we split the seed address list based on ASes, as we assume similar addressing patterns within the same AS. We limit the eligible ASes to those with at least 100 IPv6 addresses to increase the probability of 6Gen and Entropy/IP identifying patterns. Third, we take a random sample of at most 100 k IPv6 addresses per AS to use as input for 6Gen and Entropy/IP. The capped random sample ensures that we provide a balanced input for each AS. Fourth, we run Entropy/IP and 6Gen with the capped random sample as input to generate 1 M addresses for each AS separately. Fifth, we again take a random sample of at most 100 k of all generated addresses per AS for 6Gen and Entropy/IP, respectively. The capped random sample ensures that ASes with more generated addresses are not over-represented. Sixth, we perform active measurements to assess the value of generated addresses.

Learned Addresses

Entropy/IP generates 118 M addresses. Of those, 116 M are routable new addresses not yet in our hitlist. 6Gen produces slightly more addresses, 129 M, of which 124 M are new and routable. In total, we learn 239 M new unique addresses. Interestingly, there is very little overlap between 6Gen's and Entropy/IP's generated addresses: only 675 k addresses are produced by both tools, which equals to 0.2 % of all generated addresses.

Responsiveness of Learned Addresses

We probe the responsiveness of all 239 M learned addresses on ICMP, TCP/80, TCP/443, UDP/53, and UDP/443. 785 k IPv6 addresses respond to our probes, which corresponds to a response rate of 0.3 %. This low response rate underlines the challenges of finding new responsive addresses through learning-based approaches.

Comparing the responsiveness of addresses generated by 6Gen to Entropy/IP, we find that 6Gen is able to find almost twice as many responsive addresses: 489 k vs. 278 k. Our response rate for 6Gen is a lower bound: due to 6Gen's design, choosing the top generated addresses instead of random sampling would likely yield an even higher response rate.

In addition, both Entropy/IP and 6Gen found the same 17 k responsive addresses. The response rate of overlapping addresses generated by both tools is therefore 2.5 %, which is an order of magnitude higher than the general learned address population's 0.3 %. This demonstrates that Entropy/IP and 6Gen find complementing sets of responsive IPv6 addresses, with a small overlap of targets that are more likely to respond. Thus, it is meaningful to run multiple address generation tools even on the same set of input addresses.

Table 4.12: Top 5 responsive protocol combinations for 6Gen and Entropy/IP.

ICMP	TCP/80	TCP/443	UDP/53	UDP/443	6Gen	Entropy/IP
✓	✗	✗	✗	✗	66.8 %	41.1 %
✓	✓	✓	✗	✗	9.2 %	12.3 %
✗	✗	✗	✓	✗	7.3 %	23.1 %
✓	✓	✗	✗	✗	4.9 %	3.4 %
✓	✓	✓	✗	✓	3.2 %	6.1 %

When analyzing the top 5 protocols for responsive learned addresses in Table 4.12, we find particular differences between 6Gen and Entropy/IP. Two thirds of 6Gen responsive addresses answer to ICMP only, which is the case for only four out of ten Entropy/IP responsive addresses. On the other hand, Entropy/IP responsive hosts are three times more likely to be DNS servers (UDP/53). Moreover, 6Gen responsive hosts are half

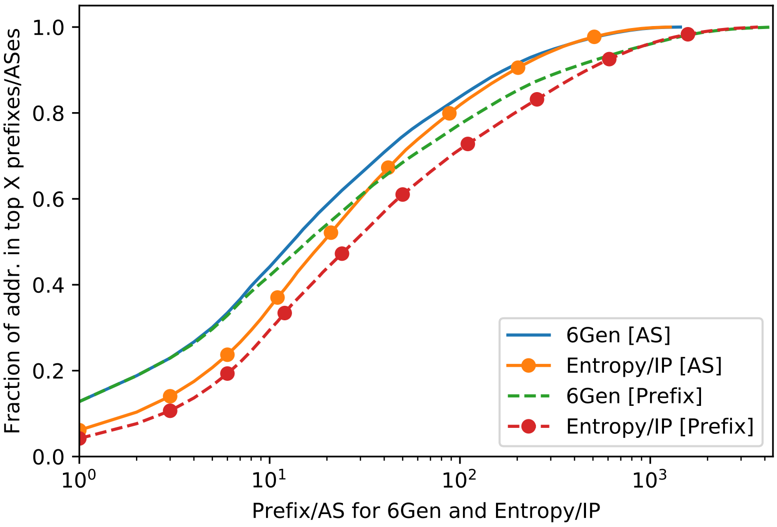
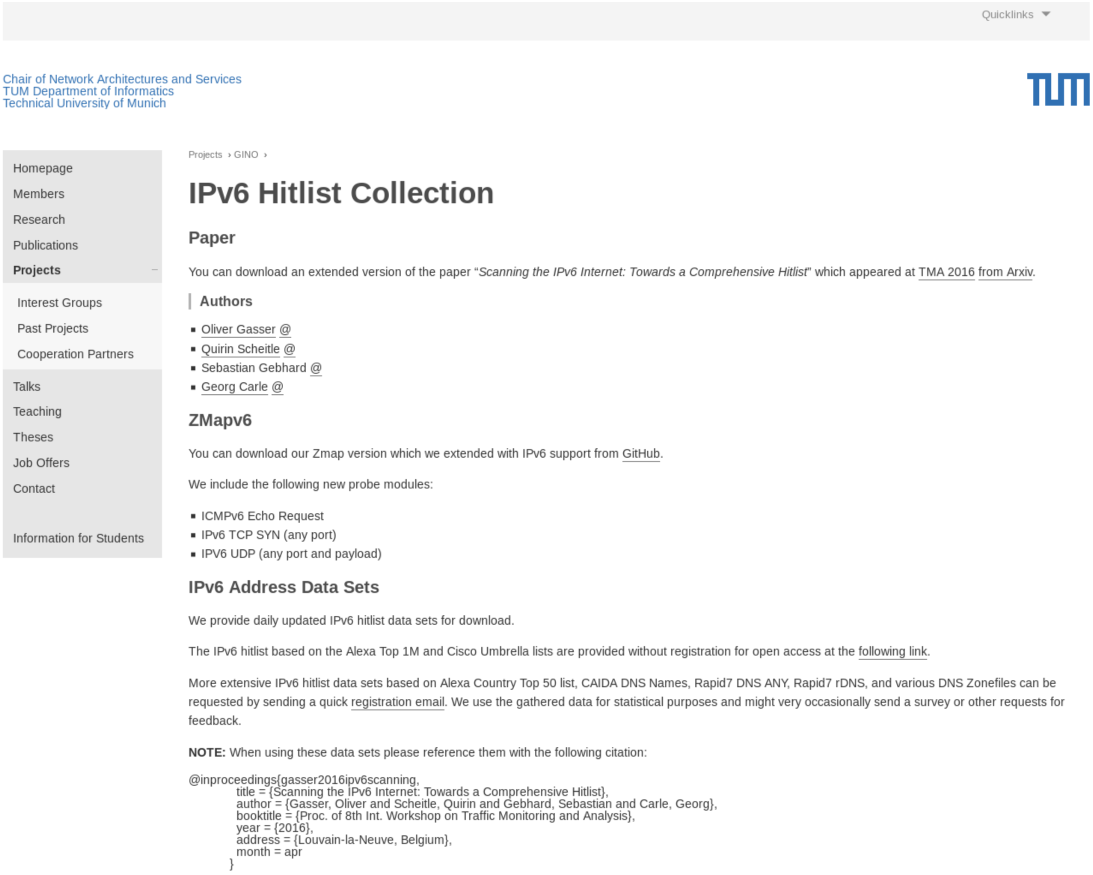


Figure 4.11: Prefix and AS distribution for responsive addresses generated with 6Gen and Entropy/IP.

as likely to be QUIC-enabled web servers (ICMP, TCP/80, TCP/443, and UDP/443) compared to Entropy/IP. This shows that 6Gen and Entropy/IP not only discover mostly non-overlapping addresses, but also different types of *populations* of responsive hosts. We suspect that these differences in populations stem from the inner workings of both tools: Entropy/IP's nature of finding patterns in addresses can identify certain structural addresses (e. g. DNS servers) better, whereas 6Gen's identification of regions with many addresses fares better with finding clients which are likely located in denser address regions.

Finally, we compare ASes and prefixes of responsive addresses for both tools. 6Gen discovers responsive hosts in 1442 ASes, while Entropy/IP does in 1275 ASes. Interestingly, responsive hosts in 384 ASes are found by only one of the tools, i. e., either 6Gen or Entropy/IP. In Figure 4.11, we show the prefix and AS distributions of responsive hosts. Entropy/IP's distribution is a bit less top-heavy compared to 6Gen's, where the top 2 responsive ASes make up almost 20 % of all addresses. Although there is some overlap in the top 5 ASes, 6Gen features more ISPs, like Sky Broadband, Google Fiber, and Xs4all Internet. In contrast, Entropy/IP's top ASes contain more CDNs and Internet services.

To summarize, 6Gen and Entropy/IP find few overlapping responsive addresses, but mostly in overlapping ASes. The services offered by these hosts differ considerably. Therefore, both tools have their advantages in finding specific addresses and populations. We suggest running both tools to maximize the number of found responsive addresses.



IPv6 Hitlist Collection

Paper

You can download an extended version of the paper "*Scanning the IPv6 Internet: Towards a Comprehensive Hitlist*" which appeared at TMA 2016 from Arxiv.

Authors

- Oliver Gasser @
- Quirin Scheitle @
- Sebastian Gebhard @
- Georg Carle @

ZMapv6

You can download our Zmap version which we extended with IPv6 support from GitHub.

We include the following new probe modules:

- ICMPv6 Echo Request
- IPv6 TCP SYN (any port)
- IPV6 UDP (any port and payload)

IPv6 Address Data Sets

We provide daily updated IPv6 hitlist data sets for download.

The IPv6 hitlist based on the Alexa Top 1M and Cisco Umbrella lists are provided without registration for open access at the following link.

More extensive IPv6 hitlist data sets based on Alexa Country Top 50 list, CAIDA DNS Names, Rapid7 DNS ANY, Rapid7 rDNS, and various DNS Zonefiles can be requested by sending a quick registration email. We use the gathered data for statistical purposes and might very occasionally send a survey or other requests for feedback.

NOTE: When using these data sets please reference them with the following citation:

```
@inproceedings{gasser2016ipv6scanning,
        title = {Scanning the IPv6 Internet: Towards a Comprehensive Hitlist},
        author = {Gasser, Oliver and Scheitle, Quirin and Gebhard, Sebastian and Carle, Georg},
        booktitle = {Proc. of 8th Int. Workshop on Traffic Monitoring and Analysis},
        year = {2016},
        address = {Louvain-la-Neuve, Belgium},
        month = apr
       }
```

Figure 4.12: Screenshot of IPv6 Hitlist Collection website.

4.4 Continuous Provisioning of IPv6 Measurement Results

To facilitate IPv6 research of fellow researchers we provide results from daily IPv6 measurements available for download.

4.4.1 IPv6 Hitlist Collection

The IPv6 Hitlist Collection is available at `https://www.net.in.tum.de/projects/gino/ipv6-hitlist.html` [102]. Figure 4.12 shows a screenshot of the website where fellow researchers can find information about the paper, used software, and the IPv6 Hitlist Collection.

We provide two versions of the IPv6 Hitlist Collection: the open version and the public version. The former can be accessed right away, to access the latter you need to send a brief registration email. This helps us to assess the value of our hitlist for other researchers.

The IPv6 Hitlist Collection contains IPv6 addresses from daily DNS resolution measurements and other IPv6 data extracted from various sources:

The open dataset contains:

Alexa Top 1M This dataset contains IPv6 addresses gathered from resolving domains from the Alexa Top 1M [9] for AAAA DNS records. The dataset is available since April 2017.

Statvoo This dataset contains IPv6 addresses gathered from resolving domains from the Statvoo Top Websites [231] for AAAA DNS records. The dataset is available between March and May 2018, since its publication was discontinued by Statvoo.

Umbrella This dataset contains IPv6 addresses gathered from resolving domains from the Cisco Umbrella Popularity List [52] for AAAA DNS records. The dataset is available since April 2017.

The public dataset which can be accessed after sending a registration email contains:

Alexa Country This dataset contains IPv6 addresses gathered from resolving domains from the Alexa Top Sites by Country [10] for AAAA DNS records. The dataset is available since April 2017.

CAIDA DNS Names This dataset contains IPv6 addresses extracted from the CAIDA IPv6 DNS Names Dataset [40]. The dataset is available since April 2017.

Certificate Transparency This dataset contains IPv6 addresses gathered from CommonName and SubjectAlternativeName entries in certificates downloaded from Certificate Transparency [156] logs and resolving them for AAAA DNS records. The dataset is available since April 2017.

RIPE IPmap This dataset contains IPv6 addresses extracted from RIPE NCC's IPmap dataset [201]. The dataset is available since April 2018.

Rapid7 ANY DNS This dataset contains IPv6 addresses extracted from Rapid7's ANY DNS dataset [197]. The dataset is available since April 2017.

RIPE Atlas This dataset contains IPv6 addresses extracted from RIPE Atlas traceroute dataset [202] which contains mostly router addresses. The dataset is available since April 2017.

Zonefiles This dataset contains IPv6 addresses gathered from resolving domains from various zonefiles for AAAA DNS records. The dataset is available since April 2017 for zonefiles from the following TLDs: .au, .biz, .com, .de, .info, .mobi, .net, .nu, .org, .se, .sk, .xxx, and novel TLDs shared under the ICANN's Centralized Zone Data Service [135].

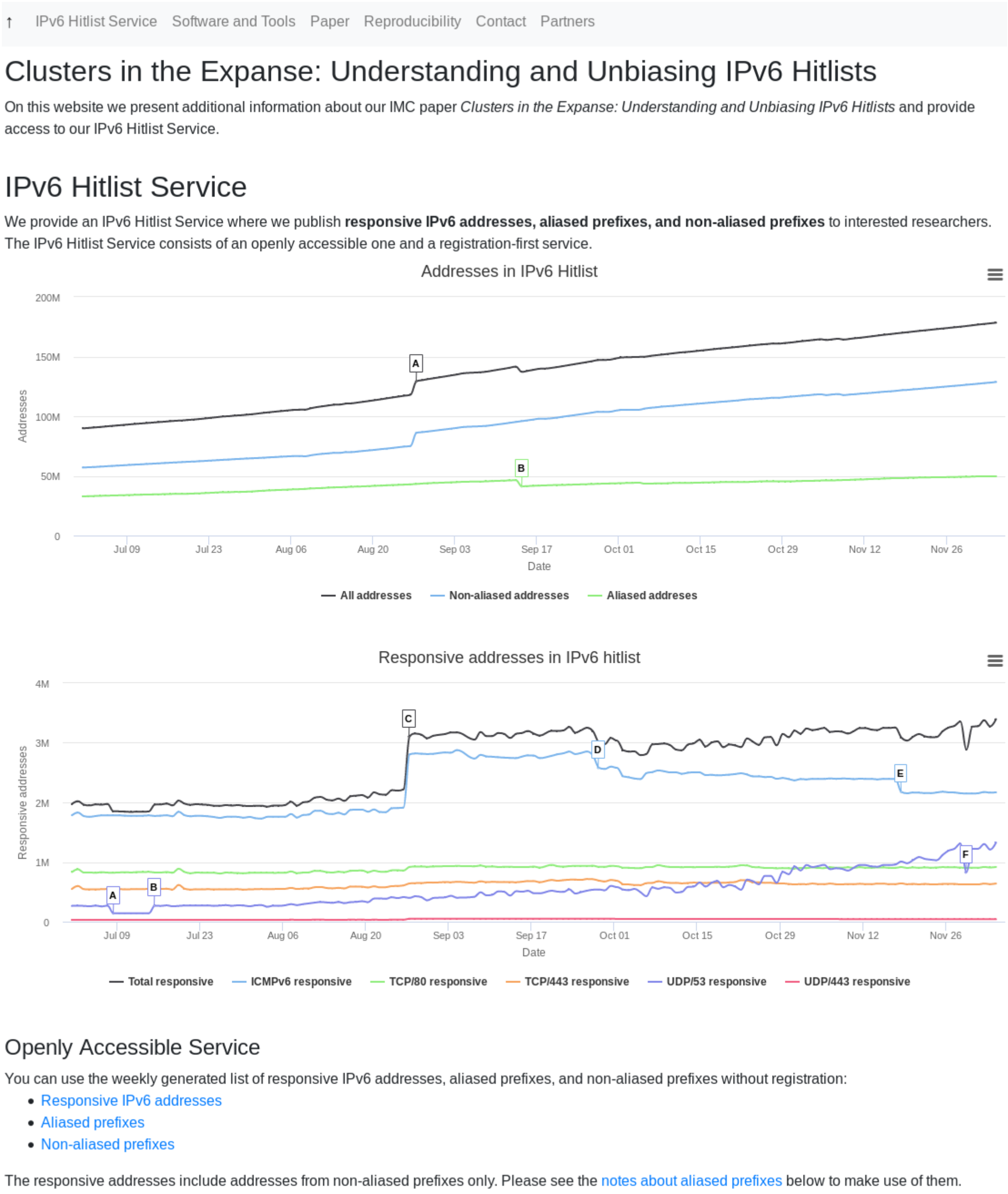
We also provide access to legacy IPv6 Hitlist Collection files for some of the sources between February and June 2016. This legacy datasets were results from measurements on a different measurement infrastructure.

As of December 6, 2018 there are 24 fellow researchers registered for the public dataset, who use the data for IPv6 measurement research. Several papers who make use of our data sharing were published at conferences: They used our data to evaluate the HTTPS ecosystem [16, 101, 215], characterize load-balancing behavior in the IPv6 Internet [13], or to improve IPv6 topology discovery [27]. In addition, the hitlist has also seen non-academic use [115].

4.4.2 IPv6 Hitlist Service

To make our analyses [100] on responsiveness of IPv6 addresses over time and aliased prefixes available to other researchers, we launched the IPv6 Hitlist Service at `https://ipv6hitlist.github.io/` [103] in 2018. Figure 4.13 shows a screenshot of the website where fellow researchers can find information about the IPv6 Hitlist Service, the accompanying research paper, our used software and tools, and information on reproducibility.

What is most apparent in the comparison between the newer IPv6 Hitlist Service compared to the older IPv6 Hitlist Collection websites is that the former features interactive graphs which give visitors a fast way to grasp the growth and development of known IPv6 addresses and displays information on reachability statistics over time for ICMPv6, HTTP, HTTPS, DNS, and QUIC protocols. These graphs are updated automatically on a daily basis to always provide visitors with up-to-date information. In addition, we also mark significant events (e. g. if a prefix stops responding to our probes) which allows readers to more easily understand abrupt changes in input size and responsiveness. From an implementation perspective we store the input data for the graphs as a JSON file which can be accessed by anyone on GitHub. We then use the Highcharts [124] JavaScript library to create the interactive plots.



Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists

On this website we present additional information about our IMC paper *Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists* and provide access to our IPv6 Hitlist Service.

IPv6 Hitlist Service

We provide an IPv6 Hitlist Service where we publish **responsive IPv6 addresses, aliased prefixes, and non-aliased prefixes** to interested researchers. The IPv6 Hitlist Service consists of an openly accessible one and a registration-first service.

Openly Accessible Service

You can use the weekly generated list of responsive IPv6 addresses, aliased prefixes, and non-aliased prefixes without registration:

- Responsive IPv6 addresses
- Aliased prefixes
- Non-aliased prefixes

The responsive addresses include addresses from non-aliased prefixes only. Please see the notes about aliased prefixes below to make use of them.

Figure 4.13: Screenshot of IPv6 Hitlist Service website.

4.5 Discussion and Implications on Future IPv6 Measurements

This section is based on the publications "Scanning the IPv6 Internet: Towards a Comprehensive Hitlist" by Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle, which was published at TMA 2016 [104]; "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists" by Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczyński, Stephen D. Strowes, Luuk Hendriks, and Georg Carle, which was published at IMC 2018 [100]. In Section 4.7 details on the author's contributions and differences between the thesis text in relation to the published papers are given.

In this section we discuss the results from our IPv6 hitlist measurements and lay out its implications and lessons learned for future work.

4.5.1 Tailoring Hitlist Sources

In our evaluations we present many different sources to create a hitlist for IPv6 measurements. However, the value and usefulness of each of these sources depends on the research question to be answered and consequently the type of scan to be carried out. Moreover, not all of these sources might be available to every researcher. In addition, the effort in terms of data storage, processing power and network bandwidth should not be underestimated. Therefore we dedicate this section to recommend the most efficient combination of sources tailored specifically to the type of scan in question.

Public vs. Non-Public Sources

The question on which sources to use for your IPv6 hitlist, depends mostly on which sources are available to you In our new study [100] we exclusively use active sources which are available to everyone. This allows to share the results with fellow researchers and makes it possible to reproduce our research.

Even though non-public sources such as IPv6 addresses obtained from passive measurements [98] have their drawbacks, they can also provide valuable insights into client developments and usage of different services.

Excluding Certain Addresses

To reduce bias when conducting IPv6 measurements, we generally advise to strive for an evenly balanced hitlist across prefixes and ASes, and to remove addresses in aliased prefixes. Depending on the goal of the study, researchers can pivot from an even address distribution to a stronger focus on certain address types (e. g. HTTPS web servers). Depending on the type of study it may also be desirable to include or exclude specific data sources. For example, studies analyzing hosting providers can use server addresses, while for residential networks, researchers can focus on sources containing mainly CPE and client addresses.

Internet Structure

Evaluating the Internet structure aims at finding as many routers and transit links as possible. Therefore, it is of essence to maximize the count of ASes and announced prefixes in the hitlist (in contrast to maximizing raw IP address count). Using scamper or other tracerouting tools such as yarrp provide a large number of router IPv6 addresses. Note that the resulting addresses might also contain CPE devices, which might be included or excluded from measurement depending on the type of study.

Assessing Security Posture

Empirically assessing the Internet's security posture aims at scanning as many responsive hosts as possible, although frequently only servers are of interest.

4.5.2 Time-to-Measurement

We collect IPv6 addresses from a variety of sources containing server, router, and clients addresses. Our analysis shows that server IPv6 addresses are more responsive and stable in comparison to CPE and client devices. As a result, when using an IPv6 hitlist as an input for a specific measurement study, researchers need to consider the time-to-measurement: client devices need to be measured within minutes to obtain sensible response rates, whereas servers remain responsive over weeks.

4.5.3 Raw Number of IP Addresses as a Metric

Our research showed that a simple count of IPv6 addresses is not valuable: First, privacy extensions dominate the observations at passive sources and keep the IPv6 address count growing almost linearly, while the number of Autonomous Systems and prefixes found quickly saturates. Second, few addresses form a stable core and are frequented by many clients: Almost half of all traffic at passive sources is directed to a stable set of HTTP(S) servers. Third, this stable core covers a significant part of ASes and prefixes: We find that servers cover about half of all seen prefixes and more than half of all seen ASes. This is due to an AS announcing several prefixes. Covering one of these prefixes already results in counting the AS.

While our 2018 study's focus on responsive addresses is reasonable for a hitlist which is used as direct input for a measurement study, there might be scenarios where also unresponsive addresses could be of value. Unresponsive addresses can be used to understand addressing schemes inside a prefix. They can also be used as an input for address learning algorithms (e. g. Entropy/IP or 6Gen) which might then output responsive addresses.

4.5.4 Providing an IPv6 Hitlist Service

As laid out in Section 4.4 we provide daily updated lists of IPv6 addresses from various sources, responsiveness of these addresses on five different protocols, and lists of aliased prefixes.

These hitlist services already provided value for other researchers [13, 16, 27, 101, 215], therefore we also strive to publish results from IPv6 measurement in the future.

4.6 Key Contributions of This Chapter

This chapter addressed research question RQ II which consists of tackling challenges C 3, C 4, C 5, and C 6. We analyzed passive and active sources for IPv6 addresses and identified imbalanced bias in order to reduce bias in IPv6 hitlists. In addition we provided the IPv6 Hitlist Service to fellow researchers. As a result we answered the question by showing different types of biases in IPv6 hitlist sources.

In the following we list the key contributions of this chapter:

Passive sources We analyze passive IPv6 address sources—flows collected at an Internet Exchange Point and packet data from the Munich Scientific Network—and clearly identified weekly patterns. Due to the client nature of the majority of addresses, we found that measurements needed to be conducted swiftly, before targets go offline. Finally, we evaluate the interface identifier distribution of IPv6 addresses and found high shares of privacy extension addresses, another indicator for the short-lived nature of addresses from passive sources. With the analysis of passive sources we tackled challenge C 3.

Active sources We analyzed IPv6 addresses obtained from a multitude of active sources: domain lists, Certificate Transparency, DNS ANY, DNS rDNS, DNS AXFR, the TLDR project, the Bitnodes API, RIPE Atlas, and traceroutes. We evaluated the runup of addresses per source over time and identified clusters of addresses in specific prefixes and sources. Using this analysis of active address sources we addressed challenge C 4.

Bias in IPv6 hitlists We investigated the bias of IPv6 address hitlists. We identified clusters of addresses in always-responding prefixes, so called “aliased prefixes”. We performed in-depth analyses of these aliased prefixes and provided an algorithm to detect and filter these prefixes. In addition, we analyzed biases stemming from imbalanced prefix and AS distributions. We also applied two different address learning algorithms and compared the responsiveness and nature of newly discovered responsive hosts. Finally, we evaluated longitudinal responsiveness and identified source-protocol combinations with differing response rates after a few days. With the in-depth bias and clustering analysis we addressed challenge C 5.

IPv6 Hitlist Service With the IPv6 Hitlist Service we have started providing a measurement service for fellow researchers. By providing access to raw results from daily running IPv6 measurements we have been providing researchers with up-to-date measurement results and updated time series graphs. From registration information we know that a few dozen researchers are actively using this service in various efforts to further the understanding of the Internet and improve the Internet ecosystem and its security. With the IPv6 Hitlist Service we successfully tackled challenge C 6.

With overcoming these four challenges we can provide an answer to research question RQ II, as we find different levels of imbalanced and biased sources.

4.7 Statement on Author's Contributions

This chapter is partially based on the publications "Scanning the IPv6 Internet: Towards a Comprehensive Hitlist" by Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle, which was published at TMA 2016 [104]; "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists" by Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczyński, Stephen D. Strowes, Luuk Hendriks, and Georg Carle, which was published at IMC 2018 [100]. The IPv6 hitlist work was first started as part of Sebastian Gebhard's Master's thesis ("IPv6 Scanning - Smart address selection and comparison to legacy IP") [108], which the author advised.

The author made the following contributions to the IPv6 hitlist study presented in the TMA 2016 paper [104] and the adapted sections in this chapter. The author contributed significantly to the implementation of ZMapv6. Additionally, the author contributed significantly to carrying out the active measurements and contributed to the passive data collection process. The author contributed to the overall evaluation and made significant contributions in the hamming weight analysis.

In the following we describe changes between Sections 4.2, 4.3 and 4.5 and the IPv6 hitlist study presented in the TMA 2016 paper [104]. We adapted certain tables to running text, to present highlights which are interesting in this dissertation's context. In addition, we shortened certain tables to highlight certain findings (e. g. most important port-protocol combinations) which are relevant for this thesis. We focus on the findings of passive sources and only highlight specific findings of active sources (e. g. hamming weight analysis), as most of the active sources analysis is superseded by the IMC 2018 study. We combine the discussion and recommendation points raised in the TMA 2016 study with our discussion in the IMC 2018 study.

The author made the following contributions to the IPv6 hitlist study presented in the IMC 2018 paper [100] and the adapted sections in this chapter. The author coordinated and lead the group of researchers in this study. The author provided significant contributions in the study's design and execution. Moreover, the author contributed significantly in the analysis of hitlist sources and the inception of the aliased prefix detection methodology. The author contributed to the fingerprinting of aliased prefixes. Additionally, the author contributed significantly to the address probing and learning of new addresses analyses, and the discussion of implications on IPv6 hitlists.

In the following we describe changes between Sections 4.3 and 4.5 and the IPv6 hitlist study presented in the IMC 2018 paper [100]. We only include paper parts which are relevant in the context of this dissertation and omit other parts (e. g. crowdsourcing analysis). We include a larger version of the responsiveness over time figure, which was not feasible in the constrained space of the IMC 2018 paper. We shorten the fingerprinting aliased prefixes part and highlight its findings. We combine the discussion and recommendation points raised in the IMC 2018 study with our discussion in the TMA 2016 study.

Part III

Network Security Measurements

Chapter 5

HTTPS Ecosystem

In this chapter we present analyses on the security of the HTTPS ecosystem. In the following Section 5.1 we discuss various HTTPS security techniques and their evolution in the HTTPS ecosystem. Subsequently in Section 5.2 we evaluate a relatively new technology, namely Certificate Transparency, and assess its impact on the HTTPS ecosystem. Next, we lay out the key contributions of this chapter in Section 5.3. We conclude the chapter with the statement on the author's contributions in Section 5.4.

5.1 HTTPS Security Techniques

This section is based on the publication "Mission Accomplished? HTTPS Security after DigiNotar" by Johanna Amann, Oliver Gasser, Quirin Scheitle, Lexi Brent, Georg Carle, and Ralph Holz, which was published at IMC 2017 [16]. In Section 5.4 details on the author's contributions and differences between the thesis text in relation to the published papers are given.

Due to the many issues discovered in the HTTPS ecosystems [15, 73, 129, 183, 195], a number of countermeasures were put forward. In this study we look at six of these HTTPS security techniques which strive to enhance the HTTPS ecosystem's security posture: Certificate Transparency [146], HTTP Strict Transport Security [127], HTTP Public Key Pinning [83], SCSV Downgrade Prevention [169], Certification Authority Authorization [117], and TLS Authentication [128].

In this section we focus on the two HTTP-based techniques HTTP Strict Transport Security and HTTP Public Key Pinning. We analyze Certificate Transparency in detail in Section 5.2. The remaining three techniques are evaluated in detail in our research papers [16, 215].

5.1.1 Background

In this section we give background information on the surveyed techniques, HSTS and HPKP.

HSTS HTTP Strict Transport Security (HSTS) [127] is a mechanism added to the HTTPS ecosystem to avoid downgrade attacks. Servers can send the HTTP header `Strict-Transport-Security` to signal clients that they must connect using HTTPS only in future connections. This avoids potential downgrade attacks where a Man-in-the-Middle intercepts an HTTPS connection attempt and makes the victim downgrade to plaintext HTTP. Servers can send a duration (using the `max-age` parameter) specifying how long clients should store the server's requirement to connect via HTTPS. Servers can also force HTTPS for all its subdomains via the `includeSubDomains` flag.

HPKP HTTP Public Key Pinning (HPKP) [83] is a mechanism added to the HTTPS ecosystem to pin specific TLS public keys to a domain. Servers can send the HTTP header `Public-Key-Pins` to signal clients that they must only establish future TLS connections if the provided TLS certificate chain contains a pinned key. HPKP is specifically aimed to counter the possibility of Man-in-the-Middle attacks in the case of compromised CAs. Servers can pin more than one key, e. g. a main and a backup key in case a certificate needs to be exchanged due to a security breach. Similar to HSTS, HPKP also allows for `max-age` and `includeSubDomains` parameters to specify the duration of a pin and potential inclusion of subdomains in the specified pinning.

To counter the vulnerability to interception in the first connection, both HSTS domains and HPKP domains are shipped in browsers in so-called preloading lists [48, 49, 171, 172].

5.1.2 Related Work

Our research stands in the long line of research analyzing security properties of the TLS and HTTPS ecosystems. Studies have previously analyzed particular segments of these ecosystems. Researchers analyzed the state of the PKI [8, 17, 73, 129], evaluated security in communication protocols [74, 130], surveyed the practices regarding certificate revocation [258, 262, 264], analyzed cryptographic properties of TLS and their weaknesses [3, 123, 132], and examined problems stemming from faulty implementations [26].

Most closely related to our work surveying HTTPS security techniques, VanderSloot *et al.* [250] examine the HTTPS ecosystem from several perspectives, including active scans, passive monitoring and Certificate Transparency logs. The authors use *.com*, *.net*, and *.org* domains to scan 153 M domains. We extend this approach by adding domains from *.biz*, *.info*, *.mobi*, *.sk*, and *.xxx* from PremiumDrops [192]; *.de* and *.au* from ViewDNS [253]; from the Alexa [9] and Umbrella [52] Top 1M, all Alexa Country Top 50 [10], plus domains from 748 zones from ICANN's Centralized Zone Data Service [135].

The following publications target specific aspects of the HTTPS ecosystem security. Clark and van Oorshot [53] theoretically studied the effects of HTTP extension headers in 2012. Kranch and Bonneau [148] study the deployment of HSTS and HPKP based on both the preload and the Alexa Top 1M lists. De los Santos *et al.* [210] analyze the implementation of HSTS and HPKP for several dozen domains using Shodan. Given the novelty of both standards, we find the uptake of HSTS and HPKP to have significantly changed since these early studies.

5.1.3 Methodology

We conduct active scans from the University of Sydney (IPv4), and the Technical University of Munich (IPv4 & IPv6). Our scans are based on domain names as opposed to IP addresses. This captures SNI-based servers [51, 250] and avoids accidentally connected devices. In total we perform measurements to 193 M domain names, about 58 % of the 330.6 M registered domains in March 2017 [251].

We resolve domains from both Munich (TUM) and Sydney (USyd) using a modified version of massdns [29] and an unmodified version of unbound [154]. From Munich, we find 154 M IPv4-enabled and 9.7 M IPv6-enabled domains, with 9.5 M intersecting domains. From Sydney, we considered only A records, as the university network does not support IPv6. 650 k (0.4 %) fewer domains could be resolved. This is within expectations: Rijswijk-Deij *et al.* [200] show that daily deviations of around 0.6 k are expected for large-scale DNS measurements.

IP addresses learned from our DNS scans are port-scanned using an IPv6-enabled version of ZMap [244]. We perform TLS handshakes using goscanner [241], a custom highly-parallelized scanning tool. goscanner connects to each IP address, sending the domain name in the SNI extension, one name per connection.

If we can establish a TLS connection, we send an HTTP `HEAD` request to obtain HSTS and HPKP headers. In about 50 % of cases, we receive an HTTP 200 ("OK") response code. In the remaining cases, we receive mainly redirect codes, error codes, or no HTTP response at all.

Table 5.1: Overview of DNS resolutions and active scans, conducted from April 11 through April 16, 2017.

# of	TUM IPv4	USyd IPv4	TUM IPv6	Related Work
Input Domains	192.9M	192.9M	192.9M	≈153M [200, 250]
Domains $\geq$ 1 RR[1]	153.5M	152.9M	9.7M	149M [200]
IP addresses	8.8M	8.9M	6.2M	
TCP/443 SYN-ACKs	4.0M	3.2M	316k	249k [104]
<domain,IP> pairs	80.4M	79.2M	11.0M	
Successful TLS SNI[2]	55.7M	58.0M	5.1M	42M [250]
HTTP response 200 SNIs	28.4M	28.1M	1.9M	

1: Domains that server 1 or more Resource Records of A or AAAA type.

2: <Domain,IP> tuples with successful TLS SNI connections.

Table 5.2: Unique HTTP code 200, HSTS, and HPKP domains responding to MUCv4, SYDv4, SYDv6, and any scan. Last row displays domains with consistent headers across scans.

	HTTP 200	HSTS	HPKP
MUC IPv4	26.8M	960.0k (3.59 %)	5.9k (0.02 %)
SYD IPv4	26.5M	948.5k (3.58 %)	5.8k (0.02 %)
MUC IPv6	1.2M	38.8k (3.36 %)	1.0k (0.09 %)
Total	27.8M	1.0M (3.60 %)	6.2k (0.02 %)
Consistent	27.8M	984.1k (3.54 %)	6.2k (0.02 %)

Table 5.1 provides an overview of scan results along our scanning chain, across locations and protocols.

Ethical Considerations

We minimize interference by following best scanning practices, such as those outlined in [71], by maintaining a blacklist and using dedicated servers with informing rDNS names, websites, and abuse contacts. We assess whether data collection can harm individuals or reveal private information as proposed by [67, 186].

5.1.4 HSTS and HPKP Evaluation

In this section we analyze deployment, consistency, lifetime, and cryptographic validity of the received HSTS and HPKP headers. Table 5.2 provides an overview of domains responding with HTTP 200 (“OK”) to our requests. We merge responses from the same domain if the HSTS and HPKP headers are consistent across all IP addresses.

Header Consistency

We first investigate intra-scan consistency, i. e., whether the headers for a domain are consistent within each of the TUMv4, TUMv6, and SYDv4 measurements. For each scan, we find a tiny fraction (≈ 0.003 %) of domains exhibiting inconsistent behavior, largely (>65 % of inconsistent cases) caused by domains setting HSTS or HPKP headers on one set of IP addresses, but not on another. Many of the remaining cases send inconsistent HSTS *max-age*, or *includeSubDomains* values; one domain pins different HPKP keys. The inconsistent domains within individual scans are 6 for MUCv6, 25 for MUCv4, and 22 for SYDv4, a tiny fraction compared to the millions of scanned domains per vantage point. This group of inconsistent domains partially consists of services that are globally distributed under shared administration, such as pooled NTP and OpenPGP key servers. We limit the following analyses to domains consistent within one scan.

In the following we evaluate inter-scan consistency, i. e., whether headers are consistent between scans. We find about 2 % of HSTS/HPKP-enabled domains to serve different headers across at least two scans; the difference is nearly exclusively caused by configurations serving HSTS in one scan, but not the other. In detail, we find 15 k domains inconsistent between the MUCv4 and SYDv4 scan, and 754 domains inconsistent between the MUCv4 and MUCv6 scan. From sample analysis, we identify three potential reasons for this behavior: (1) timing differences between our scans, (2) differently configured IP anycast services, revealed using speed-of-light constraints, or (3) load-balancers with inconsistently configured servers. We limit the following analyses to domains that serve consistent headers across all scans.

Deployment

We now analyze deployment of HSTS and HPKP headers across domains.

3.5 % (984 k) of the domains answering with HTTP-200 and with consistent behavior provide support for HSTS. 0.2 % of HSTS-enabled domains send incorrect HSTS headers—typically due to typographical mistakes, such as *includeSubDomains* missing the plural *s*. 41 k domains do not use HSTS effectively, setting the *max-age* attribute to 0, resulting in a “deregistration” from HSTS use (24 k domains), to a non-numerical value (16 k domains), or to an empty value (1 k domains).

For HPKP, we find that only 6181 of all 28 M domains (0.02 %) send an HPKP header. Of these, 29 do not send a valid *max-age* directive and 12 do not contain any pins.

In the following we analyze deployment of different HSTS and HPKP attributes:

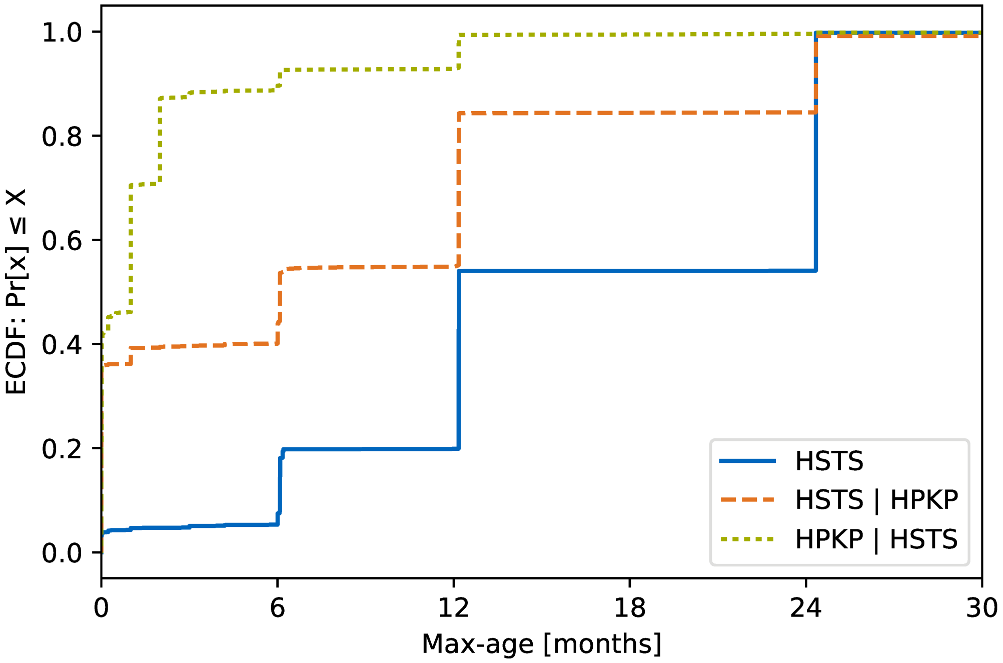


Figure 5.1: Distribution of the *max-age* attribute for HPKP and HSTS headers: Domain owners typically set much higher max-ages for HSTS than HPKP.

Max-Age: The *max-age* attribute indicates the lifetime of HSTS and HPKP headers, which browsers will update on every domain visit. Figure 5.1 shows the distribution of *max-age* across all HSTS domains, HPKP's *max-age* for the subset of domains that also support HSTS ($\text{HPKP}|_{\text{HSTS}}$), as well as HSTS's *max-age* for the subset of domains that also support HPKP ($\text{HSTS}|_{\text{HPKP}}$). The intersecting sets generally have shorter durations with the majority of HPKP *max-age* values being 10 minutes (33 %), 30 days (22 %), and 60 days (15 %). HSTS domains that also send HPKP headers choose 5 minutes (32 %), 1 year (26 %), and 2 years (14 %). The largest values are sent by the set of all HSTS domains with 2 years (46 %), 1 year (32 %), and 6 months (10 %). The median *max-age* of HSTS is one year, but only one month for HPKP. This suggests operators exercise caution when using HPKP, which carries high availability risk through lock-out (cf. the Cryptocat lock-out [248]). We also note an extreme outlier setting an HSTS *max-age* of 49 million years (a likely accidental duplication of the string for half a year).

includeSubDomains: 56 % of HSTS and 38 % of HPKP domains use the attribute *includeSubDomains* in their header. This attribute enables HSTS for all subdomains of the domain setting this attribute. This has many benefits, for example, helping to avoid insecure cookies from subdomains. However, it may cause operational difficulties when subdomains do not actually support HTTPS. As some domains do not even use subdomains, it is difficult to assess this percentage.

Preloading Lists: We investigate the HSTS *preload* list included in Chrome [49], which also is the base for Mozilla's preloading list for Firefox [172]. A domain can be added to the list by (1) setting the HSTS directive, (2) including the non-RFC *preload* parameter, and (3) opting in through sites such as Chromium's `hstspreload.org`. Interestingly, we find a large fraction (379 k domains, 38 %) of scanned domains to include the *preload* directive, but only 23 k domains in the preload list of the current version of Chrome (version 58 at the time of the measurement), with the intersection consisting of just 6 k domains. Two possible explanations for this anomaly are that the inclusion process is slow to catch up, or operators do not follow all prescribed steps for inclusion, e. g. because they just copy-and-paste directives from tutorials.

Our scans include 13 k of the 23 k domains in the preload list (the remainder are domains without A/AAAA records, from TLDs not in our list, or subdomains we do not scan). We successfully connect to 6.6 k of these 13 k domains. 6026 of them send the HSTS header and 5656 include the *preload* attribute. The remaining domains do not satisfy the preloading criteria anymore and will be removed from the preloading list eventually. Further examination of HSTS-preloaded domains reveals that some popular domains only preload subdomains, but not their base domain: for the Alexa Top 1M list, 91 of the 2715 preloaded domains only preload a subdomain. One example is *theguardian.com*, which sets dynamic and preloaded HSTS for its *www* subdomain, but not for its base domain. This exposes users of the base domain to HTTPS stripping and redirect attacks. Another example is Google who enables HSTS for select subdomains. We contacted Google who stated that each service and its subdomain has to be verified individually, making the HSTS roll-out a long process.

There is no publicly accessible preloading mechanism for HPKP. Browser vendors, however, include important domains in their internal HPKP preloading list. Mozilla's HPKP preloading list, which extends Chrome's list [171, 172], includes a total of 479 domains, mostly from Google, Facebook, Yahoo, Twitter, Mozilla, and the Tor project.

Public Key Pinning: We analyze the validity of HPKP pins. The majority (86.0 %) of scanned HPKP domains use HPKP correctly and provide at least one valid pin. Examining non-matching cases reveals that for 8.5 % of HPKP domains the certificate is known to us, but missing from the handshake. Exploring the top 5 cases reveals 4 intermediate CA certificates missing from the handshake (a TLS standard violation, but accepted by browsers) and one certificate falsely copied from an HPKP tutorial website.

The majority of the remaining 5.5 % of HPKP domains with pins where we find no matching public key in our certificate set use bogus pins, many being syntactically invalid SHA256 hashes. The top 3 are the pins from the RFC example section, the text *<Subject Public Key Information (SPKI)>*, and *base64+primary==*, *base64+backup==*. Pins that do not have the correct format are ignored by browsers.

Deployment Ranking: Figures 5.2 and 5.3 differentiate HSTS and HPKP usage for both dynamic and preloaded deployment across domain rank. Note that our 100 % baseline is the fraction of Top 1M, Top 10k, and Top 1k domains answering with HTTP 200, amended by all preloaded domains. For both technologies, we find that few domains of the base population deploy them. The rising share of dynamic and preloaded domains with domain popularity is encouraging. This is in line with expectations that more popular domains also have more resources to configure and maintain these security extensions. The share of preloading among top domains, especially for HPKP, also is encouraging.

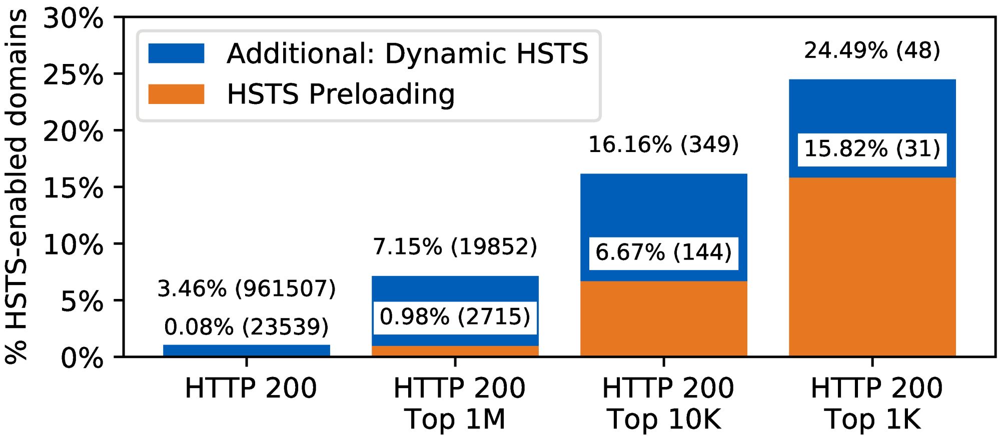


Figure 5.2: HSTS: Significant usage among top domains. Preloading essentially absent in general population, but with significant deployment among top domains.

Comparison to Related Work

We compare our results to those by Kranch and Bonneau [148], who evaluated HSTS and HPKP extensively in 2014. Both technologies have gained much usage, in both dynamic and preloaded fashion: The HSTS preload list has grown from 1258 domains in 2014 to 23.5 k domains in 2017, and the number of dynamic HSTS domains in the

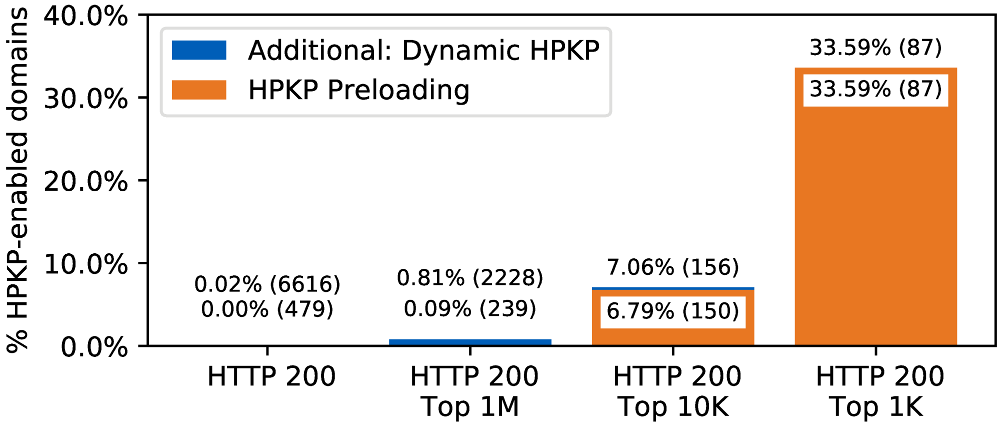


Figure 5.3: HPKP: Low usage among general population, significantly higher usage through preloading list among Alexa top domains.

Alexa Top 1M list has grown from 12.5 k in 2014 to a lower bound of 18 k in 2017. As HPKP was still in the standardization process in 2015, Kranch and Bonneau found only 18 domains supporting HPKP. This number has risen to about 6 k in our measurement. We can confirm many of the issues and oddities observed by Kranch and Bonneau, such as mistyped directives, mismatches between preloaded and dynamic HSTS domains, and redirections from top domains not covered under HSTS. We agree with their conclusion that forcing all operators to determine and configure a *max-age* is prone to mistakes, and support their suggestion of a reasonable default setting.

5.1.5 Comparison to Other Techniques

In this section we compare HSTS and HPKP results to other HTTPS security techniques. First, we map protection mechanisms against attack vectors and assess the number of protected domains. Second, we relate our findings to the deployment effort and risks to site availability.

Correlation of Security Feature Application

In this section, we investigate the correlation between deployment of different features. Table 5.3 shows the conditional probability for a feature Y to be effectively deployed given that another feature X is effectively deployed.

The lower left triangle of the matrix shows that deployment of a frequently deployed feature such as SCSV or CT does not imply the use of less common features. The upper right triangle offers more interesting insights, of which we discuss the highlighted cells here:

Table 5.3: $P((Y|X)$ in %, giving the empirical probability that technology Y is deployed when X is. For comparability across all features, sets only contain HTTP 200 domains, making these numbers incompatible to in-depth analysis per feature.

Y↓ , X→	SCSV	CT	HSTS	HPKP	CAA	TLSA	Top 1M	HTTP 200
n	26M	2.3M	944k	6k	1.2k	0.5k	0.3k	28M
SCSV	100.00	95.65	67.86	96.03	92.57	91.49	96.08	94.94
CT	8.37	100.00	7.10	45.88	12.14	13.54	13.36	8.31
HSTS	2.43	2.90	100.00	92.21	49.12	70.21	6.60	3.40
HPKP	0.02	0.12	0.61	100.00	9.82	19.92	0.72	0.02
CAA	0.00	0.01	0.07	1.98	100.00	14.70	0.04	0.00
TLSA	0.00	0.00	0.04	1.66	6.07	100.00	0.02	0.00
Top 1M	1.01	1.60	1.93	32.12	8.63	8.32	100.00	0.99
HTTP 200	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

First, effective deployment of SCSV is less frequent for domains that use HSTS. Further investigation of this intriguing fact reveals that 280 k domains, roughly equal to the drop-off compared to average deployment are hosted by the controversial [11, 12, 32, 88, 236] provider Network Solutions/web.com. The hoster apparently enabled HSTS for a large set of domains, without support for SCSV and even does not provide valid certificates for those domains. We conclude from this that the drop-off in support of SCSV for domains that use HSTS ($SCSV|HSTS$) compared to the overall SCSV population stems from this large hosting provider who handles SCSV incorrectly and sets HSTS for a large population of likely unused domains.

Second, domains that use HPKP very frequently also use CT and HSTS headers. We expect users who successfully master the complicated HPKP setup to also deploy other techniques. One reason why CAA and TLSA usage remains relatively low among HPKP users, could be the required control over the domain's DNS server.

Third, use of CAA or TLSA is frequently combined, and often correlates to HSTS or HPKP deployment. Given the low dissemination of CAA and TLSA, it is not surprising to find its users be aware of other, more common security techniques.

5.1.6 Correlating Effort, Risk, and Usage

We relate deployment effort, risk to site availability, and measure deployment of technologies that have emerged after the DigiNotar incident in Table 5.4.

We classify effort as follows:

None where the server administrator has to take no action (e. g. SCSV, embedded SCTs).

Low applies where the operator has to enable an extension, but no complex configuration is needed. *E.g.*, this applies to HSTS, as one can just copy a configuration string from a tutorial.

Medium applies where the operator needs to make simple adjustments to instructions from manuals, e. g. replacing the domain name or CA name, e. g. for CAA.

High effort is assigned where configuration requires careful thought or multiple steps, e. g. for HPKP.

We classify availability risk as how easily misconfigurations can lead to serious availability issues for a domain. We introduce the following availability risk levels:

None for techniques where no risk exists.

Low is assigned to technologies that do not offer much risk potential or are easy and quick to fix.

Medium applies to mechanisms that are more difficult to fix when deployed incorrectly and can affect a large user base at once.

High is used for technologies that can harm availability for a large user base and that are difficult and slow to remediate. Examples are wrongly preloaded HPKP pins [248] and hostile pinning, where an MITM attacker sends incorrect pins to restrict user access [83].

Using this classification system, we clearly see in Table 5.4 that technologies with low effort and low risk to availability seem to hold a bigger market share.

5.1.7 Conclusion

A number of new security measures for the HTTPS ecosystem have been developed in the five years since the compromise of DigiNotar. While these techniques protect against a wealth of different attacks and would have been able to prevent or at least lessen the impact of the DigiNotar compromise, we find that deployment is disappointing for most of them. Our findings suggest a correlation between configuration effort, incurred risk to site availability, and actual deployment status. Technologies that are easy to deploy and have little risk to availability have the highest deployment (Certificate Transparency and SCSV). Those that have either high deployment effort or carry a high risk of misconfiguration have often low deployment.

In this dissertation we focused the analysis on the HTTP-based security techniques HSTS and HPKP and compare their deployment to other techniques. More details on other techniques can be found in the accompanying research paper [16].

Table 5.4: Correlation of age, # deploying domains, effort and availability risk of various HTTPS ecosystem security extensions, sorted by Top 10K domains. Low effort and low availability risk can drive wide-spread use.

Mechanism	Standard	Deployment		Effort	Availability
SCSV	2015	49.2M	6789	none	low
CT-x509	2013	7.0M	1788	none[2]	none
HSTS	2012	0.9M	349	low	low
CT-TLS	2013	27,759	171	high	none
HPKP	2015	6616	156	high	high
HPKP PL.	2012[1]	479	150	high	high
HSTS PL.	2012[1]	23,539	144	medium	medium
CAA	2013	3057	20	medium	low
TLSA	2012	973	3	high	medium
CT-OCSP	2013	191	0	low	none

1: Preloading list first added to Chrome in 2012

2: Requires deployment effort on CA side and a new site certificate.

5.2 Certificate Transparency

This section is based on the publications "In Log We Trust: Revealing Poor Security Practices with Certificate Transparency Logs and Internet Measurements" by Oliver Gasser, Benjamin Hof, Max Helm, Maciej Korczynski, Ralph Holz, and Georg Carle, which was published at PAM 2018 [101]; "The Rise of Certificate Transparency and Its Implications on the Internet Ecosystem" by Quirin Scheitle, Oliver Gasser, Theodor Nolte, Johanna Amann, Lexi Brent, Georg Carle, Ralph Holz, Thomas C. Schmidt, Matthias Wählisch, which was published at IMC 2018 [218]. In Section 5.4 details on the author's contributions and differences between the thesis text in relation to the published papers are given.

One of the Internet's most important protocols, Transport Layer Security (TLS), relies critically on server certificates being issued with diligence by the Web's trust anchors, the Certificate Authorities. It had long been suspected that this degree of trust may be misplaced [80], but from late 2008 on a string of security incidents relating to poor certification practices [203] culminated in the compromise of the DigiNotar Certificate Authority [195]. Being one of the affected parties and a major player on the WWW, Google began work in the IETF on Certificate Transparency (CT) as a response. While this technology is not designed to prevent actual attacks from happening, it can reduce the time to detection drastically. CT essentially turns the Web PKI inside out: a number of independent and neutral logs keep track of issued certificates. This enabled an unprecedented degree of transparency: both certificate misissuance and CA malpractice can now be detected by site operators and third parties. In the years since DigiNotar, Certificate Transparency has won widespread support. Browser vendors take incidents and malpractice seriously: a number of CAs have been called out for poor practices [174, 234], and the CA PROCERT has been removed from Mozilla's products due to violations of the industry's Baseline Requirements [165].

In this section, we carry out a thorough analysis of certificates stored in CT and assess CA compliance with the Baseline Requirements.

5.2.1 Background

In this section we provide information on protocols relevant for the CT study.

CA/Brower Forum In order to provide an industry standard for the behavior of CAs in the context of HTTPS, the CA/Browser Forum continuously negotiates technical policies for CA operations. Supplementing specifications such as RFC 5280 [56], it publishes the Baseline Requirements (BRs) [38].

Baseline Requirements The Baseline Requirements specify important properties for Internet security, for example which algorithms used in certificates are considered secure or what the maximum life time of a certificate may be.

Certificate Revocation Lists Certificate Revocation Lists (CRLs, see RFC 5280 [56]) provide a mechanism to withdraw trust from misissued certificates, e. g. in case of a key compromise.

Certificate Transparency Repeated misissuances of certificates have led to substantial scrutiny of CAs [53]. Certificate Transparency (CT, see RFC 6962 [156]) is a measure to monitor CA behavior. In CT, certificates are submitted into untrusted, public, append-only logs. The primary goal of CT is to allow site operators to observe which certificates were issued for their DNS names. To do this, they inspect the logs, retrieving and examining all certificates included in them. A secondary goal is improving compliance of CAs by easing discovery of misissuances.

On submission of a certificate, the log returns a signed inclusion promise called Signed Certificate Timestamp (SCT). Sites attach the SCT when presenting their certificate, notifying the browser of their participation in CT. Logs regularly produce signed commitments to a fixed entry list (Signed Tree Heads, STHs). A certificate is considered included in a log when it is covered by an STH.

At the time of the study, the Chrome browser required CT only for "Extended Validation" certificates. From April 2018 on, CT is required by Chrome for all newly issued certificates [225]. Public logs for this purpose are operated by Google and some certificate authorities.

A possible attack by a CT log server is presenting different views to different parties, also called equivocation. This can be addressed with gossip protocols, where participants inform others about the log view presented to them. One such proposal for CT exchanges SCTs and STHs via defined API endpoints on HTTPS servers [184]. The Chrome browser implements an alternative model, where STHs are transferred to the browser via the internal component updater [226]. Inclusion proofs are requested via a custom DNS-based protocol [157].

5.2.2 Related Work

The analysis of TLS certificates has become increasingly important, in particular with HTTPS becoming a *de facto* protocol for the Web and many of its APIs [89]. A number of analyses have been carried out, most commonly based on active scans and sometimes passive traffic observation. Our methodology relies to a large degree on a new, different data source, namely CT logs. In this section we list previously published related work and compare it to our CT ecosystem analysis as well as certificate analysis.

Several published works also exploit CT logs, albeit with different research questions. Amann *et al.* examine the use of Certificate Transparency in the context of general improvements to the TLS ecosystems since 2011, a year with a number of major CA incidents [16]. The authors' focus is on the deployment and practical use of these improvements. They do not investigate the properties of logged certificates. Aertsen *et al.* use CT logs to analyze the rise of the Let's Encrypt CA and the resulting more widespread use of encryption that enables smaller websites and hosting providers to acquire free certificates [5]. Gustafsson *et al.* use CT logs in combination with passive traffic monitoring to analyze the basic properties of logs and certificates, such as signature algorithm and key lengths of certificates [116]. They do not investigate violations of issuance standards. VanderSloot *et al.* combine CT logs with seven other certificate collection techniques to obtain a picture of the overall HTTPS ecosystem and how different data sources help to make it accurate [250]. They conclude that no collection method covers all certificates. However, they observe that CT logs in combination with active scans cover 98.5 % of their certificates. In our work we make use of this finding to also leverage CT logs and active scans. In parallel to our study, the performance impact of CT on HTTPS [185] and the deployment of sub-par certificates sourced from CT logs were analyzed [151].

A number of earlier publications investigates properties of certificates and TLS deployment. Holz *et al.* provides the first large-scale, long-term analysis of this kind [129]. Later, Durumeric *et al.* extend this approach to the entire IPv4 space [73]. The publications focus on basic properties of TLS certificates such as weak encryption keys,

Table 5.5: Overview of conducted measurements and used data sources.

Data source	Time period	# Entries	Size
CT log downloads	until Oct. 9, 2017	600 M entries	732 GB
Active HTTPS scans			
IPv4	Oct. 3–8, 2017	196.3 M hosts[1]	259.1 GB
IPv6	Oct. 1, 2017	8.8 M hosts[1]	73.0 GB
CRL downloads	Oct. 11, 2017	25.3 M entries	1.9 GB
Passive CT over DNS			
MWN UDP/53	Sep. 20–27, 2017	2.3 G pkts	10.5 TB
DNSDB TXT #1	Jul. 2016	36.4 M RRs	6.0 GB
DNSDB TXT #2	Sep. 20, 2017	2.4 M RRs	429.8 MB

1: unique IP–domain tuples, e. g. (216.58.207.142,google.com).

invalid path length constraints, invalid validity periods, and revoked certificates and sibling CA certificates. Chung *et al.* use TLS scans to analyze certificates without a valid root [51]. They show that invalid certificates make up the majority of collected certificates. Cangialosi *et al.* study private key sharing within the HTTPS ecosystem [42]. A large-scale study of HTTPS-induced browser errors was carried out by Acer *et al.* [1].

5.2.3 Methodology

In this section we present our methodology for conducting active and passive measurements. We use various different sources to get a large view of the certificate universe: We download certificates from CT logs, obtain certificates from active scans, retrieve CRLs, and conduct active and passive measurements to analyze CT gossiping deployment. Table 5.5 gives an overview of these sources, detailing the time of data collection, the number of entries, and the size of the acquired data. We also detail ethical and reproducibility considerations.

CT Log Downloads

We extend Google's CT tool to incrementally download certificates and their certificate chains from 30 CT logs. We publish our extended CT tool on GitHub [238]. In total we download 600 M log entries, resulting in 216.8 M unique certificates and 7.8 M unique certificates in chains.

Active HTTPS Measurements

To compare the certificates seen in CT logs to the actual HTTPS deployment we conduct active measurements over IPv4 and IPv6. First, we collect a total of 1.2 G domains from three different sources: TUM's hitlist [104], domains contained in CN and SAN of downloaded CT log certificates, and Farsight's DNSDB [87]. Second, we filter auto-generated disposable domains [46] from the DNSDB data by removing subdomains such as `netflixdnstest1.com` and domains with less than 100 queries within a month as indicated by DNSDB. Third, we resolve the remaining domains for A and AAAA records. Fourth, we conduct port scans on TCP/443 using ZMap [71] for IPv4, and our IPv6-enabled version [244] for IPv6. Fifth, we use our highly parallelized goscanner [241] to establish TLS connections to 191.4 M and 8.8 M IP address–domain name tuples for IPv4 and IPv6, respectively. To obtain the correct certificate we send the domain name in the SNI extension. Upon successful connection establishment we send HTTP requests to retrieve the server's HTTP headers and check for the presence of gossiping and pollination endpoints [184].

CRL Downloads

In order to determine the revocation status of certificates, we extract CRL URLs from certificates of active scans and CT logs. We then download these CRL files as well as Mozilla's OneCRL [170]. In total we extract 25.3 M entries from CRLs. We do not check OCSP as it is disabled in Chrome and previous work shows limited support [160].

Passive DNS Measurements

To analyze the use of Google's CT over DNS approach [157], we conduct passive measurements. We evaluate one week of DNS traffic at the Internet uplink of the Munich Scientific Network. Additionally, we use Farsight's DNSDB data [87] to further improve our client coverage.

Ethical Considerations

We follow an internal multi-party approval process before any measurement activities are carried out. This process incorporates the proposals of Partridge and Allman [186] as well as Dittrich *et al.* [67]. We assess whether our measurements can induce harm on individuals in different stakeholder groups. As we limit our query rate and use conforming HTTP requests, it is unlikely for our measurements to cause problems on scanned systems. Using the REST API provided by CT logs, we perform incremental downloads to reduce the impact on target systems. We follow best scanning practices such as maintaining a blacklist and using dedicated servers with informing rDNS names, web sites, and abuse contacts. We limit our passive measurements to DNS TXT records. The conclusion of this process is that it is ethical to conduct the measurements, but that we will only share data from our active measurements and not release passive data to protect the privacy of involved parties.

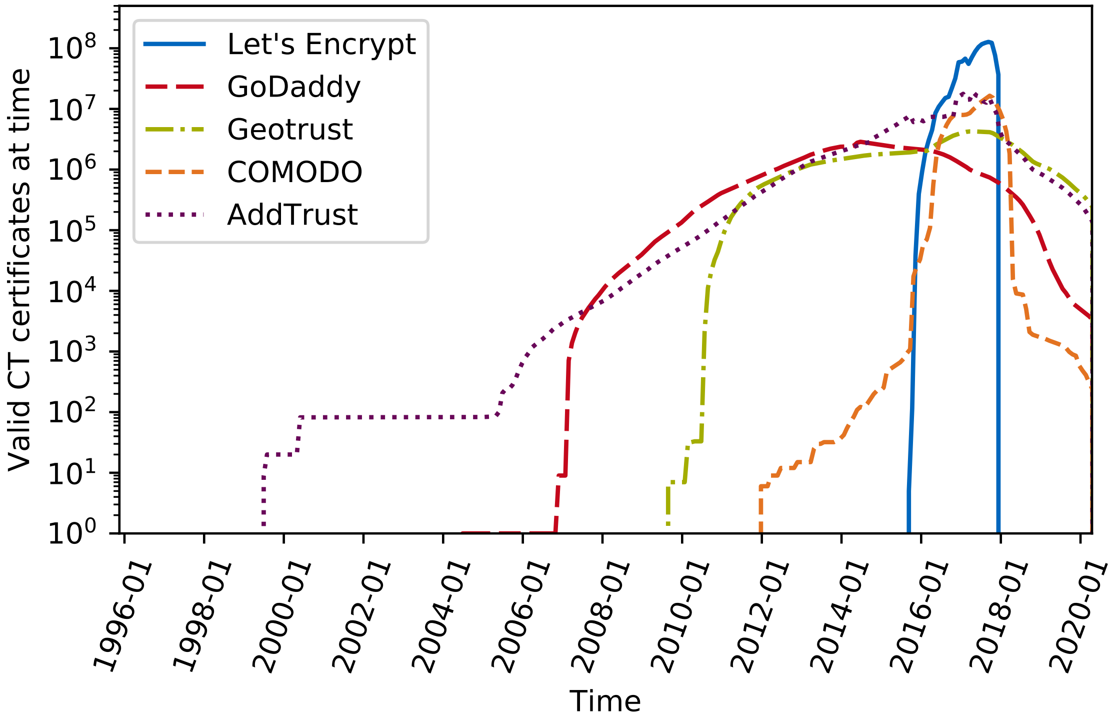


Figure 5.4: Non-expired certificates in CT logs by issuing CA. Y-axis is log-scaled.

Reproducible Research

To encourage reproducible research in network measurements [2, 219], we publish source code and data in the long-term availability archive of the TUM University Library: `https://mediatum.ub.tum.de/1422427`

5.2.4 Baseline Requirements

In this section we analyze the certificates found in CT logs, with a particular focus on their compliance with the Baseline Requirements.

Figure 5.4 shows the result of a quantitative analysis of non-expired certificates of the top 5 CAs over time. As is to be expected, the number of current, non-expired certificates peaks for most CAs around our cut-off date of October 9, 2017. One exception is GoDaddy, whose number of issued, non-expired certificates has been decreasing since 2014. We see that the vast majority of certificates in logs are issued by Let's Encrypt (LE), which saw exponential growth after the service became publicly available in 2016. Furthermore, due to the 3 month validity period of LE certificates, a sharp decline of certificates can be seen at the beginning of 2018. Due to longer validity periods, this decline is less pronounced for other CAs.

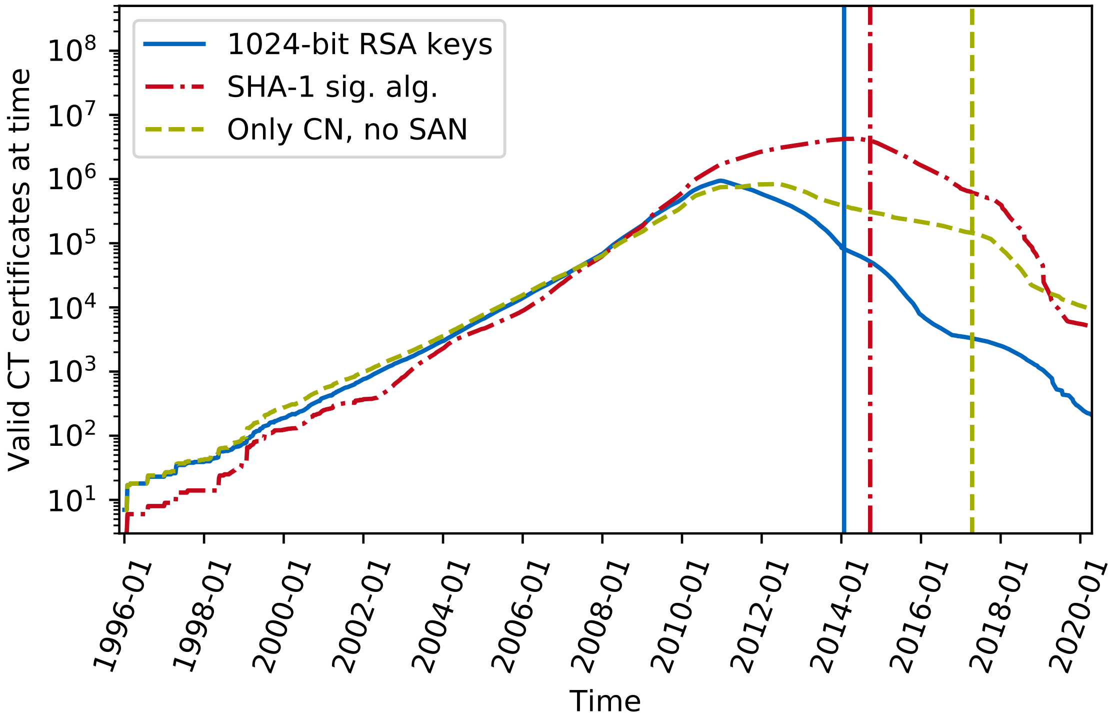


Figure 5.5: Non-expired certificates in violation of Baseline Requirements. Vertical line is Chrome enforcement date. Y-axis is log-scaled.

To evaluate the conformance of certificates to BRs, we run the cablint tool [237] on all non-expired certificates found in CT logs. We find 907 k certificates (1.3 %) in violation of BRs. Three major security relevant changes in the last years are shown in Figure 5.5, with vertical bars denoting deprecation steps by the Chrome browser. We observe that the prohibition of practices such as short keys is followed by a substantial reduction in the number of affected certificates. It takes years, however, until all old non-compliant certificates are expired.

Next, we investigate violations of requirements or recommendations in the current BRs. We categorize these violations as pertaining to the *identity* (e. g. SAN or CN), *signature* (e. g. hash algorithm), *key* (e. g. key usage or size), or *validity time*. Grouping certificates by year of issuance, Figure 5.6 shows the proportion of certificates exhibiting errors in these categories. This allows us to see the proportion of problematic certificates independent of the issuance rate. Generally, the proportion of certificates with errors is declining over time, with identity and key issues being predominant. In 2017, signature related issues become the prevalent cause of errors.

Attributing these violations to specific CAs, we select the 5 CAs with the highest number of infringing certificates. We show the number of violating certificates in the different

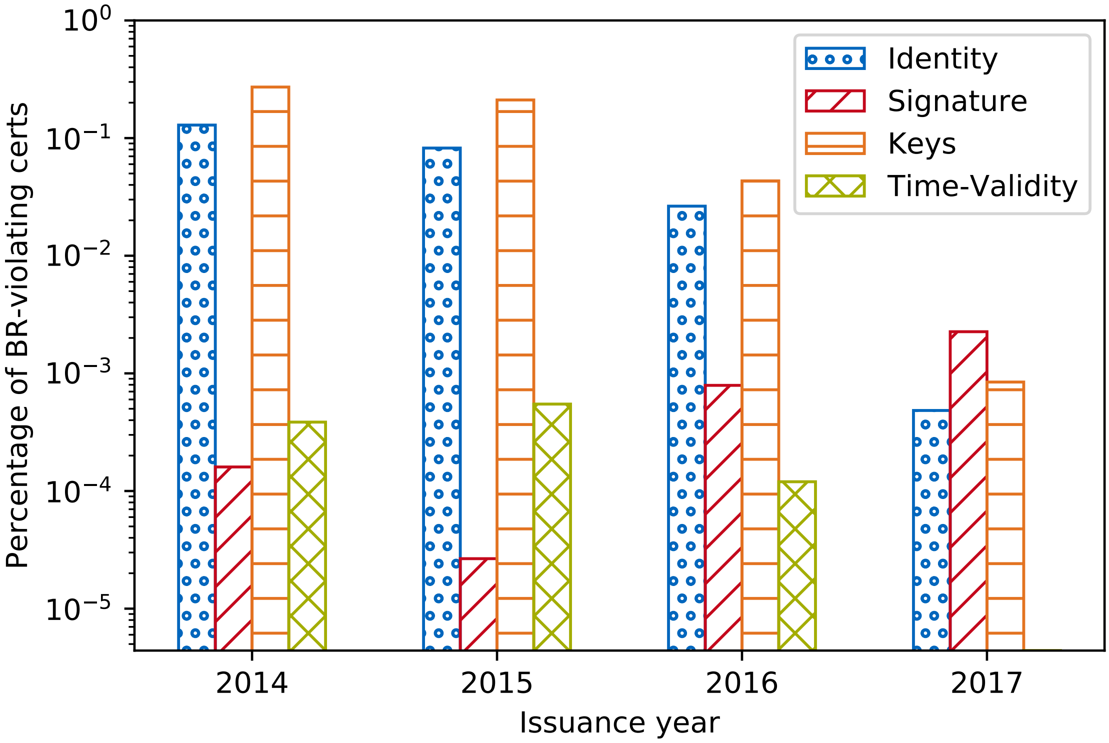


Figure 5.6: Proportion of certificates in violation of Baseline Requirements per violation category over year of issuance. Y-axis is log-scaled.

categories per CA relative to their total issued certificates in Figure 5.7. The most significant infractions are SHA1 signatures by CloudFlare and use of non-critical key usage extensions by WoSign. Upon closer investigation we find that most certificates with BR violations are signed by revoked intermediate certificates. We use our measurement results to improve issuance practices by notifying affected CAs. Furthermore, we note that Let's Encrypt has never committed any BR violations, while issuing the most certificates. Their service therefore improves Internet security not only by democratizing encryption [5], but by doing so in exemplary accordance with best practices.

5.2.5 Comparing CT Log Data to Active Scans

In this section we evaluate the differences between certificates in CT logs and those obtained from active scans dating back until 2009. Additionally, we take a first look at the deployment of CT-specific HTTP headers and determine the value of CT logs to create IP address hitlists.

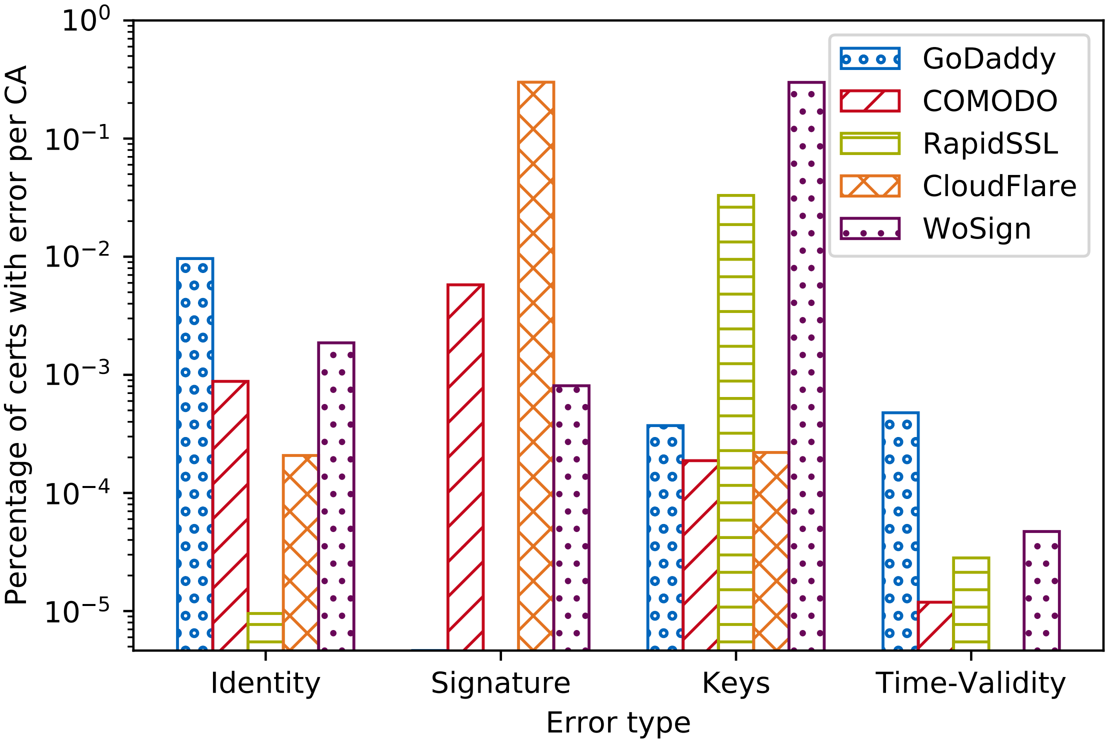


Figure 5.7: Proportion of certificates in violation of Baseline Requirements per CA over violation category. Y-axis is log-scaled.

Table 5.6: Comparison of certificates found in CT logs and active scans.

Cert source	Total	Not revoked	Not expired	Not self-signed	Browser-valid	BR-valid
CT logs	216.8 M	216.6 M	70.2 M	216.8 M	70.2 M	206.3 M
Active scans	128.1 M	127.5 M	118.8 M	109.4 M	74.8 M	115.4 M

Certificate Deployment and Validity

In our active scans we collect 316.3 M certificates (32.8 M unique) from 128.3 M successful handshakes with IPv4 hosts and 4.2 M IPv6 hosts. When the same certificate is presented for a name under all its IP addresses, within and across IP versions, we call the domain *consistent*. The vast majority of domains (e. g. 99 % for IPv6) delivers consistent certificate chains. We investigate inconsistent domains and find that these are mostly due to TLS services offered by Content Distribution Networks (CDNs): 86.9 % of IPv6 inconsistencies can be attributed to CloudFlare, 5.4 % to Akamai. Inconsistent chains use the same certificate key and *Common Name* in about 80 % of the cases. *Subject Alternative Name* entries, however, are deviating to a large extent. We conclude that inconsistent certificate chains are mostly due to CDNs dynamically adding client

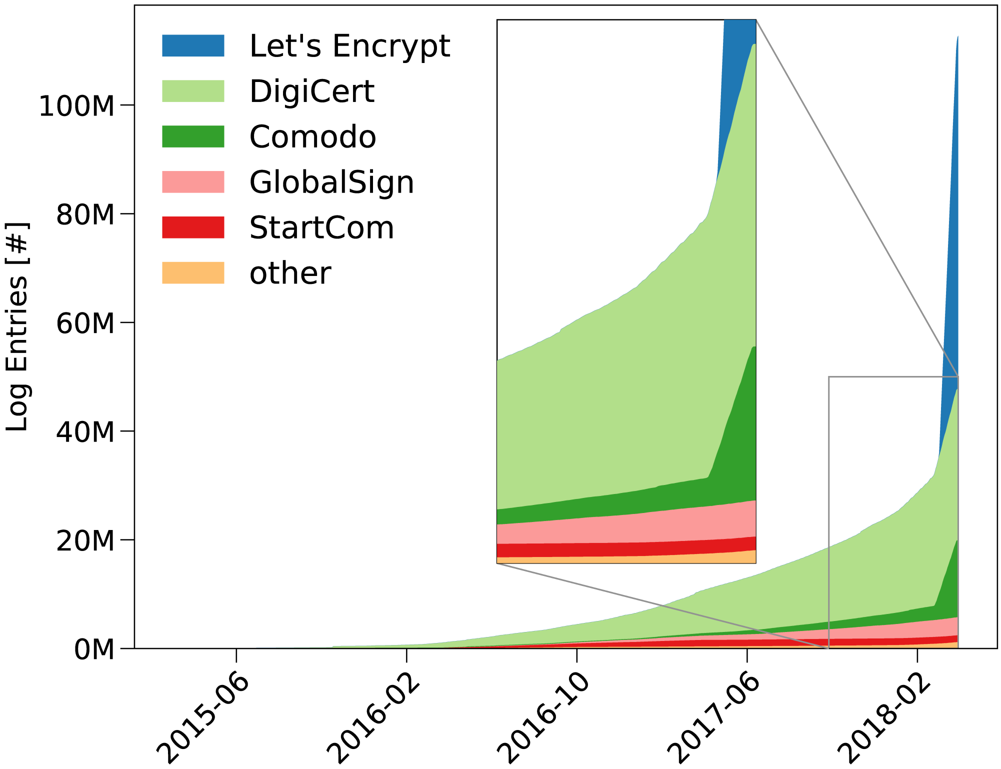


Figure 5.8: Runup of precertificates over time by CA, zoomed in close to Chrome's April 2018 requirement of all newly issued certificates being logged [225]. Created by Theodor Nolte [218].

domains to certificates. In the following we limit our analysis to the 128.1 M consistent domains in order to make quantitative statements more intuitively understandable.

We analyze the overlap of certificates in CT logs and certificates obtained from active scans and find that 109.8 M (85.7 %) certificates from active scans are logged in CT. This high percentage is an encouraging milestone towards the goal of logging all deployed certificates. Since April 2018, the Chrome browser only accepts newly issued certificates which provide valid SCTs to prove their inclusion in Certificate Transparency logs [225].

Figure 5.8 shows the cumulative increase of precertificates in CT logs per CA. It confirms our finding that the volume of logged certificates strongly increases shortly before the April 2018 Chrome deadline. The majority of logged certificates is issued by three CAs: Let' Encrypt, DigiCert, and Comodo. In-depth analysis regarding the rise of CT and its implications on the Internet as an ecosystem can be found in the respective publication [218].

In Table 5.6 we distinguish certificates by revocation, expiration, self-signed, browser-valid status, as well as conformance with the Baseline Requirements.

For CT log and active scan certificates, we find low numbers of certificates revoked through embedded CRLs or OneCRL [170].

More than 92 % of certificates found in active scans are not expired. In CT logs, however, more than two thirds of certificates are expired. This is to be expected, since CT logs explicitly keep expired certificates. This feature allows to easily evaluate trends in the certificate ecosystem over time.

The picture changes when evaluating self-signed certificates: CT logs only accept certificates valid under root stores and therefore do not contain self-signed server certificates. In active scans we find 14.6 % self-signed certificates, which is a decrease compared to previous studies [51, 73]. This could be an indicator of Let's Encrypt's democratizing impact [5], where the lower end of the market moves from self-signed to free CA-signed certificates.

Next, we analyze whether certificates are accepted by web browsers. These are a subset of certificates which are neither revoked nor expired nor self-signed. Additional conditions (e. g. matching domain, correct chain to root cert) must be met as well. Since CT logs only accept root store-anchored certificates, all valid CT log certificates are accepted by browsers. However, only 63 % of not expired certificates from active scans are browser-valid. Therefore a non-negligible number of certificates found in the wild is resulting in security warnings to users.

Moreover, we compare BR violations of certificates found in CT logs and found using active scans. 95.2 % of logged certificates are valid according to the BRs, compared to 90.1 % of deployed certificates. This finding underlines the importance of logging all certificates in order to make violations more easily traceable and CAs more accountable.

Furthermore, we assess the impact of the impending distrust of Symantec root certificates [173]. We find 4.2 M domains where one of the Symantec root certificates is used. Limiting our analysis to specific certificate validity periods allows us to quantify the impact more precisely: 1.9 M domains will not be trusted anymore in May 2018, whereas 777.7 k domains will be affected by the complete removal of Symantec root certificates in October 2018. These findings show that many domains have not yet switched to other CAs and stress the importance of a smooth transition to the new Symantec CA owner DigiCert.

CT-Specific HTTP Headers

Similarly to enforcing HTTPS-only connections using the HTTP Strict Transport Security (HSTS) header (see RFC 6797 [127]), web servers can require the presence of certificates in CT logs. Requiring the presence in logs allows to detect man-in-the-middle attacks where the original server certificate is replaced by an attacker. We analyze the deployment of the unofficial RequireCT [204] and the draft RFC Expect-CT [230] headers.

We find eight domains sending HSTS headers with a RequireCT directive and 7.3 k domains with Expect-CT headers. In the following, we investigate the Expect-CT deployment. This header consists of a mandatory *max-age* field and optional *enforce* and *report-uri* fields. We find 12.1 % of domains to omit the mandatory *max-age* directive. The majority of domains sets the *max-age* to zero, effectively disabling the Expect-CT mechanism. Only 29.9 % of domains *enforce* Expect-CT, the majority makes only use of the reporting feature. With 608 domains, less than 10 % enforce Expect-CT with a duration of one day or more.

We check whether domains which send an Expect-CT header have in fact logged their certificate in CT. The majority of certificates can be found in CT logs. However, 83 Expect-CT domains (1.2 %) do not send certificates which are logged. 48 of these enforce Expect-CT with a *max-age* greater than zero. These domains do not comply with the Expect-CT specification. We find a lower misconfiguration percentage in Expect-CT compared to the more established yet complex public key pinning via HPKP headers [16].

CT Logs as a Source for IP Address Hitlists

CT logs contain not only valuable information about certificates, but are also an additional source of domain names. We analyze the value of domain names extracted from CN and SAN of logged certificates by comparing them to our publicly available hitlist [104]. TUM's hitlist provides IP addresses based on domains from zonefiles, Alexa Top 1M, Cisco Umbrella, CAIDA, and Rapid7.

The CT log data adds 82.2 M domains, 5.4 M IPv4, and 489 k IPv6 addresses to the hitlist. This corresponds to respective increases of 50.5 %, 56.2 %, and 69.6 %. Especially the large increase of IPv6 addresses can aid future measurement studies. We make the hitlist enhanced with CT domain data freely available [242].

5.2.6 Gossiping and Inclusion Proofs

CT offers gossiping protocols to detect equivocation attacks, where a log presents different views to different parties. Gossiping allows clients to exchange their log view with each other. Clients can also request inclusion proofs from the log, demonstrating that a specific certificate was indeed incorporated by the log. We conduct active and passive measurements to evaluate if these techniques are used.

As part of our active scans, we send HTTP requests to responding domains in order to evaluate the deployment of CT gossiping endpoints among HTTPS websites. These requests are targeted at specific URL paths used in CT gossiping [184]. Additionally, we send one request to a non-existent path that serves as the baseline of how web servers answer requests for non-existent paths.

In the course of these measurements, we receive answers from 109.2 M domains and inspect the HTTP return codes. We remove hosts that answer with 2xx or 3xx to the non-existent baseline path, send the same answer for CT paths as the baseline request, or answer with 4xx to the CT paths. After this filtering 16.8 k (0.015 %) domains remain. This is an upper bound of domains supporting HTTP-based CT gossiping, as web servers might be configured in a way which triggers different behavior for CT and the baseline path. To lower this upper bound, more complex measurements would need to be performed. These low numbers, however, suggest that HTTP-based gossiping is not widespread.

The gossip requests generated a magnitude more abuse notifications compared to other scans. This should be considered in the protocol specification, e. g. by using an HTTP header as a discovery mechanism less prone to undue excitement. Alternatively, browsers could gradually acclimate operators to this new reality.

In addition to active HTTPS scans, we conduct passive DNS measurements as described in Section 5.2.3. Since HTTPS URL paths are encrypted in TLS and therefore not visible, we instead evaluate the deployment of Google's proposal to fetch inclusion proofs over DNS [157]. Even though the CT over DNS proposal is implemented in Google's Chrome browser [50], we could not find any TXT record matching the document specification in our passive data. This was confirmed by Google, who said they never activated the protocol due to privacy concerns [167].

We conclude that protection against split-view attacks by logs which is an architectural necessity in CT has next to no deployment in the wild.

5.2.7 Conclusion

In this study we investigated the Baseline Requirements adherence of certificates found in CT logs and through active scans. We mapped these violations to issuing CAs and inform them of our findings. Furthermore, we compared the results from CT logs and active scans, finding that logged certificates exhibit less violations. Additionally, we observed that CT gossiping, although required in the security model of CT, does currently not have any substantial deployment.

5.3 Key Contributions of This Chapter

This chapter addressed research question RQ III which includes tackling challenges C 7 and C 8. We analyzed HTTPS security extensions, focusing our efforts on HTTP-based header extensions and comparing our analysis to other techniques. Moreover, we evaluated Certificate Transparency (CT) and investigated the quality and security of certificates found in CT logs.

In the following we list the key contributions of this chapter:

HTTPS security extensions We performed an in-depth analysis of HTTP header-based security extensions, namely the HSTS and HPKP headers. We found varying deployment—HSTS sees significant deployment, whereas HPKP has very low deployment—and compared it to other HTTPS security extensions. As part of our analysis we correlated effort and risk with deployment and found that techniques with low effort and low risk saw higher deployment (e. g. HSTS) compared to high effort and risk techniques (e. g. HPKP). With these findings we tackled challenge C 7.

Certificate Transparency We analyzed both the Certificate Transparency (CT) landscape as well as certificates logged in CT logs. The CT landscape saw a large increase in participation shortly before the April 2018 deadline imposed by Google's Chrome browser. Additionally, we found that a part of certificates in CT logs exhibit severe weaknesses (e. g. short keys and insecure signature algorithms). On the bright side it became clear that imposing stricter rules can improve the situation, which was also the case for the quality and security of certificates in CT logs. This analysis allowed us to tackle challenge C 8.

After tackling these two challenges we can provide an answer to RQ III, as we find that there are still significant numbers of HTTPS servers with lacking HTTPS security extensions and weak certificates which makes them vulnerable to Man-in-the-Middle attacks.

5.4 Statement on Author's Contributions

This chapter is partially based on the publications "Mission Accomplished? HTTPS Security after DigiNotar" by Johanna Amann, Oliver Gasser, Quirin Scheitle, Lexi Brent, Georg Carle, and Ralph Holz, which was published at IMC 2017 [16]; "In Log We Trust: Revealing Poor Security Practices with Certificate Transparency Logs and Internet Measurements" by Oliver Gasser, Benjamin Hof, Max Helm, Maciej Korczynski, Ralph Holz, and Georg Carle, which was published at PAM 2018 [101]; “The Rise of Certificate Transparency and Its Implications on the Internet Ecosystem” by Quirin Scheitle, Oliver Gasser, Theodor Nolte, Johanna Amann, Lexi Brent, Georg Carle, Ralph Holz, Thomas C. Schmidt, Matthias Wählisch, which was published at IMC 2018 [218].

The author made the following contributions to the HTTPS security extension study presented in the IMC 2017 paper [16] and the adapted sections in this chapter. The author contributed to the design and execution of the active measurements. The author designed and implemented goscanner, the tool used in this study. Moreover, the author contributed significantly to the analysis of the HTTP header-based extensions. Finally, the author contributed to the discussion and correlation analysis.

In the following we describe changes between Section 5.1 and the HTTPS security extension study presented in the IMC 2017 paper [16]. In this dissertation we focus on the active measurements—specifically HSTS and HPKP measurements—as the chapter’s focus lies on HTTPS security. Therefore, we omit most of the passive measurements except their results in the correlation analysis.

The author made the following contributions to the Certificate Transparency (CT) log study presented in the PAM 2018 paper [101] and the adapted sections in this chapter. The CT log analysis work was first started as part of Max Helm’s interdisciplinary project (“Evaluating TLS Certificate Transparency Logs using Active Scans”) [120], which the author advised. The author coordinated and lead the group of researchers in this study. The author provided significant contributions in the study’s design and execution. Moreover, the author contributed in the log downloader design and in the certificate analysis. The author contributed significantly in the active measurements and their result analyses. Finally, the author also provided minor contributions to the comparison to legacy and non-HTTPS certificates.

In the following we describe changes between Section 5.2 and the CT log study presented in the PAM 2018 paper [101]. Most of the analyses in the paper are present in this dissertation. We omit the comparison to legacy and non-HTTPS certificates as it was not the focus of our evaluation in this thesis.

The author made the following contributions to the Certificate Transparency (CT) log study presented in the IMC 2018 paper [218] and the adapted sections in this chapter. The author contributed to the design of the study and its execution. The author provided the CT log data used in the analysis of CT growth as well as the active HTTPS measurement results.

5.4 Statement on Author's Contributions

In the following we describe changes between Section 5.2 and the CT log study presented in the IMC 2018 paper [218]. We only use the developments over time in CT logs in this dissertation to highlight newest developments around the April 2018 Chrome enforcement deadline. The rest of the paper is omitted in this dissertation.

Chapter 6

BACnet

This chapter is based on the publications "The Amplification Threat Posed by Publicly Reachable BACnet Devices" by Oliver Gasser, Quirin Scheitle, Benedikt Rudolph, Carl Denis, Nadja Schricker, and Georg Carle, which was published in JCSM 2017 [106]; "Security Implications of Publicly Reachable Building Automation Systems" by Oliver Gasser, Quirin Scheitle, Carl Denis, Nadja Schricker, and Georg Carle, which was published at WTMC 2017 [105]; "Öffentlich erreichbare Gebäudeautomatisierung: Amplification-Anfälligkeit von BACnet und Deployment-Analyse im Internet und DFN" by Oliver Gasser, Quirin Scheitle, Carl Denis, Nadja Schricker, and Georg Carle published at DFN-Konferenz Sicherheit in vernetzten Systemen 2017 [107]. In Section 6.12 details on the author's contributions and differences between the thesis text in relation to the published papers are given.

In a connected world Internet security is becoming increasingly important. Attacks, which are frequently executed by botnets, can impact people in their everyday life. A ubiquitous kind of attack is the amplification attack, a special type of Denial-of-Service attack. Several protocols such as DNS, NTP, and SNMP are known to be vulnerable to amplification attacks when security practices are not followed.

In this research we evaluate the vulnerability of BACnet, a building automation and control protocol, to amplification attacks. To assess BACnet's vulnerability we conduct active traffic measurements on an Internet-wide scale. We find 16 485 BACnet devices, the largest number to date. Additionally, more than 14 k of these devices can be misused as amplifiers, with some generating amplification factors up to 120. To remediate this potential threat we employ a vulnerability notification campaign in close coordination with a CERT. We assess the success of the campaign and find that the number of publicly reachable BACnet devices decreased only slightly. Additionally, we employ passive measurements to attribute the majority of BACnet traffic in the wild to scanning projects. Finally, we also give suggestions to thwart the amplification attack potential of BACnet.

This chapter is structured as follows: First, in Section 6.1 we introduce the problem statement and motivate our research. Then, we briefly describe the BACnet protocol and our choice of scanning payload in Section 6.2. We continue with our scanning methodology and ethical considerations in Section 6.3. Section 6.4 details the BACnet deployment evaluation based on our scan results. In Section 6.5 we analyze in detail how BACnet devices can be used for amplification attacks. Section 6.6 investigates BACnet traffic seen in the wild using passive traffic measurements. In Section 6.7 we detail our CERT-backed notification campaign and assess its success. Additional efforts to remediate the threat posed by publicly accessible BACnet devices are discussed in Section 6.8 and related work is presented in Section 6.9. Sections 6.10 to 6.12 conclude this chapter with a summary, listing of key contributions, and a statement on the author's contributions.

6.1 Introduction

In the last years the number of Denial-of-Service attacks increased dramatically in both frequency and data rate. These attacks more and more misuse Internet of Things (IoT) devices or embedded systems. Many of these devices are not properly secured and are therefore an ideal target for misuse and attacks. They can be used directly by being part of a botnet, or indirectly as a reflector or amplifier. An example for direct abuse is the Mirai botnet which attacked the Internet infrastructure company Dyn causing partial outages for Twitter, Amazon, and Netflix [149], and started a DDoS attack which Akamai was unable to mitigate [150]. An example of indirect abuse is the use of open DNS resolvers as amplifiers in the attack on Spamhaus [194].

These examples highlight problems arising from two sources: First, IoT devices without proper security posture may be taken over for arbitrary abuse. Second, embedded devices may offer insecure and easy to abuse services such as open DNS resolvers and misconfigured NTP servers. Most of these security problems, however, are only discovered when their exploitation causes fallout. It is therefore crucial to identify potentially insecure devices before they are being misused in attacks. We focus our measurements on the building automation protocol BACnet [19] and assess its vulnerability to amplification attacks. BACnet is capable of connecting a wide range of devices and offers remote monitoring and control features. Security was not a priority when the BACnet protocol was designed and the recommendation [181] is to never connect BACnet devices to the Internet, but always place them in a segmented, separate network. We investigate whether this recommendation is followed by probing for BACnet devices which are reachable in the public Internet.

6.2 The BACnet Protocol

This section provides an overview of the BACnet protocol, highlighting aspects important for this research.

The development of the BACnet protocol started in 1987 [181], with the first release of its specification in 1995 by ASHRAE [19]. BACnet was designed as a standalone network protocol, including its own network layer with 16-bit network and device identifiers. BACnet's dedicated network layer implied segmented networks, hence security was not a consideration in protocol design. In 1999, BACnet/IP was defined to use IP as the network layer, which comes with many security implications. Security advice for BACnet/IP to date is to segment BACnet networks.

BACnet/IP uses a rather complex packet structure with multiple internal header layers. In its design, BACnet properties somewhat resemble SNMP MIBs.

6.2.1 BACnet Payload

For our measurements we use the generic wild-card device ID `0x3fffff` and select the following suitable payloads to identify BACnet devices:

IPv4 We conduct IPv4 measurements using a *ReadPropertyMultiple* request payload. This type of request allows to specify a list of BACnet property IDs (e. g. `0x46` = model name, `0x79` = vendor name). The queried BACnet device returns a list of corresponding property values (e. g. model name: Niagara AX, vendor name: Tridium Inc.).

IPv6 IPv6 support for BACnet was added in 2016 [20]. The standard defines new header types for IPv6, requiring a different payload to identify IPv6-capable devices. We use a *VirtualAddressResolution* request to discover BACnet devices over IPv6. IPv6-capable BACnet devices return the remote virtual address which is needed in the subsequent *ReadPropertyMultiple* request.

Amplification The payload in our amplification scans is made up of the regular payload amended with additional properties which promise a high amplification factor, such as *PropertyList*.

6.3 Methodology

This section describes our methodology by giving details on our active scans, the processing of answers, and ethical considerations guiding our research.

6.3.1 Scan Overview

BACnet is run on UDP ports 47808 – 47823 by default [19]. Using different strategies, we probe those ports via IPv4 and IPv6. We verify responses for valid payloads to filter for actual BACnet devices. Using a different scanning payload, we then further survey these BACnet devices to determine their vulnerability for amplification attacks. Depending on the number of targets, we optimize packet sending rate to (1) minimize network load and (2) achieve tractable scanning duration. Table 6.1 gives an overview of the scan types, listing the number of conducted scans, the number of scanned ports, the used packet rate, the scan duration, the number of targets, received responses ("Resp."), and parsable BACnet payloads ("BACnet").

6.3.2 Internet-wide IPv4 Scans

We probe the IPv4 address space on the previously mentioned UDP ports using ZMap [71] and a BACnet UDP payload. We exclude IP addresses that are (1) on our blacklist, or (2) part of the IANA reserved ranges [133], or (3) not routed according to BGP data from our routers.

For the IPv4 scans we choose a rate of 25 kpps, resulting in a duration of 41 hours for each performed scan. The scans are run from four measurement machines located in a dedicated measurement network.

We conduct four IPv4-wide scans: The first scan was executed in December 2016 and is used to evaluate the BACnet deployment (see Section 6.4). The three subsequent IPv4-wide scans are conducted in February, July, and August of 2017. We use those to assess the success of the notification campaign (see Section 6.7).

We use the same filtering process to identify valid responses for all scans. In the following we describe this process and show breakdown numbers from the December 2016 scan.

In a first step, we filter the raw ZMap results for packets with the queried source port. We discard about 20 % of mismatching responses, which stem from source ports such as UDP/53 (DNS) or UDP/39999 (unregistered, but linked to Sygate [55]). These responses might be counter-scans from infected or malicious devices, probing our IP address for vulnerabilities. After this filtering step, we count responses from 32 k unique IPv4 addresses. From the scan on port 47808 we get about 17 k (53 %) responses, from

Table 6.1: Overview of all BACnet scans.

Type of scan	Scans	Ports	Rate	Duration	Targets	Resp.	BACnet
IPv4-wide	4	16	25 kpps	41 h	2.4 G	32 868	16 485
IPv6 hitlist	1	1	5 kpps	2 min	407 k	0	0
Amplification	1	16	100 pps	3 min	16 k	15 598	15 429

port 47809 about 3 k (9 %), and from port 47810 about 1.1 k (3 %), and ports 47811 – 47823 hold about equal shares of the remaining 35 %. This result shows that not all BACnet devices run on the same port and it supports our decision to scan for all 16 official BACnet ports to obtain a complete picture of the BACnet deployment. Scanning only the most prominent port UDP/47808 as e. g. done by Mirian *et al.* [168] misses about 47 % of publicly reachable BACnet IP address – port combinations.

In a second step, we filter the responses for valid BACnet payloads. We use our tailor-made Python BACnet module which we publish on GitHub [97]. We filter for compliance with the following characteristics, which are required for a genuine response to our packet: The transport type is BACnet/IP (`0x81`), the payload is an original unicast NPDU (`0x0a`), the BACnet version is the only valid version 1 (`0x01`), no reserved NPDU control bit is set, and the application payload type is BACnet-ComplexACK-PDU (`0x03`). After the filtering phase 16 485 (of initially 32 868) valid BACnet responses with payload content remain. Spot-checks on non-compliant packets reveal payloads that are e. g. invalid, mirrored, randomized, or associated to other protocols. These might stem from honeypots, BACnet simulators, or unusual device configurations. Further investigation of these devices would require more intrusive scanning.

By scanning all standardized BACnet ports we also obtain more valid BACnet payloads: Our 16.4 k valid responses exceed Mirian *et al.* 's 12.8 k “valid handshakes” [168].

The distribution of ports after this filtering is more centric towards port UDP/47808 (84.4 % of responses). We evaluate the responses from all four scans in detail in Sections 6.4 and 6.7.

6.3.3 IPv6 Scans

As IPv6 support for BACnet was added in early 2016 [20], we scan for BACnet devices in the IPv6 space.

Since scanning the full address space is not feasible in IPv6, we follow the domain-resolution approach of our IPv6 hitlist [104] (see also Chapter 4). We also gain IPv6 addresses from responsive IPv4 BACnet devices by querying their rDNS record for `AAAA` records. We query 407 k unique IPv6 addresses, but do not receive any reply. This is likely due to a lack of IPv6 support in the field. As BACnet simulators do not support IPv6 yet, we can not validate our payload, which we thoroughly check against the BACnet standard.

6.3.4 Amplification Scans

Based on the subset of responsive BACnet devices, we conduct additional scans to evaluate the amplification potential of those devices. Compared to previous scans we now request additional BACnet properties. Since these scans might produce more load on target systems we reduce the scanning rate to 100 packets per second. We apply the same filtering steps as for the IPv4-wide scans. This removes about 170 responses from non-scanned IP addresses.

6.3.5 Ethical Considerations

We follow an internal multi-party approval process before any measurement activities are carried out. This approval process incorporates the proposals of Partridge and Allman [186] as well as Dittrich *et al.* [67]. We assess whether our measurements can induce harm on individuals in different stakeholder groups. As we use a valid payload in accordance to the BACnet standard, it is unlikely for our scans to cause problems on scanned devices. We minimize interference of our scans by following best scanning practices such as maintaining a blacklist and using dedicated servers with informing rDNS names, web sites, and abuse contacts. We consider that publication of IP addresses of possibly vulnerable and amplifying devices may be abused by third parties. The conclusion of this process is that it is ethical to conduct the experiment, but that we will, in contrast to our usual policy, not share data from this work with the public. Instead, we will only make the data available upon request to other researchers for reproducibility and comparison, and to the DFN-CERT for vulnerability notification of affected parties. During our scans we did not receive any complaints.

6.4 BACnet Deployment

In this section we evaluate the BACnet deployment by analyzing the responses obtained from our December 2016 scans.

6.4.1 Vendor Analysis

We find devices from a total of 97 different vendors, with just the top 3 vendors representing 52 % of all devices. Table 6.2 shows the five most frequent vendors found in our scans. When comparing our results to related work, we see that Mirian *et al.* [168]

Table 6.2: Top 5 BACnet vendors in results.

Pos.	Vendor ID	Vendor Name	Count	%
1	35	Reliable Controls Corporation	3740	24.8
2	36	Tridium Inc.	2079	13.8
3	8	Delta Controls	2004	13.3
4	5	Johnson Controls Inc.	1328	8.8
5	24	Automated Logic Corporation	1051	7.0

Table 6.3: Top 5 ASes by count of BACnet devices.

Pos.	ASN	Organization	Count	%
1	7018	AT&T Services, Inc.	1510	9.2
2	7922	Comcast Cable Communications, Inc.	1450	8.8
3	22394	Cellco Partnership DBA Verizon Wireless	774	4.7
4	852	TELUS Communications Inc.	697	4.3
5	6327	Shaw Communications Inc.	454	2.8

also find Reliable Controls (12.7 %) and Tridium (10.6 %) as their top BACnet vendors, however the share of these vendors in our evaluation is larger.

6.4.2 Topological Clustering

We next investigate the distribution of BACnet devices over Autonomous Systems (ASes) and announced BGP prefixes. We use CAIDA's routeviews data [39] to map IP addresses.

We find AS coverage rather sparse, with BACnet devices present in 1439 ASes, with a median of 2 devices per AS. This is a small share of the 55 738 total ASes as of 2016 [39].

We also find our number of 1439 ASes to be in line with Mirian *et al.* , who discover BACnet devices in 1330 ASes.

The BACnet devices from our scans cover 5109 announced BGP prefixes, of which 3021 only contain 1 device. The top 5 prefixes are /16 or larger prefixes belonging to ASes of major Internet service providers listed in Table 6.3.

6.4.3 Geographical Clustering

We also map the IP addresses of BACnet devices to countries using the IP2Location database [140]. While research has shown that IP geolocation databases can introduce significant biases [190], we believe them still to be indicative of the top countries of deployment. We find BACnet devices to be very centrally clustered with 60 % in the US and 20 % in Canada. With significantly less devices, Australia (3 %), France (2 %) and Spain (2 %) follow.

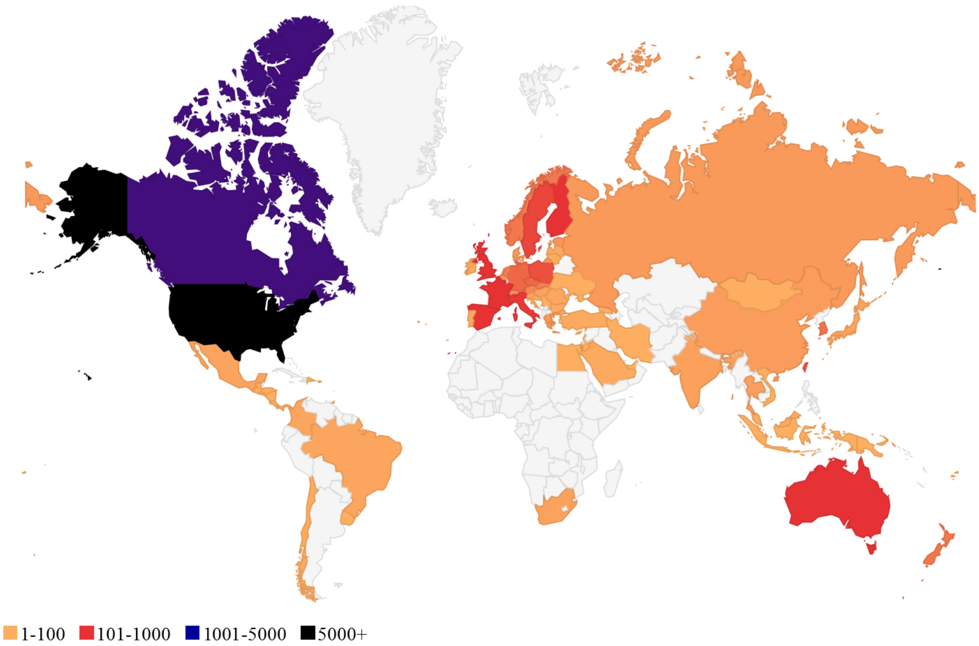


Figure 6.1: Geographical distribution of found BACnet devices. Created using Google Charts by Nadja Schricker [222].

In Figure 6.1 we visualize the country distribution on a world map, with countries clustered in majority of devices (5000+), many devices (1001 – 5000), some devices (101 – 1000), and few BACnet devices (1 – 100). For countries depicted in white we found no publicly reachable BACnet devices.

6.5 Amplification Attacks using BACnet

This section describes BACnet's vulnerability to amplification attacks. We evaluate the number of available amplifiers as well as the bandwidth amplification factor (BAF) of BACnet. BACnet supports both single property and multiple property requests. As the names already suggest, single property requests query one property (e. g. model name), whereas a multiple property request queries a list of properties from a BACnet device (e. g. model name, vendor name, firmware version). To assess the amplification potential of BACnet devices we scan with a generic multiple property payload. From this, we derive (1) the empirical BAF for our generic payload, (2) the calculated BAF for individual properties in a single property request, and (3) the calculated BAF for individual properties when repeatedly requesting the specific property in a multiple property request.

6.5.1 Amplification Attack Characteristics

An amplification attack is a type of Denial-of-Service attack where (1) the response payload is larger than the request payload. This ratio is called the bandwidth amplification factor (BAF) [207]. In addition to a BAF >1, there are two other typical characteristics for amplification attacks: (2) the used protocol is stateless and (3) no authentication is required.

BACnet/IP is a UDP-based protocol and does not require any handshake. This stateless property already satisfies characteristics (2) and (3). Since we are free to choose the requested property which the BACnet device will then send an answer to (provided the device supports the property), we can select properties which will most likely trigger a large response by the queried device. In the following we evaluate which properties provide us with a large BAF. If such a property is found, characteristic (1) is satisfied and BACnet can be used in amplification attacks.

6.5.2 Number of BACnet Amplifiers

We find 15 429 responsive BACnet devices with our amplification scans on ports 47808 – 47823. If a device does not support a requested property, it will reply with a four byte error, resulting in a property BAF<1. We quantify the amplification attack threat per BACnet property and device using error-free responses only. We focus the amplification attack analysis on variable length properties (i. e., strings or arrays) as these are more likely to give a larger BAF.

As shown in Table 6.4 we see stark differences in the number of available amplifiers depending on the requested BACnet property: Most properties provide us with about 14 k amplifiers, whereas three properties are available on significantly fewer devices: 2316 (15.0 %) of BACnet devices send us their serial number, 1958 (12.7 %) give information about their profile name, and 1389 (9.0 %) provide their list of available properties. We investigate the reason for this and find that many devices answer with the BACnet error *property unknown* for these three properties. This is not surprising as the properties *serial number*, *profile name*, and *property list* were only added in 2012 to the BACnet standard. In conclusion, this analysis shows that we need to take the different numbers of amplifiers into account when trying to assess the potential threat posed by BACnet-based amplification attacks.

Table 6.4: Property BAF and payload BAF as mean over *all*, top *50 %* and top *10 %* amplifiers.

		Property BAF			Payload BAF		
Property	Amplifiers	all	50 %	10 %	all	50 %	10 %
model_name	14 072	6.2	8.3	8.5	1.5	1.7	1.7
vendor_name	14 072	9.0	13.9	14.5	1.8	2.2	2.3
firmware_revision	14 072	11.2	19.6	35.0	2.0	2.8	4.2
app_sw_version	14 071	5.9	10.3	14.0	1.5	1.9	2.2
object_name	14 039	6.8	9.1	11.0	1.6	1.8	2.0
description	13 741	5.5	10.9	13.0	1.4	1.9	2.1
location	13 360	2.5	5.1	7.5	1.1	1.4	1.6
serial_number	2316	4.9	5.6	5.0	1.4	1.4	1.4
profile_name	1958	5.0	7.0	7.0	1.5	1.8	1.8
property_list	1389	141.0	193.8	200.0	7.3	9.7	10.0

6.5.3 Amplification Factor of Scanning Payload

We now investigate the bandwidth amplification factor for our used BACnet scanning payload.

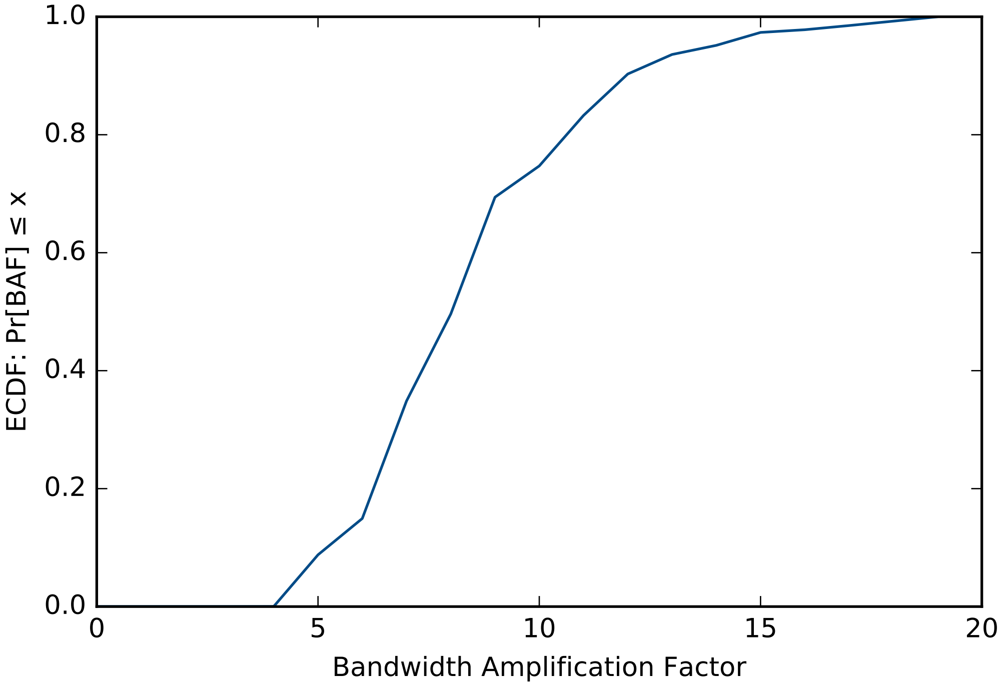


Figure 6.2: Distribution of BAF for our generic *ReadPropertyMultiple* amplification payload used in scans.

Figure 6.2 shows the empirical CDF of the bandwidth amplification factor for our 49 bytes long scanning payload. We can see that more than 90 % of requests generate responses with a BAF ≥ 5. The median BAF is 9, and the maximum BAF is 19.8 (with a response payload length of 942 bytes).

6.5.4 Amplification Factor per Property

We now evaluate the BAF on a per property basis i.e., if we would send a request for a single property. To this end, we first calculate the sending and receiving overhead of BACnet headers and the static part of the payload. The sending overhead ($SEND\_OVERHEAD$) and receiving overhead ($RECV\_OVERHEAD$) caused by BVLC, NPDU, and APDU headers in addition to the static part of the BACnet payload is 19 bytes. When requesting a property we need to add 2 or 3 additional bytes to the sent payload, depending on the length of the requested property ID ($prop\_id\_len$). With the response property length ($prop\_len$), we can now calculate the BAF for a single property payload as shown in Equation (6.1).

$$BAF = \frac{RECV\_OVERHEAD + prop\_len}{SEND\_OVERHEAD + prop\_id\_len} \tag{6.1}$$

Table 6.4 also shows a per-property BAF analysis: Property BAF details the length ratio of returned property and queried property ID. Payload BAF shows the received and sent payload length ratio for a packet requesting only this property.

We can see that *property list* has by far the largest property BAF with an average of 141. On the other hand, more than ten times as many amplifiers are available for properties such as *description*, *location* or *model name*. The property *firmware revision* combines many available amplifiers with a high BAF.

Due to the overhead introduced by BACnet headers, the payload BAF is much smaller than the property BAF.

6.5.5 Tuning the BACnet Payload

When issuing a request for a single property (as simulated with payload BAF in Table 6.4), the amplification potential of BACnet is not fully exploited. Requesting multiple properties in the scanning payload can significantly increase the payload BAF. In Figure 6.2 we see that our multi-property scans generate a median payload BAF of 9, exceeding all single-property mean payload BAFs in Table 6.4.

To raise the payload BAF even further, we can tailor a payload of multiple requests of the same property with a high property BAF factor. We very carefully test this behavior with a small number of BACnet devices. The devices not only answer the request without error, but also send the property multiple times. This allows us to leverage the property BAF, minimize the overhead of BACnet headers, and hence boost payload BAF factors up to 120.

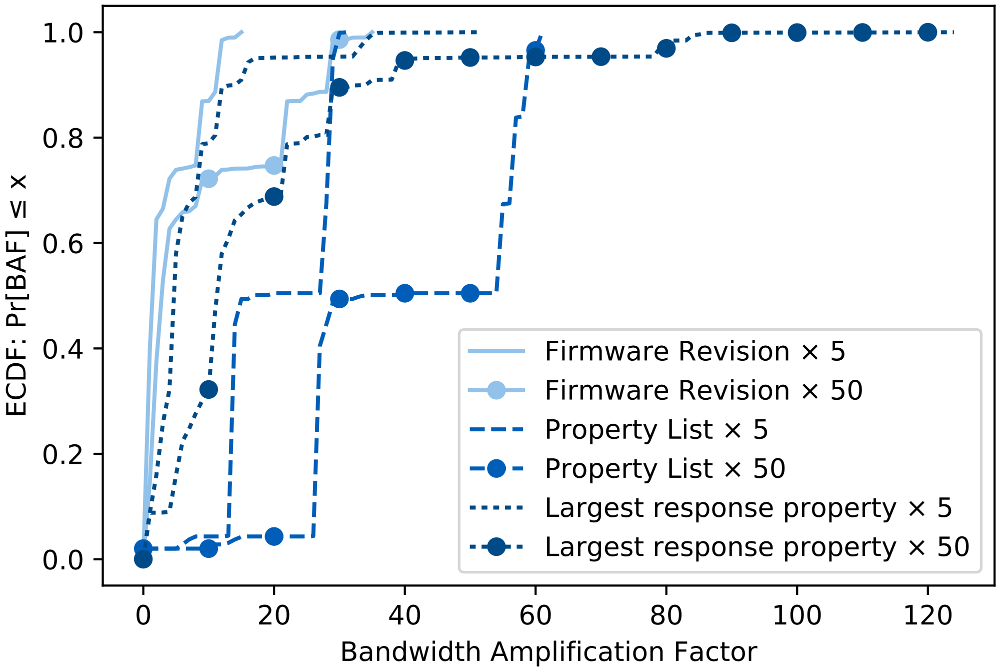


FIGURE 6.3: Payload BAF when issuing multiple requests for the same property (within a single Multi-Property packet).

Figure 6.3 shows the distribution of payload BAF when the same property is requested multiple times. In this comparison, we choose the properties *property list* as it provides the largest average BAF and *firmware revision* as it has the most amplifiers with second largest average BAF. Additionally, we include the property triggering the largest response on a per-device level. The influence of BACnet headers decreases when we increase the number of requested properties from 5 to 50.

The majority of the amplifiers answering to *firmware revision* requests give us a BAF below 10. About 10 % offer a BAF of about 30 when requesting this property 50 times.

Requesting *property list* five times already generates a larger BAF than 50 requests for *firmware revision*. About half of the 1389 amplifiers generate a BAF of 27 and 55, for 5 and 50 requested properties, respectively. This BAF is larger than for SNMP-based amplification attacks and similar to those exploiting open DNS resolvers [207].

The distinctively noticeable steps in Figure 6.3's *property list* distributions are a result of vendor clustering: Devices produced by Trane, which occur 449 times, always have a *property list* length of 93 bytes. We found that all devices by Reliable Controls send a 188 bytes or longer *property list*. This is a consequence of the large number of properties supported by these devices. However, it also means that these devices are particularly valuable targets for attackers who want to misuse them in amplification attacks.

Using the largest property on a per-device level includes all BACnet devices and gives us a higher BAF than *firmware revision*. 30 % of all BACnet devices allow for a BAF of 20 or larger. This type of attack, however, is more complex than simply choosing a single property: A preceding reconnaissance scan to find the largest property for each device and a device-specific payload would be necessary.

6.6 BACnet Traffic in the Wild

In this section we evaluate BACnet traffic as observed in the wild through two vantage points: First, we look at flow data of a large European IXP in Section 6.6.1. Second, we analyze raw packet data at a Japanese research backbone network in Section 6.6.2.

6.6.1 IXP Flow Data

Our first vantage point at a large European IXP allows us to obtain an authentic view of BACnet traffic in the Internet [45]. The IXP is located in central Europe and interconnects about 700 ASes which exchange more than 5 Tbit/s at peak times. We rely on flow data from the IXP's switching fabric, where we sample every 10,000th packet from December 1, 2016 until July 12, 2017. Due to technical issues flow data is missing between January 16 and February 5.

To preserve comparability with active scanning we filter UDP traffic on all 16 BACnet ports. We remove traffic with ports < 1024 as these are very likely cases where a BACnet port was randomly chosen as an ephemeral client port. We also identify traffic from our own active BACnet scans by source IP address. Figure 6.4a shows the BACnet traffic volumes in Mbit/s at the IXP for the measured period, with the overwhelming majority being IPv4 traffic. In addition to our own scans which are clearly identifiable, we notice a spiky pattern which indicates regular scanning activities by other parties. Most importantly, we see a steady decline in traffic levels to less than 100 Mbit/s after we notify affected networks via the CERT (denoted by the vertical line). More details on the notification campaign are described in Section 6.7. In Figure 6.4b we pivot to a different unit of measurement, i. e., packets per second. This confirms our findings as (1) scanning patterns are clearly visible again and (2) the number of packets decreases after the CERT-backed notification campaign.

Next, we analyze the distribution of transport layer ports for BACnet traffic seen at the IXP. The distribution of used ports is quite different for source and destination, as can

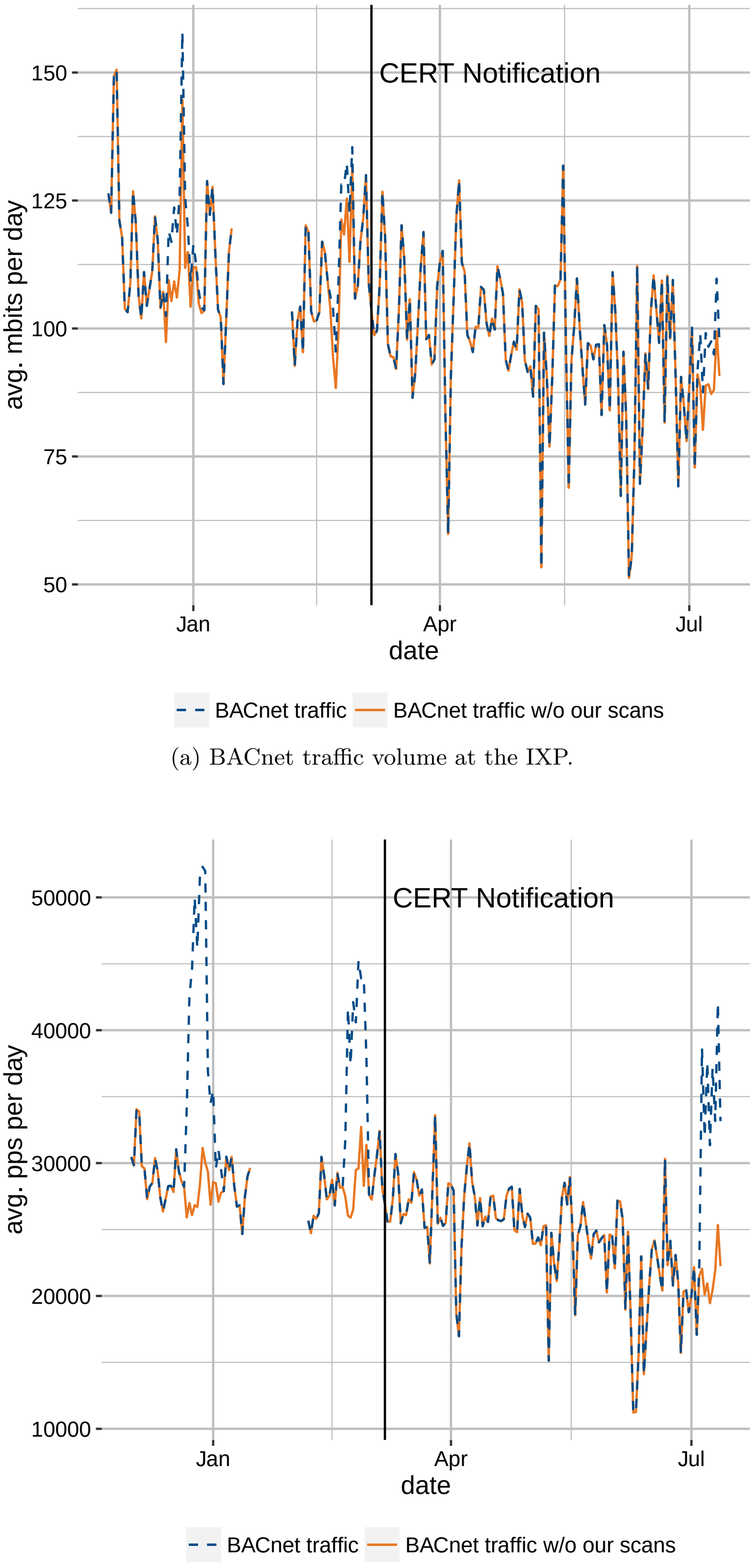
Table 6.5: Top 5 source and destination ports of BACnet traffic in IXP dataset, ordered by destination port.

Port	Dst %	Src %
47808	21.34	2.82
47820	2.71	2.83
47822	2.66	2.84
47816	2.64	2.71
47810	2.61	2.56

be seen in Table 6.5. The source port distribution is dominated by BACnet ports, evenly distributed from 2.82 % for port 47808 to 2.24 % for port 47819. The top non-BACnet source port is 7985 with 1.83 % of flows. The distribution of destination ports is different. Port 47808 is the most frequent one and accounts for 21.34 % of all seen BACnet flows. The other BACnet ports are again evenly distributed and in the range from 2.7 % to 2.2 %. In both distributions we find (after the BACnet ports) a small fraction of UDP application ports, each with a share of < 1% (except for source port 7985). This is most likely UDP application traffic on non-privileged ports where the client has accidentally chosen a BACnet port as an ephemeral port. The port distribution indicates that the majority of BACnet traffic stems from scans on port 47808. Scanners use 47808 as their destination port, but get few responses which is why the percentage of 47808 on the source port side is much lower.

In Figure 6.5 we plot the packet size distribution of BACnet flows. While small packets between 66 and 83 bytes are the most prevalent, there is also a significant number of packets larger than 1400 bytes. This hints at a large number of small sized scanning packets (typically around 45 to 70 bytes) in combination with application layer UDP data where a BACnet port was by chance chosen as the ephemeral port.

In summary, a large portion of BACnet traffic seen at the IXP is most likely scanning traffic. We analyze this phenomenon more in depth using raw packet data in the following Section 6.6.2.



(a) BACnet traffic volume at the IXP.

(b) BACnet packets per day at the IXP.

Figure 6.4: BACnet traffic at the IXP. We distinguish between traffic with and without our own scans. The date of the CERT-backed notification campaign is denoted by a vertical line. Created by Benedikt Rudolph [106].

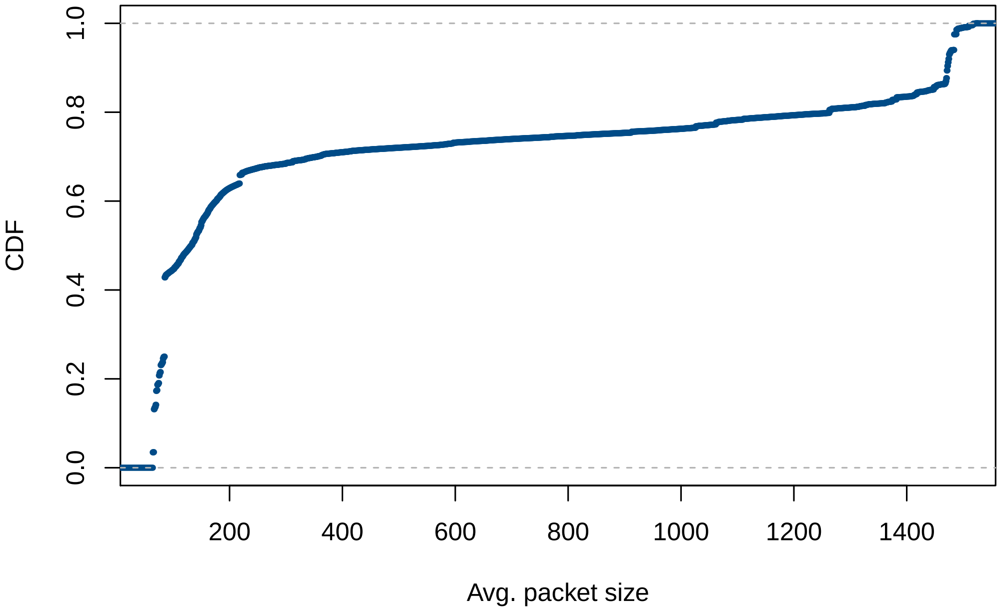


Figure 6.5: Cumulative distribution of the average BACnet traffic packet size. Created by Benedikt Rudolph [106].

6.6.2 MAWI Raw Packet Data

To gain additional insight into BACnet traffic beyond flow data, we analyze raw packet data from our second vantage point, the MAWI dataset [255]. We use 48 hour traces captured at the transit link of the WIDE research network to the upstream ISP on April 12 – 13, 2017. The traces comprise of 3.9 G packets with more than 12 TB of data, resulting in an average data rate of 550 Mbit/s.

We use a filtering cascade to identify likely BACnet traffic for later analysis. We first filter the dataset to only retain UDP traffic on all 16 BACnet ports. 403 837 packets are remaining after this step. We then remove traffic where a BACnet port was chosen as an ephemeral client port. We do this conservatively by eliminating traffic where we find a low port (< 1024) in combination with a BACnet port. By looking at the payload of top non-BACnet ports we find additional occurrences of BACnet as an ephemeral port and remove Teredo, SSDP, and Netis router backdoor scans. After this port filtering stage 339 274 packets with traffic on BACnet ports remain.

Next, we parse the UDP payload to find out whether the remaining traffic is in fact BACnet traffic. We identify BACnet/IP traffic by filtering for the the distinct BACnet/IP transport type (`0x81`) at the beginning of the UDP payload. We also try to identify BACnet/IPv6 traffic (transport type `0x82`), but could not find any. Interestingly, however, we find 31 packets with BACnet/IP payloads built for IPv4 but sent over IPv6. We also check the valid BACnet version (`0x01`) and ensure that no reserved NPDU control bit is set. In contrast to our filtering procedure for active scans, we do not restrict the payload to BACnet-ComplexACK-PDU, but allow all valid BACnet application payload types. In the payload filtering phase we remove 4120 packets, with 335 154 packets with BACnet payload remaining.

We analyze the BACnet payload of the remaining packets which are all destined to port UDP/47808. Surprisingly, these 335 154 packets contain only four different payloads. All four payloads are BACnet requests, no responses are present. This hints at scanning activities instead of regular BACnet traffic.

Table 6.6 shows an overview of the four payloads, with the number of seen packets, number of source IP addresses, requested BACnet properties, and a classification of the scan.

Table 6.6: BACnet packets in MAWI dataset classified according to their payload.

#	Packets	Source IPs	Req. Properties	Classification
1	263 273	10	List of 10 properties	Short Time Scanning Project
2	66 670	26	Object ID	Shodan
3	4441	1	Vendor ID	Chinanet
4	770	242	List of 9 properties	Kudelski Security

In the following we analyze all four payloads in detail:

Payload #1 The most common payload contains a Multi-Property request querying a list of 10 properties. It is sourced from 10 IPv4 addresses in different subnets and Autonomous Systems. The reverse DNS name mapping of two of the IP addresses hints at rented private servers, one rDNS entry hints at a research project (*thisissecurityresearch.com*). We find a web server running on eight of the ten IP addresses, informing of a "Short Time Scanning Project" and giving the possibility to be excluded from scans.

Payload #2 This payload requests the object ID and occurs in more than 66 k packets. We find 25 IPv4 source addresses and one IPv6 source address. The single IPv6 address, however, sends a BACnet/IP payload instead of a correct BACnet/IPv6 payload. We attribute 23 of the 25 IPv4 addresses to the scanning service Shodan [224] due to the reverse domain name mapping.

Payload #3 The third most common payload requests the vendor ID and stems from only a single IPv4 address without an rDNS entry. The IP address belongs to the Autonomous System of Chinanet, a Chinese ISP.

Payload #4 The least common payload consists of a Multi-Property request containing 9 properties. Its 242 IPv4 addresses are all located within the same /24 subnet belonging to Kudelski Security, a company offering Internet security services. The WHOIS entry also states that the network is used for port scanning activities.

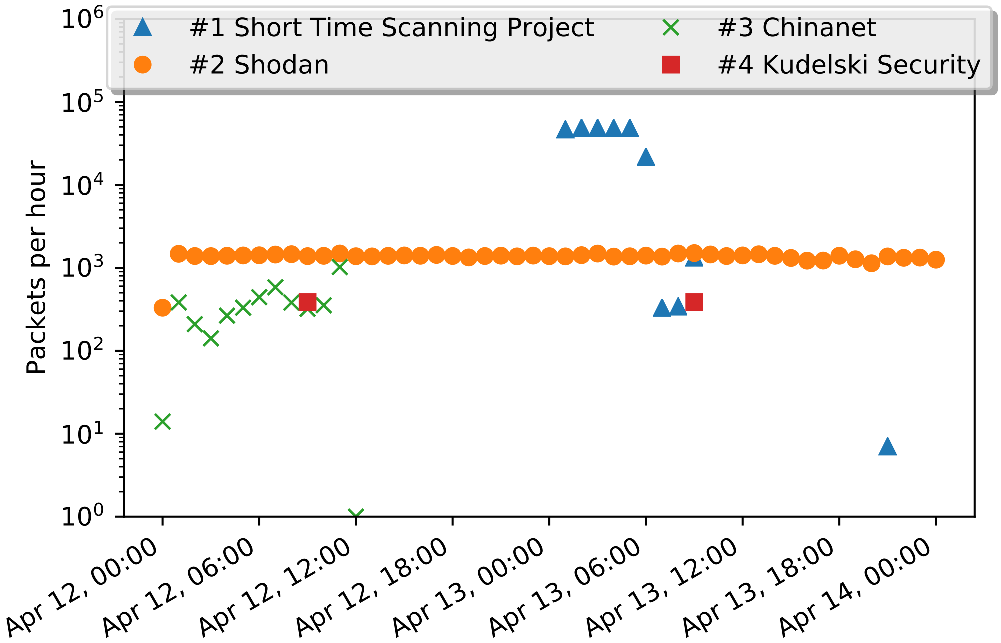


Figure 6.6: Packets per hour for each of the four payloads. Continuous vs. burst scanning clearly visible. Note that the y-axis is log-scaled.

Next, we analyze the temporal scanning patterns of the four payloads. Figure 6.6 shows the number of packets seen per hour for each different payload. We can clearly see distinct scanning patterns: The "Short Time Scanning Project" conducts high-rate scans with more than 40 k packets per hour. These high-rate scans, however, only last for some hours, after which they decrease in rate and vanish completely. This bursty phenomenon could be due to non-random scanning, where adjacent IP addresses are probed close after each other. Shodan on the other hand continuously scans for BACnet devices over the two day period with a very constant packet rate. In the MAWI dataset we find about 1000 packets each hour originating from Shodan IP addresses. Chinanet scans are only observed in the first six hour period, exhibiting a relatively constant packet rate. Kudelski Security seems to be conducting brief daily scans, which we observed at the same hour each day.

To summarize, the MAWI dataset gives us a glimpse into BACnet traffic, specifically scanning practices. We do not find any bidirectional BACnet traffic as we see only BACnet requests. This hints at port scans of which the majority seems to be conducted by security companies and researchers. We identify clear temporal scanning patterns based on the four payloads.

6.7 Notification Campaign

We use our measurement results to improve Internet security by notifying the owners of affected BACnet devices. We cooperate with the DFN-CERT, which is the Computer Emergency Response Team for the German National Research and Education Network (DFN). We supply the DFN-CERT with relevant information of the affected systems. The DFN-CERT notified 76 different CERT teams and additionally the CERT Coordination Center for affected systems in the US and Canada. The notifications were sent out in the week of March 7, 2017. By notifying vulnerable systems we hope to reduce the number of publicly reachable and abusable BACnet devices. Li *et al.* show that notification campaigns can drive measurable impact [158]. We assess the impact of our notification campaign using follow-up active scans and passive analysis at a large European IXP.

6.7.1 Follow-up Active Scans

To assess the impact of our notification campaign we conduct a total of four IPv4-wide BACnet scans. Table 6.7 shows all four scans: The first was conducted in December 2016 for BACnet deployment analyses. In February 2017 a second scan was performed to get an updated list of IP addresses. This list was given to the DFN-CERT, which conducted the notification in the beginning of March 2017. The third and fourth scans were conducted to assess the impact of the notification campaign, in July and August 2017 respectively.

Table 6.7: Overview of conducted IPv4-wide BACnet scans to assess notification campaign impact.

Scan date	Responses	BACnet	Unique IPs	Prefixes	ASes	Unique IDs
Dec 2016	41 103	16 485	15 350	5110	1439	9319
Feb 2017	39 581	16 645	15 495	5159	1465	9392
Jul 2017	758 611	16 351	15 152	5020	1425	9269
Aug 2017	141 567	16 247	15 030	5040	1428	9188

In Table 6.7 we see that we receive many more responses in the third and fourth scans. The vast majority of these responses, however, are not genuine BACnet responses. Instead they are mirrored BACnet request packets containing the exact same payload as our scans. These mirrored packets could be caused by misbehaving or misconfigured hosts or routers on the path. As we only evaluate genuine BACnet responses, we remove all mirrored packets. The number of valid BACnet responses and unique IP addresses remains mostly constant over the four scans, with small reductions in the July and August 2017 scans. The number of network prefixes and Autonomous Systems (ASes) with BACnet devices decreases slightly between the second and third scan.

To better understand these slight trends we evaluate the sets of responding IP addresses of each scan in more detail and correlate them with each other. During this analysis we see that BACnet IP addresses are not steadily going offline as suggested by data in Table 6.7, but exhibit a rather churny behavior. We find 10 841 IP addresses which respond to at least one but not all our probes. In each scan about half of these IP addresses are responsive. These IP addresses can be deemed unstable or dynamic.

We compare the AS distribution of the unstable IP addresses to those of all IP addresses but can not find any major differences.

To further analyze the changes between the scans we try to fingerprint BACnet devices. We use the device properties model name, location, object name, vendor ID, and vendor name to create a device ID (i. e., BACnet device fingerprint). We then check whether this ID is unique in each of the four scans. We find about 9 k unique IDs per scan. Next, we check if the IDs of IP addresses stay the same on subsequent scans. About 400 IP addresses change their unique device ID between subsequent scans. By inspection we deduce that most of these changes are caused by IP address reassignment and the resulting device ID swapping. Additionally, we identify unique device IDs which change their IP address between scans. There are between 220 and 450 of these devices. When inspecting the corresponding AS, between 68 % and 87 % of these IP addresses belong to the same Autonomous System.

Consequently, we conclude from the follow-up active scans that the majority of dynamic BACnet devices are a result of IP address changes within the same organization. Even though the overall number of publicly reachable BACnet devices has slightly decreased, the notification campaign seems to have had only a small impact.

6.7.2 IXP Temporal Comparison

In addition to our follow-up active scans we also conduct passive analysis using an IXP dataset, presented in Section 6.6.1. When looking at Figures 6.4a and 6.4b we see a slight downward trend of BACnet traffic and packets respectively. This again suggests that the notification campaign contributed to an improvement of the situation.

6.8 Discussion

We use this section to discuss the implications of our results and expand on ideas on how to improve the security state of BACnet devices. Accordingly, this section explores: (1) strategies for affected parties to detect and prevent BACnet-based attacks and (2) actions that the community can take to remedy the problem of publicly accessible BACnet devices.

6.8.1 Mitigation Strategies

Affected network operators can adopt various strategies to reduce the impact of BACnet attacks. First, and preferably, the operator of the device can move the device to a separate, not publicly accessible network enforced by, e. g. VLAN or VPN. However, this may not be feasible in many small network scenarios. Second, the network operator may deploy rule-based access filtering to restrict access from the public Internet. Third, at network operator or ISP level, strategies may be deployed to detect ongoing amplification attacks, e. g. by measuring traffic entropy [37]. Detected attacks could be rate-limited by an ISP.

6.8.2 Standardization Efforts

BACnet standardization could harvest quick wins in mitigating amplification attack potential by not allowing multiple reads of the same property within the same packet, which we found critical in achieving a high BAF. However, this comes with a certain complexity and computational cost. We contacted ASHRAE regarding changing the BACnet standard to thwart the attack potential, but did not receive a reply.

6.9 Related Work

In this section we elaborate on existing related work in the areas of Internet scanning for BACnet and similar protocols, amplification attacks, and vulnerability notification. We highlight differences to our study and compare our findings to theirs.

6.9.1 Internet-wide Scanning

Both Mirian *et al.* [168] and Feng *et al.* [90] scan for BACnet and other ICS devices. In contrast to them we scan for all 16 standardized BACnet ports. We identify twice as many IP addresses that do not respond on port UDP/47808 for a total of 3.7 k valid BACnet responses missed by previous research. Neither of them discusses amplification potential of BACnet.

Censys [72], Project Sonar [197], and Shodan [224] perform regular BACnet scans, finding between 5.2 k and 7.8 k fewer devices than our scans.

6.9.2 Amplification Attacks

In 2014, Rossow [207] investigated numerous UDP protocols for their susceptibility to amplification attacks. He measures amplification factors, verifies the number of available reflectors and estimates how quickly they could be harvested by a malicious actor. We add BACnet to the list of affected protocols and evaluate its potential for amplification attacks.

In 2017, Sargent *et al.* [211] discuss the amplification potential of IGMP. They find ∼305 k amplifiers with a median amplification factor of 2.4. For BACnet, we find fewer amplifiers but a significantly higher amplification factor.

To detect amplification attacks at the reflector network, Böttger *et al.* propose a protocol-agnostic technique based on BAF and payload entropy [37]. Krämer *et al.* present AmpPot, a honeypot designed to track amplification attacks [147].

6.9.3 Notification

In 2016, Li *et al.* [158] investigated the effectiveness of reporting vulnerabilities to operators. They identify 45 770 devices supporting at least one industrial protocol. They compare remediation rates based on communication method, verbosity, website link, translated messages. They achieve a remediation rate of about 8 %. This measurable impact motivates our notification campaign. Our notification campaign, however, was not able to achieve these high remediation rates.

6.10 Conclusion

We conducted multiple Internet-wide active measurements to identify 16 485 BACnet devices. We found that they were heavily clustered in certain ASes and prefixes. Subsequently, we uncovered that 14 k of these devices can be misused for amplification attacks. We evaluated the bandwidth amplification factor for a single property requested once, and a tuned payload where the same property is requested multiple times. Using this tuned payload we achieve amplification factors up to 120. We evaluated BACnet traffic in the wild and attributed the majority of it to scanning projects. Finally, we conducted a notification campaign through a CERT, observed small reductions in the number of BACnet devices, and give further advice on how to secure BACnet deployments.

6.11 Key Contributions of This Chapter

This chapter addressed research question RQ IV which includes challenges C 9, C 10, and C 11. We showed that BACnet devices could be reached from the public Internet and analyzed the BACnet device deployment. Additionally, we identified vulnerabilities in the BACnet protocol which could lead to them being misused in amplification attacks. Moreover, we conducted a notification campaign in coordination with a national CERT and evaluated the number of reachable BACnet devices over time.

In the following we list the key contributions of this chapter:

BACnet deployment We performed multiple Internet-wide measurements and identified 16 k publicly reachable BACnet devices. These devices exhibited significant typological (majority of devices from few manufacturers), geographical (majority of devices in small number of countries), and topological (majority of devices in small number of ASes) clustering. With the results from this analysis we tackled challenge C 9.

Amplification attacks with BACnet devices We surveyed the theoretical eligibility of the BACnet protocol for amplification attacks which lead to an initial confirmation. Subsequently, we built several BACnet payloads and analyzed their effective amplification factor. In this analysis we applied several tweaks (e. g. Multi-Property instead of Single-Property message, requesting the same property multiple times) to further increase the potential amplification factor. We demonstrated that BACnet devices could indeed be misused as amplifiers in amplification attacks, thus tackling challenge C 10.

Notification campaign After evaluating the results of our research, we conducted a notification campaign through a large and well-connected CERT to notify affected parties. We saw a small drop of publicly reachable BACnet devices in the months after the notification campaign. This showed us that notification campaigns can indeed drive measurable impact by reducing the number of vulnerable devices. With these results we tackled challenge C 11.

After tackling these three challenges we can provide an answer to RQ IV, as we find that there are publicly reachable BACnet devices which can be misused as amplifiers in amplification attacks. We can therefore positively answer this research question as BACnet devices are indeed vulnerable to amplification attacks.

6.12 Statement on Author's Contributions

This chapter is partially based on the publications "The Amplification Threat Posed by Publicly Reachable BACnet Devices" by Oliver Gasser, Quirin Scheitle, Benedikt Rudolph, Carl Denis, Nadja Schricker, and Georg Carle, which was published in JCSM 2017 [106]; "Security Implications of Publicly Reachable Building Automation Systems" by Oliver Gasser, Quirin Scheitle, Carl Denis, Nadja Schricker, and Georg Carle, which was published at WTMC 2017 [105]; "Öffentlich erreichbare Gebäudeautomatisierung: Amplification-Anfälligkeit von BACnet und Deployment-Analyse im Internet und DFN" by Oliver Gasser, Quirin Scheitle, Carl Denis, Nadja Schricker, and Georg Carle published at DFN-Konferenz Sicherheit in vernetzten Systemen 2017 [107].

The author made the following contributions to the BACnet study presented in the JCSM 2017 paper [106], WTMC 2017 paper [105], DFN 2017 paper [107] and the adapted sections in this chapter. The BACnet deployment analysis work was first started as part of Nadja Schricker's Bachelor's thesis ("Active Security Evaluation with Network Scans") [222], which the author advised. The author coordinated and lead the group of researchers in this study. The author provided significant contributions in the study's design and execution. Moreover, the author contributed significantly in conducting the measurements and in analyzing the BACnet deployment, the amplification potential and publicly available packet data. The author provided minor contributions in the analysis of IXP data. Finally, the author contributed significantly in coordination and evaluation of the notification campaign and discussion of implications.

In the following we describe changes between this chapter and the BACnet study presented in the JCSM 2017 paper [106], WTMC 2017 paper [105], and DFN 2017 paper [107]. Most of the analyses in the papers are present in this dissertation. We shorten the analysis of the IXP data and provide only highlighted results, as this analysis is not the focus of this thesis.

Chapter 7

IPMI

This chapter is based on the publication "Digging for Dark IPMI Devices: Advancing BMC Detection and Evaluating Operational Security" by Oliver Gasser, Felix Emmert, and Georg Carle, which was published at TMA 2016 [98]. In Section 7.8 details on the author's contributions and differences between the thesis text in relation to the published papers are given.

IPMI stands for "Intelligent Platform Management Interface" and it is the industry standard for managing devices remotely independent of their operating status. Since there are known vulnerabilities in the protocol, IPMI devices should not be directly reachable on the Internet. Previous studies suggest, however, that this best practice is not always followed. In this research we present a new unintrusive technique to find hidden IPMI devices (so called *dark* IPMI devices) through active measurements. These dark devices do not respond to conventional IPMI connection setup requests. Using our technique, we find 21 % more devices than previously known techniques. This adds a significant number of IPMI devices which could be exploited by an attacker using a Man-in-the-Middle attack. We further reveal that IPMI devices are heavily clustered in certain subnets and Autonomous Systems. Moreover, the TLS security of IPMI devices' web-interface is well below the current state of the art, leaving them vulnerable to attacks. Overall our findings draw a dire picture of the current state of the IPMI deployment in the Internet.

This chapter is structured as follows: First we introduce the topic and provide information on IPMI and out-of-band in general in Section 7.1. We also lay out security problems which are the motivation for conducting our research. In the following Section 7.2 we present related work in the area of discovering IPMI devices and assessing their security. Section 7.3 provides information about the IPMI protocol and the scanning techniques used during the experiment. In Section 7.4 we detail our scanning approach, first describing *dark IPMI devices* and which IP addresses we target during our measurements. Then we describe the software used for our network measurements and lay out our ethical considerations. In Section 7.5 we present the results of our scans and classify them with regard to security of out-of-band management. We conclude in Sections 7.6 to 7.8 by summarizing our IPMI study, listing its key contributions to this dissertation, and providing a statement on the author's contributions.

7.1 Introduction

Out-of-band network management is the process of managing devices and systems over an auxiliary communication channel, independent of their operating state. Out-of-band management enables administrators to remotely manage servers, routers, switches, and other devices. This management capability is especially important when managing hundreds or thousands of these devices, such as in data centers or colocation centers.

The de facto industry standard protocol for out-of-band management is IPMI (Intelligent Platform Management Interface). The IPMI protocol also specifies access to IPMI devices over the network via IPMI-over-IP.

The IPMI-over-IP protocol, however, has some inherent weaknesses. Attackers can exploit insufficient authentication checks and other vulnerabilities to compromise the host system as detailed e. g. by HD Moore [118]. These weaknesses allow an attacker to gain access to a powerful interface over the network. This introduces new attack vectors independent of the host system's security. Once gained access to an IPMI device, an attacker can e. g. power off the device (essentially a Denial-of-Service attack), rebooting into a custom operating system, or installing a rootkit to eavesdrop on the communication. In short, an attacker has full control over the system and can potentially compromise the IPMI device itself.

To better assess these risks it is important to understand the IPMI deployment in the Internet. Therefore we conduct large-scale scans to find openly accessible IPMI devices and classify their security properties. We use an unintrusive measurement technique which does not attempt any authentication with the IPMI devices. We perform the scans using a modified version of ZMap, which we make available online. As a result of the scans we find significantly more IPMI devices than other current scanning efforts. Moreover, we discover that a large number of IPMI devices are not properly secured.

7.2 Related Work

In this section we present previous work related to our research. First, we survey research which analyzes the security of IPMI devices. Then we detail works in the field of Internet-wide scans for security purposes.

7.2.1 Security of IPMI Devices

In 2013, Bonkoski *et al.* [30] surveyed the security of IPMI implementations. They analyzed firmwares of IPMI devices from Supermicro and found exploits which can be used to bypass the authentication of the web front-end and gain access to the system. Furthermore, the authors estimated the number of potentially vulnerable IPMI devices by looking at TLS certificates from large-scale scans. They found that more than 40 000 potentially vulnerable IPMI devices and more than 100 000 IPMI devices in total exist in the wild. Similarly to Bonkoski *et al.* , we also use TLS certificates to identify potential IPMI devices as targets for our active measurements. In our Internet-wide measurement we found more than 220 000 IPMI devices.

In the same year Dan Farmer, [85, 86] performed Internet-wide scans for UDP/623 and found more than 230 000 IPMI devices. He analyzed the security of these devices and found that many were vulnerable to authentication weaknesses and that passwords could be brute-forced. Although we found slightly less devices in 2015 than Dan Farmer in 2013, we found many dark IPMI devices, i. e., devices which do not respond to Farmer's scanning technique.

Zhang *et al.* [261] in 2014 tried to correlate the maliciousness of networks (e. g. sending spam emails) with mismanagement metrics. One of their mismanagement metrics was the reachability of IPMI devices within Autonomous Systems (ASes). By matching regular expressions on the subject field in TLS certificates, they identified about 100 000 publicly reachable IPMI devices in different ASes. We use a similar technique to match found IPMI devices to vendors. They found a weak correlation between the reachability of IPMI devices in networks and the maliciousness of a network.

In 2014, Costin *et al.* [57] performed a large-scale analysis of embedded firmwares. They gathered firmware images using a web-crawler and a site where users could upload their firmware. Then the authors analyzed more than 32 000 of them and found 38 new vulnerabilities. Moreover, they were able to extract private TLS keys and crack hard-coded password hashes.

Similarly, Stefan Viehböck analyzed more than 4000 embedded firmware images in 2015 [252]. Unfortunately, most of the issues previously found still persist: the author was able to extract 580 unique private keys. These keys are used by 9 % of all TLS hosts on the IPv4 Internet and 6 % of SSH hosts.

Rapid7 performs monthly IPMI scans and publishes the raw response packets as part of *Project Sonar* [197]. Compared to their scanning efforts we use a different scanning technique (see Section 7.3.4) which leads to more detected IPMI devices.

7.2.2 Large-scale Security Measurements

Large-scale network measurements have recently become a valuable tool to assess specific aspects of the Internet's security. Holz *et al.* [129] conducted active and passive measurements in 2011 to assess the security of the TLS PKI. They identified multiple security issues in the TLS deployment such as incorrect certificate chains and invalid subject names in certificates.

In 2012, Heninger *et al.* [123] evaluated the cryptographic properties of TLS and SSH. They concluded that due to a lack of randomness, many keys were predictable.

In 2014, Gasser *et al.* [99] conducted multiple Internet-wide SSH scans. They were able to confirm many of Heninger *et al.* 's findings and additionally found duplicate yet cryptographically strong keys.

Similar to Heninger *et al.* , we analyze the TLS certificates of web interfaces to evaluate the security of IPMI devices. The certificates found on IPMI devices are not suitable to properly secure connections.

7.3 IPMI Background

In this section we give general information about out-of-band network management. We also provide insights into the protocols relevant for this research.

7.3.1 Out-of-Band Network Management

Out-of-band network management is a term describing different technologies enabling system administrators to remotely manage their network hardware (e. g. switches, routers, servers) independently of the system's operating state. This goal is commonly achieved by independent sub-systems connected to the main system's network hardware. These sub-systems run their own operating system on dedicated hardware and are connected to various I/O ports of the main system. Out-of-band network management devices either have their own network interface controllers (NICs) or access to one of the main system's NICs via a "side-band" interface. Out-of-band management devices commonly provide a management interfaces for administrators, e. g. using a web interfaces or via SSH.

7.3.2 IPMI Basics

Intelligent Platform Management Interface (*IPMI*) is the de facto industry standard for out-of-band network management devices used for server management. The IPMI specification [138] defines the architecture, different functionalities, and user interfaces of out-of-band network management devices for servers. IPMI devices run on embedded microcontrollers called *Baseboard Management Controller* (*BMC*). BMCs are commonly installed via daughter cards or directly integrated in the server's mainboard. IPMI's functionality may be extended by BMC manufacturers. A common extension are web interfaces on TCP ports 80 (HTTP) and 443 (HTTPS).

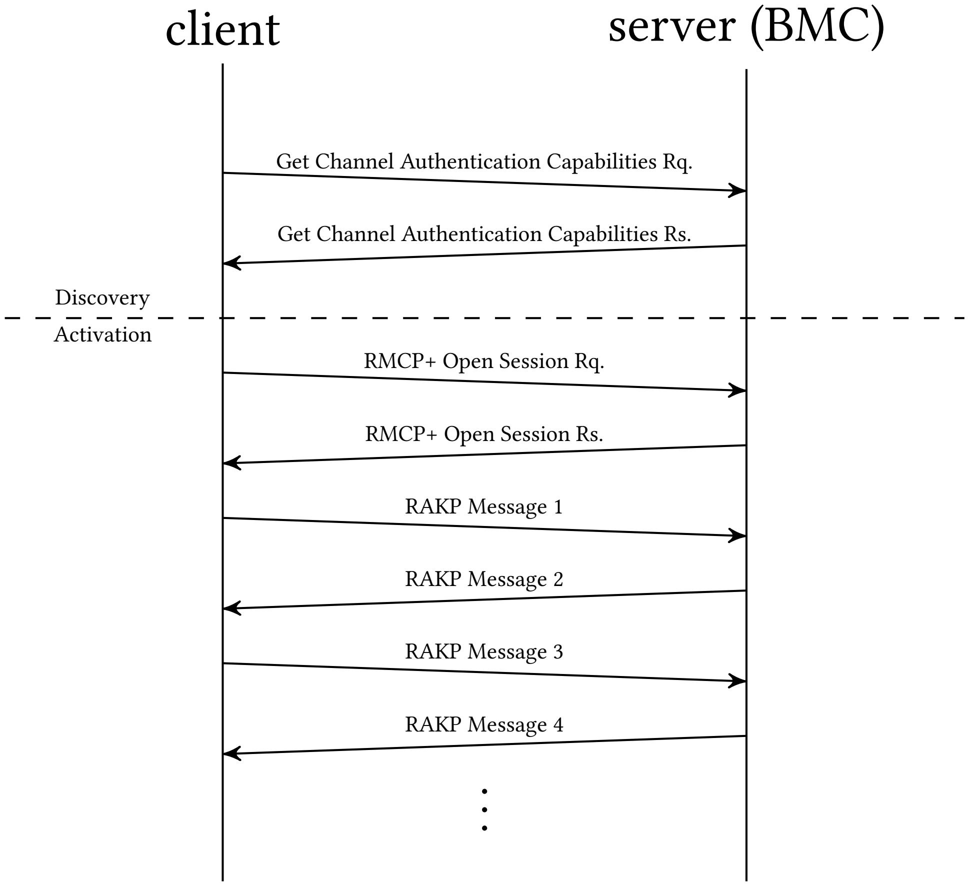


Figure 7.1: IPMI-over-IP connection establishment in IPMI 2.0.

7.3.3 IPMI-over-IP

IPMI defines its own network protocol called *IPMI-over-IP* which runs over UDP port 623 [138]. IPMI-over-IP allows for administrators to remotely login to their BMCs and perform a set of actions like rebooting the server or configuring the BMC. Using IPMI-over-IP, it is possible to take full control over the connected server. If not needed, most devices offer the possibility to deactivate IPMI-over-IP.

IPMI-over-IP has been introduced in version 1.5 of the IPMI specification and it was updated in the new version 2.0 of the specification. However, IPMI version 2.0 devices are still required to simultaneously support the old version 1.5 as a fallback mechanism [138].

In the following we describe how an IPMI-over-IP connection is established and authenticated. IPMI-over-IP's connection establishment is divided into two phases, "Discovery" and "Activation". See Figure 7.1 for a visualization of an IPMI-over-IP connection establishment with IPMI version 2.0.

In the optional "Discovery" phase of the IPMI-over-IP protocol in version 2.0 of the IPMI specification, the client sends a *Get Channel Authentication Capabilities Request* packet to the BMC. The BMC answers with a *Get Channel Authentication Capabilities Response* packet. This response packet includes the IPMI version and authentication methods supported by the BMC. If IPMI-over-IP is deactivated, the BMC will not respond.

In version 1.5 of the IPMI specification, IPMI-over-IP also supports discovery using *RMCP Ping* packets. If probed, the BMC responds by sending an *RMCP Pong* packet. This response packet does not include much information other than whether or not IPMI is supported. However, small-scale tests on a *Dell iDRAC 7* show that the BMC replies to RMCP Ping packets even if the IPMI-over-IP protocol has been deactivated.

The "Activation" phase of the IPMI-over-IP protocol is only described for version 2.0 of the IPMI specification, the older version 1.5 is out of scope for this research.

The "Activation" phase of the IPMI-over-IP protocol in version 2.0 of the IPMI specification starts with the client sending an *RMCP+ Open Session Request* packet to the BMC which responds with an *RMCP+ Open Session Response* packet. These packets contain session IDs for further communication between client and BMC. The response packet also contains information about supported cipher suites.

Next, the client sends an *RAKP Message 1* packet answered by the BMC with an *RAKP Message 2* packet. These packets contain nonces for mutual authentication (later signed using the user's password) as well as the client's username and the BMC's

GUID (globally unique ID). Since the *RAKP Message 2* packet is already signed using the password of the requested username, it is possible to perform an offline brute-force attack on the password if the requested username is valid. It is also possible to perform an online brute-force attack on the username, since the BMC tells the client whether the username is valid or not.

Finally, the client sends a signed *RAKP Message 3* packet answered by the BMC with a signed *RAKP Message 4* packet. The signature is made using the user's password similar to the *RAKP Message 2* packet.

7.3.4 Different Measurement Types

It is possible to scan for IPMI devices in various ways. The IPMI-over-IP protocol defines two different discovery methods, both over UDP/623.

BMCs queried with *Get Channel Authentication Capabilities Request* packets only reply if the IPMI-over-IP protocol has been activated. The response packet contains information about the IPMI version of the BMC (1.5 or 2.0) as well as some information about supported authentication methods.

BMCs scanned with RMCP Ping packets reply with *RMCP Pong* packets. The response packets contain little to no information about the BMC other than its presence. However, small-scale tests on a *Dell iDRAC 7* device show that the BMC replies to RMCP Ping packets even if the IPMI-over-IP protocol is deactivated. That is why we presume to find additional dark IPMI devices by scanning with RMCP Ping packets.

7.3.5 TLS Basics

TLS is a security protocol based on a Public Key Infrastructure. It is used in the WWW, but also for email, chats, and other services. TLS certificates contain identity information about a peer (e. g. a domain name) and a corresponding public key. A certificate therefore creates a binding between an identity and a public key.

We use TLS certificates in our research in the following two ways: First, to identify potential IPMI devices as measurement targets (see Section 7.4.2). Second, to evaluate the security of IPMI devices with regard to cipher security and the potential for Man-in-the-Middle attacks (see Section 7.5.4).

7.4 Approach

In this section we describe the rationale and approach of our IPMI measurements. We begin by explaining the concept of dark IPMI devices. Then, we detail the two types of measurements: The first one is limited to a small subset of IP addresses including likely dark IPMI devices. The second type of measurement is a complete scan of the IPv4 address space. Finally, we detail the used scanning software and address ethical questions regarding active network measurements.

7.4.1 Dark IPMI Devices

With our measurements we want to discover *dark IPMI devices.* These are devices which have the IPMI-over-IP port disabled. Consequently they do not respond to standard IPMI scans such as those executed by Rapid7 [197]. They do, however, respond to RMCP Ping requests as required by the IPMI specification [138]. Even though dark IPMI devices do not provide direct IPMI-over-IP access, they are still valuable to attackers. Once identified as an IPMI device, attackers could exploit other attack vectors, e. g. flaws in the web interface implementation or insecure SSH connections [142] to gain access to the BMC or the host system.

7.4.2 Target List

In order to verify that there are indeed dark IPMI devices, we perform active measurements on a specific subset of IP addresses. As a starting point we use all TLS hosts identified by Project Sonar [197] as most IPMI devices provide access via a web interface. Then, we remove the IP addresses which already responded to standard IPMI scans using Get Channel Authentication Capabilities requests. These IP addresses have already been attributed to IPMI devices and are therefore not *dark.* We use the remaining IP addresses in the first type of active measurements. Figure 7.2 shows the workflow of this type of measurement.

In the second type of active measurements we probe the full IPv4 address space to get a complete picture of the IPMI deployment in the Internet.

7.4.3 ZMap

ZMap is a network analysis tool designed for scanning different network ports across the IPv4 Internet [71]. ZMap has been optimized for speed, meaning that it is capable of scanning the entire IPv4 address space in less than 5 minutes given enough bandwidth [4]. It is possible to load custom modules for packet generation or result processing.

We build a probe module for ZMap which generates *RMCP Ping* packets. Moreover, we extend ZMap to filter out incoming UDP packets that are not addressed to the scanning machine (e. g. multicast packets). We publish the modified ZMap version on our website [243].

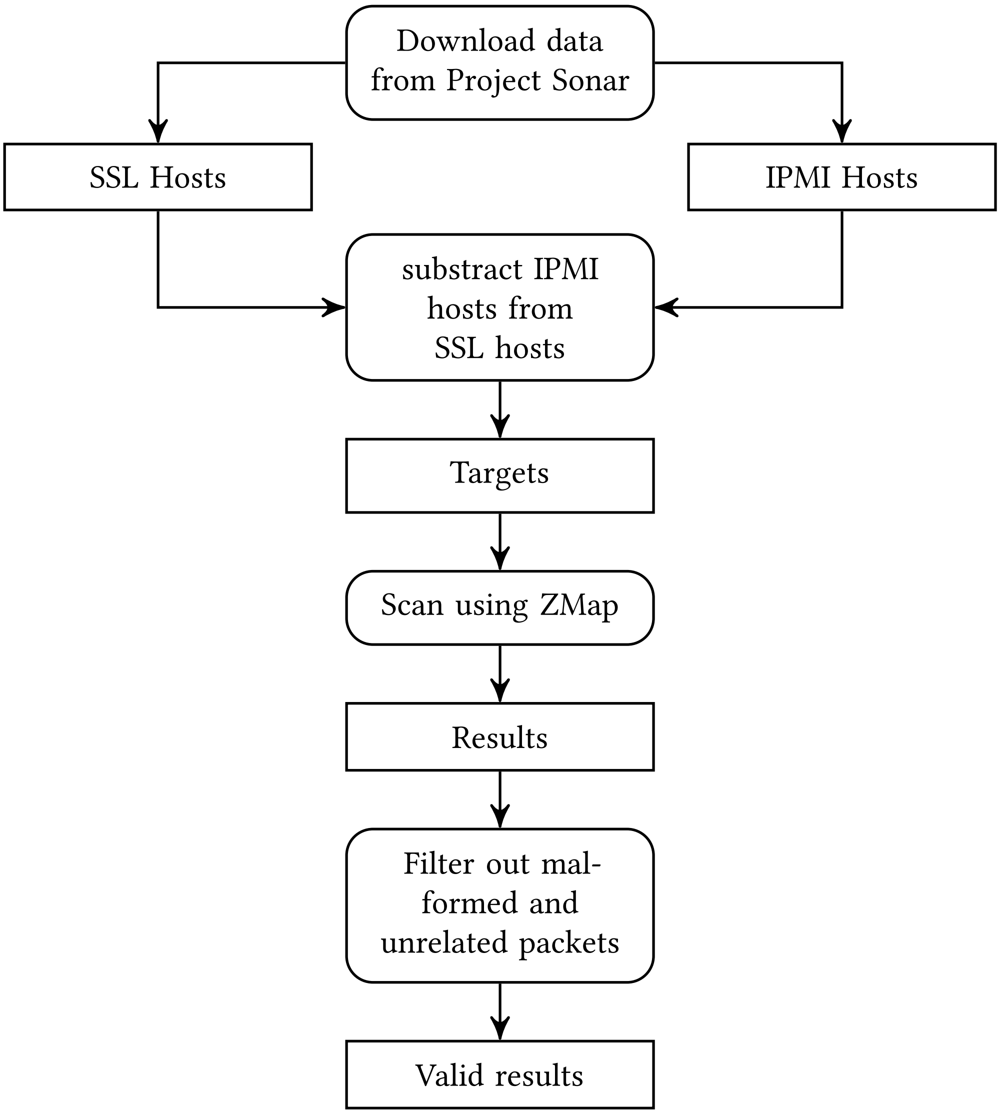

Download data from Project Sonar
SSL Hosts
IPMI Hosts
substract IPMI hosts from SSL hosts
Targets
Scan using ZMap
Results
Filter out mal-formed and unrelated packets
Valid results

Figure 7.2: Flow chart of discovering dark IPMI devices used in the first scan.

7.4.4 Ethical Considerations

Active network measurements have to be conducted in a sensible and ethical way in order not to induce negative consequences. Partridge and Allman [186] propose to evaluate whether the active measurements themselves or the release of the resulting data can harm an individual. Therefore we apply precautionary measures to reduce the impact of our IPMI scans.

First, we try to minimize the load on target networks generated by our research activities. Therefore, we do not scan with maximum speed but rather constrain our scans to 1 Mbit/s and 10 Mbit/s respectively. Additionally, ZMap's randomization feature ensures an even distribution of probes destined to a subnet over time.

Second, we use RMCP Ping requests which are less intrusive compared to Get Channel Authentication Capabilities requests. An RMCP Ping request does not attempt any authentication or login. No sensitive information other than the existence of the device itself is sent in the RMCP Pong response message. Furthermore, it does not show up in the IPMI device's log file, additionally reducing the number of false alarms.

Third, we set up a web site on the scanning machine with information about our research activities. Moreover, we provide a dedicated email address for information and blacklisting purposes. We offer network administrators the possibility to send their emails encrypted with our PGP key.

Fourth, we exclude IP addresses whose network administrator indicated in the past that they did not want to be scanned. Before our experiments the blacklist contained 148 entries resulting in 2 079 222 IP addresses. During our scans we received only two emails which were automatically sent by intrusion detection systems to the abuse contact listed in the WHOIS database. We answered both emails by providing more information with regard to our activity and offered the network administrators to put their IP ranges on our blacklist. We did, however, not receive a reply.

Fifth, we conduct an internal review at our Chair before starting any network experiment. This ensures that ethical and procedural concerns are addressed in advance and multiple viewpoints are being considered.

This research is conducted under consideration of the two ethical questions raised by Partridge and Allman [186]. We believe that the five precautionary measures ensure that the collection of data in this study does not cause tangible harm to any person's well-being. Furthermore, since we do not have plans to release the collected data, no private or confidential information is published. The results presented in this research give a general overview but no one specific individual or host is identified.

7.5 Evaluation

In this section we present the results from two IPMI scans conducted during this research. First, we give an overview of the two scans. Then we go into detail with regard to the responding hosts by comparing our measurement technique with the state of the art. Following, we evaluate deployment practices and uncover significant clustering in certain parts of the network. Subsequently, we evaluate the security of IPMI devices, specifically their TLS certificates and the supported IPMI versions. Finally, we classify

Table 7.1: Overview of both IPMI scans.

Scan	Scope	Targets	Resp.	Valid resp.	Hit rate	Scan rate	Duration
1	HTTPS hosts	33.1 M	38.2 k	37.2 k	0.11 %	1 Mbit/s	7:52 h
2	“0/0”	3.7 G	400.3 k	225.6 k	0.01 %	10 Mbit/s	2 d 21:05 h

the evaluated results and give concrete advice on hardening IPMI deployments in a network.

7.5.1 Scan Overview

We conducted two active measurement runs: the first was conducted on likely newly discoverable IPMI hosts, the second on the complete IPv4 address space. The IPMI deployment in the IPv6 Internet was out of scope in this research. Table 7.1 shows an overview with statistics about both scans.

The first scan's purpose was to gather additional active IPMI hosts compared to other IPMI scanning projects. In contrast to Project Sonar's regular IPMI scans [197] which use Get Channel Authentication Capabilities packets, we use RMCP Ping packets (see Section 7.3.4). This allows us to find IPMI devices which do not answer to Get Channel Authentication Capabilities requests. These dark devices have IPMI-over-IP deactivated, however other network interfaces (e. g. the web interface) might still be accessible. Therefore RMCP Ping requests allow us to estimate the number of accessible IPMI devices more accurately. We decide to scan all hosts with TLS certificates minus the IP addresses where an IPMI device has already been detected by Get Channel Authentication Capabilities scans. Thus the responding hosts are IPMI devices which could not be detected by Get Channel Authentication Capabilities scans because the IPMI-over-IP interface has been disabled. These devices do not pose a direct security risk as IPMI-over-IP is disabled. However, other access methods such as the web interface still pose a threat as shown by Bonkoski *et al.* [30].

The second scan covers the complete IPv4 address space. This allows us to find all publicly accessible IPMI devices in the IPv4 Internet and therefore gives us the complete picture.

For both scans we use the scanning tool ZMap [71] (see Section 7.4.3 for more details). We employ a blacklist to exclude hosts and subnets whose network administrators do not want to be scanned. Both scans were run from a physical machine on a mid-range server with a quad core Intel Core i7 CPU and 8 GiB RAM, with Debian 8 Jessie as its operating system.

7.5.2 Responding Hosts

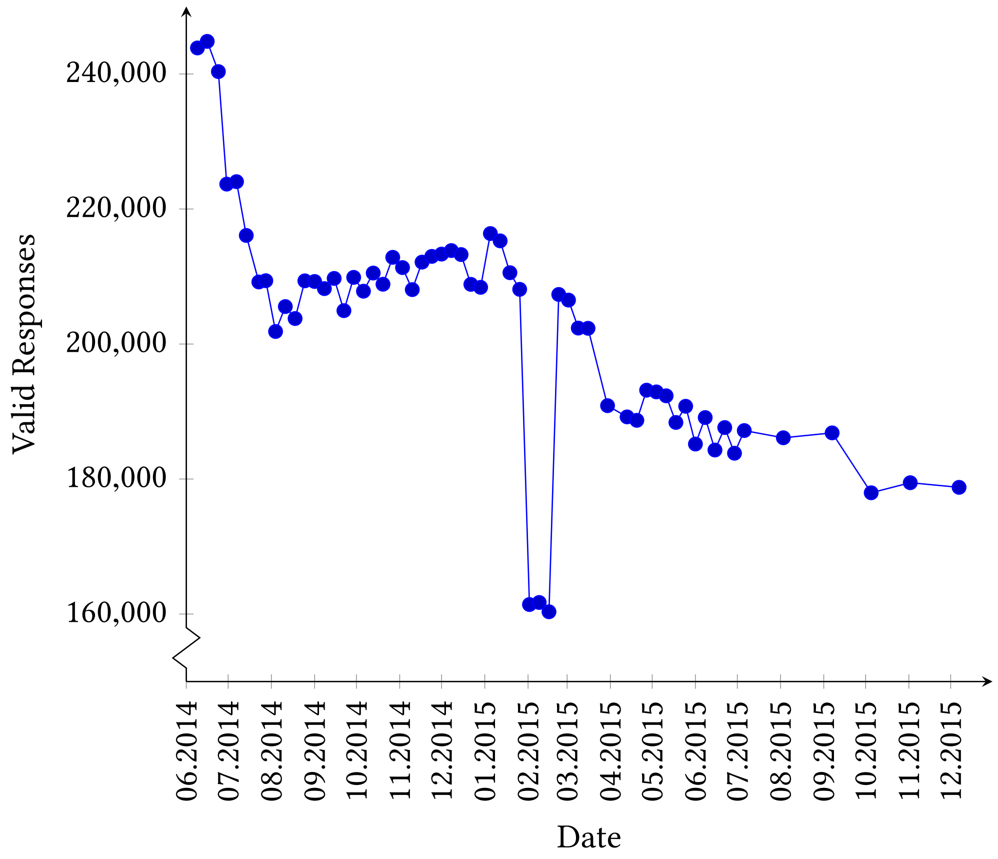


Figure 7.3: Number of valid responses from Project Sonar's IPMI datasets over time. Note the crunched y-axis for increased readability.

In our first scan targeting 33.1 M IP addresses we received replies from 38.2 k different IP addresses. After filtering out malformed and unrelated packets, 37.2 k valid responses remained. This corresponds to a hit rate of 0.11 %. Since we excluded the results of Project Sonar's IPMI scans from our target list, these responses come from dark IPMI devices, i. e., devices with IPMI-over-IP disabled.

Our second scan was conducted on the entire IPv4 address space minus our blacklist. We received replies from 400.3 k unique IP addresses with 225.6 k valid responses from IPMI devices. This corresponds to a hit rate of 0.01 %.

We compare our results with Project Sonar's IPMI scan from September 7, 2015. We remove 720 blacklisted IPs from Project Sonar's results to improve the comparability between both measurement campaigns. Our second scan delivered 21.2 % more results than Project Sonar's IPMI scan. This shows that our scanning approach is able to identify significantly more IPMI devices than other state of the art scanning methods.

When comparing our results to scans conducted by Dan Farmer in 2013 [86], we find slightly less devices than his 230 k. However, this can be explained by the general decrease of reachable IPMI devices over time. Figure 7.3 shows the number of valid IPMI responses obtained from Project Sonar's scans between June 2014 and December 2015 [1]. Note that the y-axis of the figure is crunched to increase readability. We can clearly see that the number of valid responses is dropping steadily, by a total of 65 k devices in the observation period. The decreasing number of IPMI host could be a result of previous work pointing out security risks with IPMI [30, 57, 261].

In consequence, network administrators might have isolated IPMI devices from the public Internet. Unfortunately, we could not compare our measurement method with Dan Farmer's as no detailed description was provided. Finally, Dan Farmer did not specify whether a blacklist was used during his scans.

7.5.3 Deployment Practices

In this section we evaluate the results of the second scan covering the complete IPv4 address space with regard to deployment practices. Specifically, we look at the question whether there are certain subnets with a significantly higher IPMI density compared to other less dense subnets.

To better visualize connected subnets but not constrain ourselves to a certain prefix length we use a Hilbert space-filling curve. Figure 7.4 shows the distribution of identified IPMI devices during the second scan. The figure shows a heat map of the IPMI deployment in the complete IPv4 address space. We visually highlight /8 networks to make it easier to find specific parts of the Internet. Each pixel represents one /18 network, the color indicates the number of IPMI devices found in this /18, ranging from blue (few IPMI devices) to red (many IPMI devices).

We can see that generally the IPv4 address space is sparsely populated with IPMI devices. This is no surprise and corresponds to the hit rate of 0.01 %. However, IPMI devices are not uniformly distributed over the IPv4 address space. They seem to be concentrated in some subnets whereas other subnets are completely blank indicating that no IPMI device is reachable from the public Internet. We suspect that the former could be stemming from data centers and hosting providers, whereas the latter would include private customers including DSL, cable, and fiber lines. To further analyze this scenario we take a look at parts of the Internet with a high IPMI density in more detail.

[1] Note that the strange valley between February and March 2015 in Figure 7.3 is most likely a measurement artifact at Rapid7, according to remarks by Rapid7 employees.

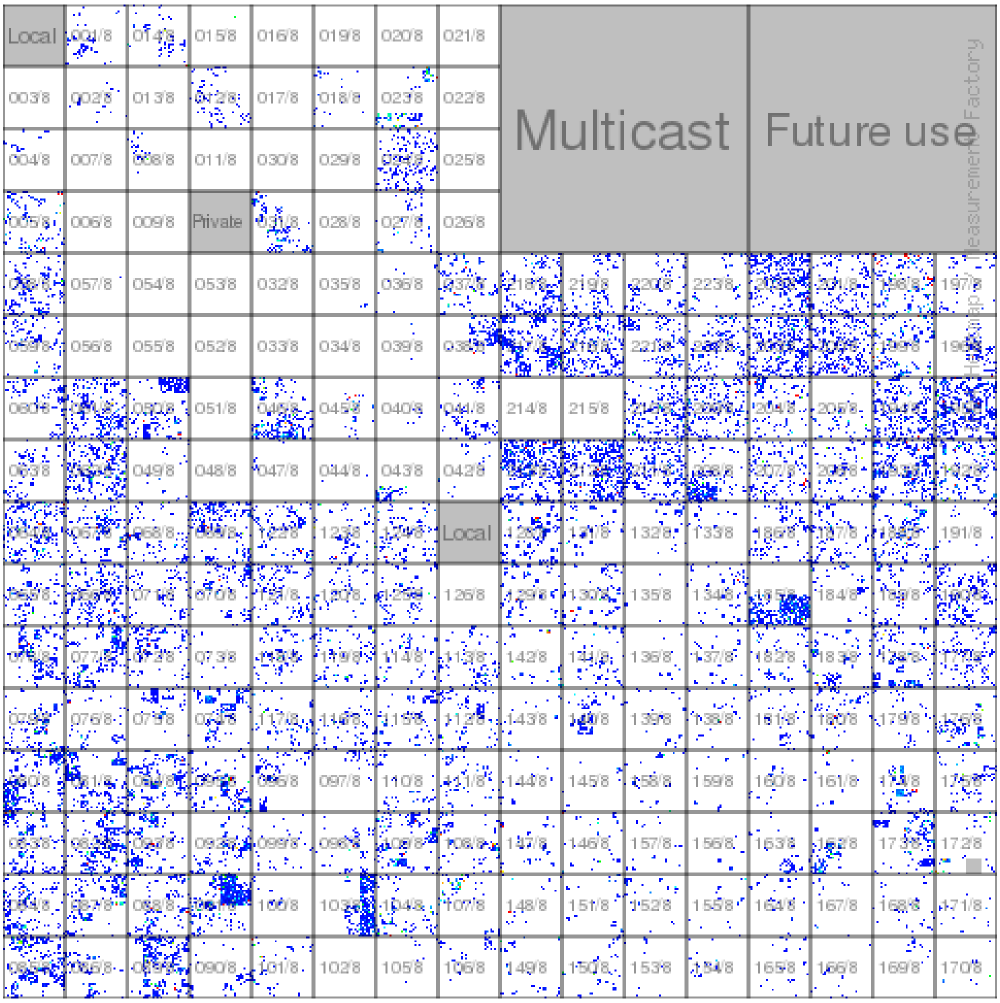


Figure 7.4: Hilbert curve of responding IPMI devices in IPv4-wide scan.

Thus we evaluate the density of IPMI devices based on Autonomous Systems (ASes).

We apply CAIDA's Prefix to AS mapping [39] to match IP addresses to their respective Autonomous Systems.[1] We find IPMI devices in a total of 7580 different ASes. Table 7.2 shows the top 10 ASes with the most IPMI devices. It is astounding that the top AS owned by NTT Communication hosts more than 30 k IPMI devices. NTT is a provider of network management solutions. The AS of NTT has more IPMI devices than the next eight ASes combined. Since NTT is one of the largest network services providers

[1] MOAS, i. e., prefixes originating from multiple ASes, are counted towards one AS only (according to CAIDA's deterministic sorting).

Table 7.2: Top 10 ASes with most IPMI devices.

Pos	ASN	AS	# IPMI Devices
1	2914	NTT-COMMUNICATIONS-2914	30 308
2	15003	NOBIS-TECH	5447
3	33781	OPQ	4687
4	16596	Univ. de Baja California	4140
5	5461	OKB MEI	3796
6	28227	NOVACIA	3281
7	35662	Redstation Limited	3132
8	60781	LeaseWeb-NL	2836
9	2607	SANET	2830
10	18978	Enzu Inc	2810

announcing numerous IPv4 prefixes (e. g. also for the Akamai CDN) and operating one single global AS this is not surprising.

The rest of the top 10 ASes is with three exceptions made up of hosting providers. Two entries are ASes pertaining to academic institutions from Mexico and Slovakia respectively. Interestingly, one AS is a special Russian agency ("Experimental Design Bureau“) for the purpose of developing aerospace and land-based antenna systems. For research purposes they also operate their own supercomputer.

As can be seen in Table 7.2 the number of IPMI devices per AS steeply decreases. In addition, Figure 7.4 further suggests that there are a few networks with many IPMI devices whereas most have little to none. To further investigate this phenomenon we plot the cumulative distribution function of IPMI devices in Autonomous Systems. Figure 7.5 shows the percentage of IPMI devices per AS. Note that the x-axis' scale is logarithmic as otherwise the function would almost immediately rise to the top due to its exponential increase. We see that the top 10 ASes are home to almost 30 % of the Internet's IPMI devices. As expected, the number of IPMI devices added per additional AS sharply decreases: The increase from 10 to 100 ASes adds about 30 % of IPMI devices, the same percentage as from AS 100 to AS 1000.

To conclude, IPMI devices are not uniformly distributed in the Internet, but rather concentrated in specific subnets (see Figure 7.4) and Autonomous Systems (see Table 7.2 and Figure 7.5). This hints at a general deployment issue with regard to IPMI: some network administrators and organizations do not seem to deem it necessary to secure their IPMI deployment which leaves them open to exploitation.

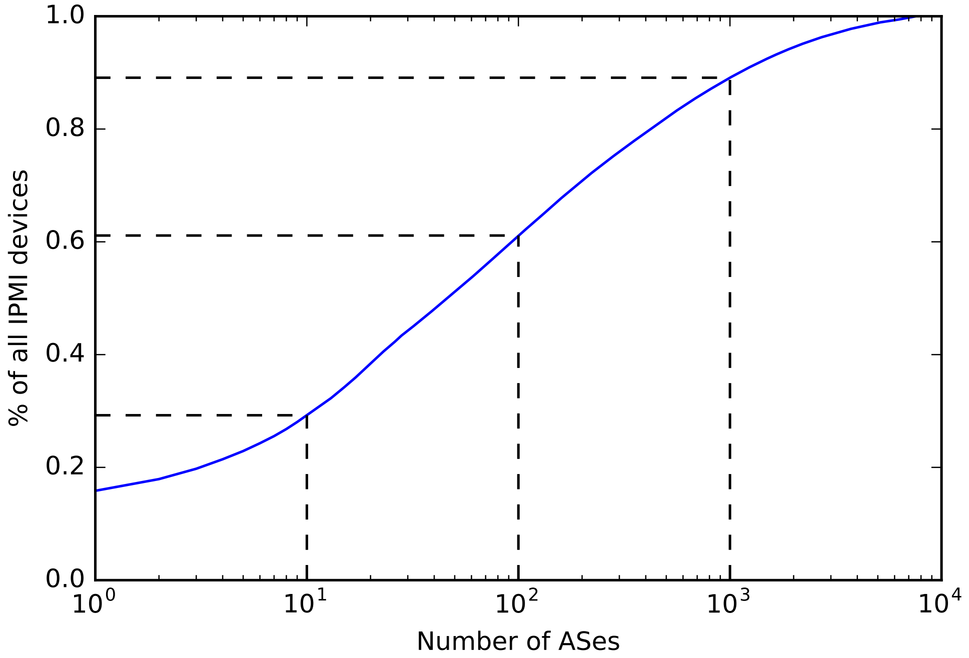


Figure 7.5: Distribution of IPMI devices in Autonomous Systems in the second scan. Note: The x-axis is log-scaled.

7.5.4 Security Analysis

We now investigate potential security issues posed by these publicly reachable IPMI devices.

TLS Evaluation

Since the target list of our first scan is based on TLS scan data published by Project Sonar, we are able to analyze the TLS certificates of the first scan's results.

First, we examine the use of default certificates in BMCs by considering the *Common Names* (CNs) and SHA1 checksums of the certificates. We find that 83.3 % of BMCs use default certificates, whereas 2.7 % of BMCs use custom generated certificates. A special case are 14.0 % of BMCs which seem to auto-generate their certificates upon installation. These exhibit the same CN schema but a differing SHA1 checksum in the certificates. Default certificates can be misused by extracting private keys from the firmware and therefore compromising the device's security [57, 252].

Additionally, we are able to determine the manufacturers of most IPMI devices by matching their certificate CN. Figure 7.6 shows the vendors of the discovered IPMI

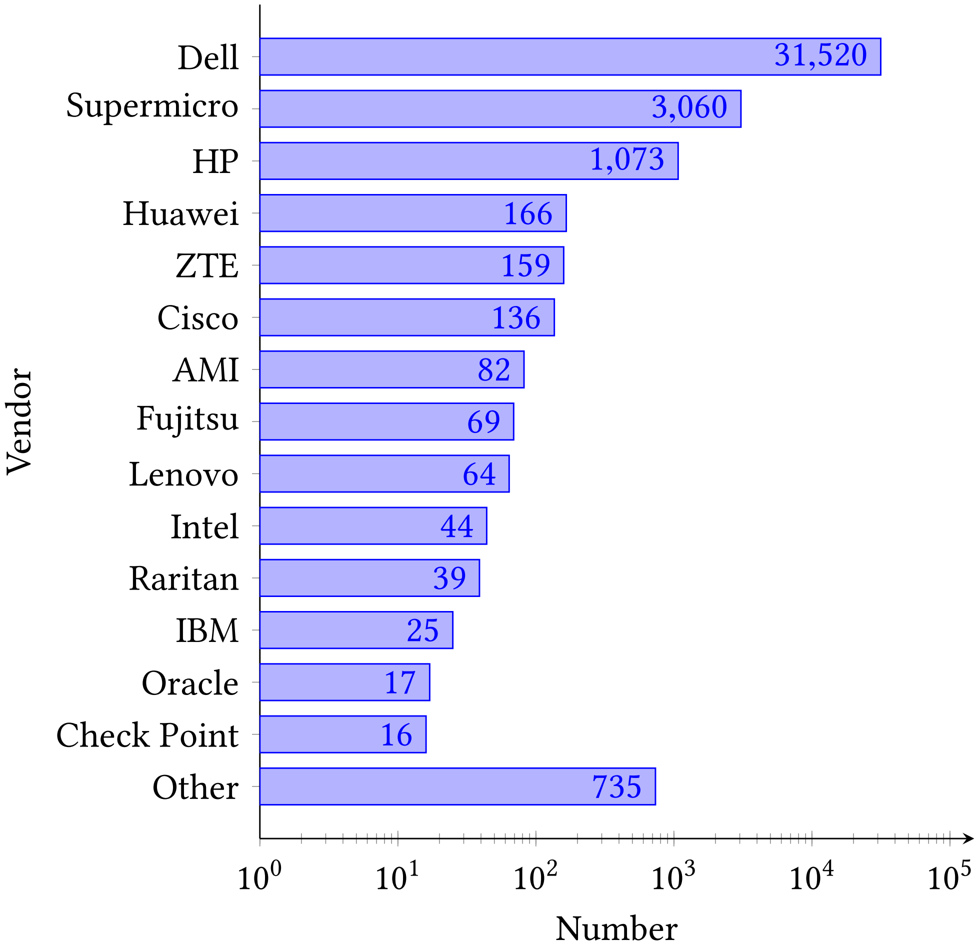


Figure 7.6: BMCs of the first scan by vendor. Note: The x-axis is log-scaled.

devices with TLS enabled during the first scan. More than 84 % of found BMCs have been manufactured by Dell, followed by Supermicro and HP.

We also investigate the number of certificates that are self-signed by building each host's trust chain. We find that 95.9 % of BMCs use self-signed certificates, while the certificates of 4.1 % BMCs are signed by issuer certificates. These issuer certificates, however, are mostly not trusted by modern browsers.

In addition we evaluate the key length of every BMC's client certificate. The results are shown in Figure 7.7 as a cumulative distribution function. Almost all (89.2 %) of keys are 1024 bit or shorter. This is not secure by today's standard as NIST proposes key lengths of at least 2048 bit [24].

Finally, we analyzed the key types of the BMCs' client certificates. This reveals that 99.3 % of certificates use RSA keys, only 0.7 % use DSA keys.

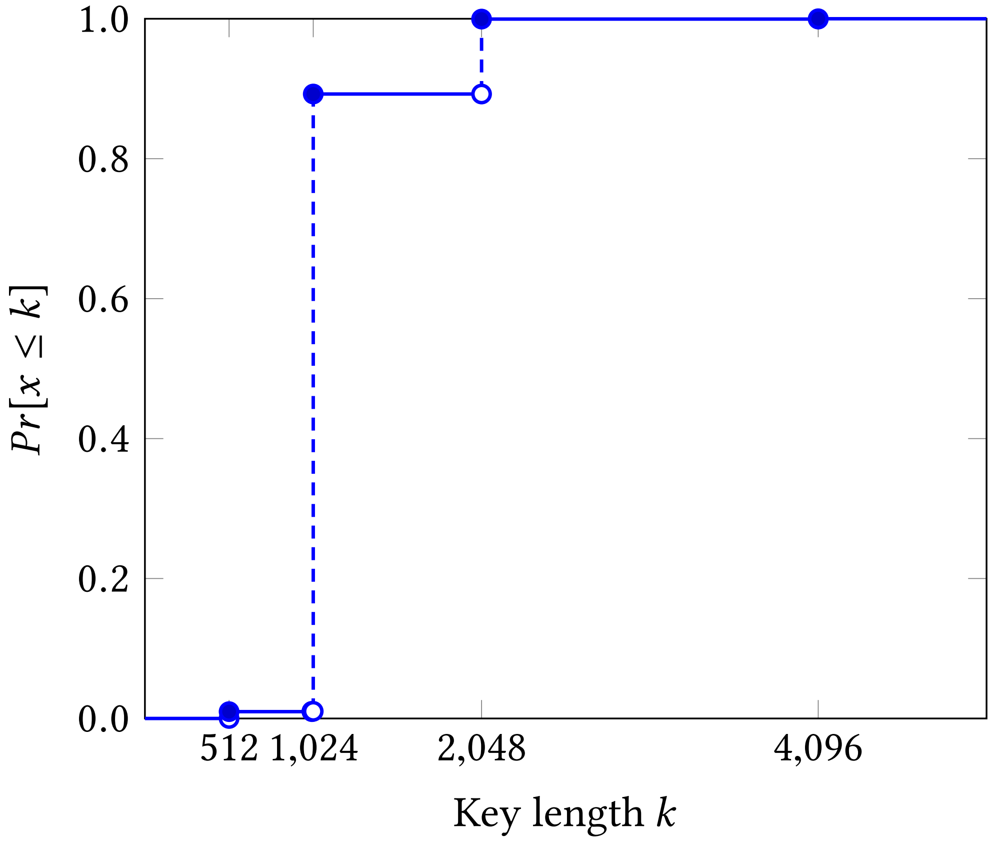


Figure 7.7: Distribution of TLS key lengths in the first scan.

IPMI Versions

RMCP Pong response packets like those gathered in our scans do not contain information about the supported IPMI version of the BMC. Therefore we use IPMI scan data provided by Project Sonar [197] to analyze supported IPMI versions. After filtering out malformed and unrelated packets, we find that about 63.0 % of IPMI devices support IPMI version 2.0 whereas the remaining 37.0 % only support IPMI version 1.5. This suggests that IPMI deployments are rarely updated since IPMI 2.0 was specified in 2004.

7.5.5 Best Practices

In this section we summarize our findings with regard to IPMI's operational security and give concrete advice to improve it.

Using the RMCP Ping discovery technique we find 21 % more devices than with conventional Get Channel Authentication Capabilities requests. After exploiting other weaknesses, e. g. in the SSH implementation [142], these additional devices can be used by actors with malicious intent to take over the device and then stage subsequent attacks. This is especially troubling when taking TLS certificate weaknesses into account. Since it is not enough to simply deactivate IPMI-over-IP on the BMC, we strongly recommend to block incoming requests on UDP/623 using an external firewall.

We found that IPMI devices are heavily clustered in subnets and ASes. Once an IPMI device has been discovered, an attacker can probe for more devices in its vicinity. This makes the scanning approach even stealthier, since not all IP addresses need to be probed to get a certain number of IPMI devices. Again, an outward facing firewall should block foreign IPMI traffic coming to the subnet or AS.

Most IPMI devices also offer access via an HTTPS web interface. This, however, introduces an additional attack vector since secure deployment of TLS is non-trivial. Our results show that most BMCs use default certificates included in the firmware and use very short keys. This makes it possible to perform various attacks preceded by Man-in-the-Middle attacks. We propose to use self-generated TLS certificates with strong keys. These certificates could be signed by a trusted party, e. g. the company's own certificate authority.

All of the above issues can be circumvented from being attacked by the outside if IPMI devices are only accessible within a VPN. This ensures that all interfaces and services (e. g. IPMI-over-IP, HTTPS web interface,...) are contained in a separate network. However, users with access to the VPN can still be a threat to these IPMI devices.

7.6 Conclusion

IPMI is the de facto protocol for out-of-band network management. Its ubiquitous use and the potential benefit from compromising an IPMI device makes it a prime attack target.

In this research we survey the current state of the IPMI deployment in the Internet. We present a method for finding dark IPMI devices. These devices have not been found using conventional methods. We then analyzed the distribution of IPMI devices in the network and found that they are heavily clustered. About one third of all devices are located in only 14 Autonomous Systems. The state of IPMI devices' TLS security is rather troubling as well. Most devices use default certificates included in their firmware and offer weak keys to TLS clients. Finally, we give concrete advice to increase the security of IPMI deployments in the network.

7.7 Key Contributions of This Chapter

This chapter addressed research question RQ V which includes challenges C 12 and C 13. We showed that IPMI devices could be reached from the public Internet. Using a novel scanning technique we identified IPMI devices more effectively. Subsequently, we analyzed the TLS security characteristics of found IPMI devices and found them to be in a dismal state. This makes many IPMI devices vulnerable to Man-in-the-Middle attacks.

In the following we list the key contributions of this chapter:

Effective IPMI device discovery Using a new scanning technique with RMCP Ping requests we were able to discover IPMI devices even if they had IPMI-over-IP disabled. This allowed us to see a more complete picture of the IPMI deployment in the Internet. We subsequently analyzed the IPMI deployment and found it to be clustered in certain ASes and prefixes. Additionally, the vast majority of IPMI we found were from Dell, showing that the IPMI population exhibits little diversity. With the novel scanning technique and results from this analysis we tackled challenge C 12.

Security of IPMI devices We conducted an in-depth TLS security analysis of all found IPMI devices. The devices lacked basic security characteristics by using default TLS certificates and weak keys. This allows attackers to stage Man-in-the-Middle attacks on these devices and assume complete control of the host device. With these results we tackled challenge C 13.

After tackling these two challenges we provide an answer to RQ V, as we find that the majority of IPMI devices lacks basic TLS security best practices and is therefore vulnerable to Man-in-the-Middle attacks.

7.8 Statement on Author's Contributions

This chapter is partially based on the publication “Digging for Dark IPMI Devices: Advancing BMC Detection and Evaluating Operational Security” by Oliver Gasser, Felix Emmert, and Georg Carle, which was published at TMA 2016 [98].

The author made the following contributions to the IPMI study presented in the TMA 2016 paper [98] and the adapted sections in this chapter. The IPMI deployment analysis work was first started as part of Felix Emmert's Bachelor's thesis (“Messung und Evaluation der Verbreitung von IPMI-Geräten mit aktiven Scans”) [81], which the author advised. The author coordinated and lead the group of researchers in this study. The author provided significant contributions in the study's design and execution. Moreover, the author contributed significantly in conducting the measurements and in analyzing the IPMI deployment. Finally, the author provided contributions to the security analysis.

In the following we describe changes between this chapter and the IPMI study presented in the TMA 2016 paper [98]. Most of the analyses in the paper are present in this dissertation. We shorten the analysis of the first scan, as this analysis only provides limited insights.

Part IV

Summary and Conclusion

Chapter 8

Conclusion

In this chapter we summarize our findings and show possible avenues for future work. First, we go into detail of each research question and draw conclusions from the outcome. Second, we show possible avenues for future research in the area of network measurements for Internet security.

8.1 Results From Research Questions

In the following section we present results from the research questions as described in detail in Section 1.2.

RQ I: How can we perform Internet-scale measurements in the IPv6 Internet?

The goal of RQ I was to find suitable measurement methodologies for Internet-wide measurements in the IPv6 Internet. While working on RQ I we tackled the following two challenges:

- C 1: Develop a tool to conduct Internet-scale port-based measurements in IPv6
- C 2: Develop a tool to conduct Internet-scale protocol-based measurements

We now present results for each tackled challenge.

Challenge 1

To tackle challenge 1 we needed to enhance the tool ZMap with IPv6 capabilities. We could not use an already available tool because they did either not provide support for IPv6 or were limited in terms of performance. We therefore extended the original

IPv4-only ZMap tool to make it IPv6 ready and created ZMapv6. In that effort we implemented IPv6 probing modules for TCP SYN, UDP, UDP DNS, and ICMPv6. These probing modules allowed us to cover a wide range of protocols, services, and measurement types. More details about ZMapv6 can be found in Section 3.1.2.

Challenge 2

To tackle challenge 2 we developed a multi-processing enhanced protocol-based measurement tool called goscanner. goscanner allowed us to perform large-scale protocol-based measurements (i. e., measurements with stateful protocol exchanges). goscanner's modular design in combination with its goroutine-based communication model ("Don't communicate by sharing memory; share memory by communicating" [109]) allowed us to perform large-scale scans of complex protocols such as TLS, HTTPS, and SSH. We make the goscanner tool publicly available [241]. For more details regarding goscanner we refer the reader to Section 3.2.

With overcoming these two challenges we can positively answer research question RQ I, as we are able to perform Internet-scale measurements in the IPv6 Internet using ZMapv6 and goscanner.

RQ II: How biased are address sources for IPv6 hitlists?

The objective of RQ II was to identify suitable targets of IPv6-based Internet measurements. While working on this research question we tackled the following four challenges:

- C 3: Extract targets for IPv6 hitlists from passive address sources
- C 4: Extract targets for IPv6 hitlists from active address sources
- C 5: Evaluate balancedness and biases of address sources for IPv6 hitlists
- C 6: Provide IPv6 Hitlist Service

We now present results for each tackled challenge.

Challenge 3

To tackle C 3, we analyzed two different types of passive data sources: (1) sampled flow data collected at a large European IXP and (2) packet data from the uplink of the Munich Scientific Network. In this analysis we identified clearly visible weekly patterns and evaluated the time-to-measurements. Due to most of the addresses belonging to clients, we found that measurements need to be conducted within minutes to ensure a high response rate. Additionally, we evaluated the Hamming weight distribution of interface identifiers (i. e., the last 64 bit in each IPv6 address) and found many privacy extension addresses in these passive datasets. More details about passive IPv6 hitlist data sources can be found in Section 4.2.

Challenge 4

We analyzed IPv6 addresses from a multitude of active sources to resolve challenge C 4. We evaluated the runup of each address source over time and identified that certain sources (e. g. domain lists and Certificate Transparency) provide far more addresses compared to others (e. g. DNS AXFR). This imbalance needs to be considered when aggregating different address sources. To learn more about the intricacies of active IPv6 hitlist sources we refer the reader to Section 4.3.

Challenge 5

To tackle challenge C 5 we conduct an in-depth analysis of IPv6 hitlist sources to identify biases. We find that the Autonomous System (AS) and prefix distributions are differing quite heavily between sources: Domain lists and Certificate Transparency are dominated by few ASes and prefixes whereas the address sets from Bitnodes and traceroutes are more balanced. While investigating this issue we uncovered prefixes where each possible IPv6 addresses (even if chosen randomly) would answer to our queries. Since these prefixes heavily bias any measurements, we devised an algorithm to detect these aliased prefixes. We regularly publish aliased prefixes and responsive IPv6 addresses for the measurement community as part of our in our IPv6 Hitlist Service. Finally, we also showed differences for address learning algorithms and evaluated longitudinal responsiveness based on protocol and port. For more details regarding biases in IPv6 hitlist we refer the reader to Section 4.3.

Challenge 6

To resolve challenge C 6 we developed the IPv6 Hitlist Service [103]. With this service we procure easy access to up-to-date measurement data which can be used in security or other types of research. The IPv6 Hitlist Service provides daily IPv6 measurement results and an updated list of aliased prefixes. To identify trends in IPv6 responsiveness we provide automatically updated graphs on the website. More than two dozens researchers have been granted access to the IPv6 Hitlist Service's raw measurement result data. For more details on the IPv6 Hitlist Service we refer the reader to Section 4.4.

With overcoming these four challenges we can provide an answer to research question RQ II, as we find different levels of imbalanced and biased sources.

RQ III: Are HTTPS servers still vulnerable to Man-in-the-Middle attacks?

Research question RQ III's goal was to evaluate the security of the HTTPS ecosystem with regard to Man-in-the-Middle attacks. During our work on this research question we tackled the following two challenges:

- C 7: Evaluate deployment of HTTPS security extensions
- C 8: Evaluate security of certificates in Certificate Transparency logs

We now present results for each challenge.

Challenge 7

To tackle challenge C 7 we performed multiple Internet-wide measurements for HTTPS servers. In this dissertation we focused our evaluation on two HTTP headers—HTTP Strict Transport Security (HSTS) and HTTP Public Key Pinning (HPKP). We found that hosts behaved very consistently even when hosted on multiple IPv4 or IPv6 addresses. For the security extensions themselves, however, we see quite some differences in deployment: HSTS is much more widely deployed compared to HPKP. We compared the deployment of these two headers to other HTTPS security extensions (e. g. Certificate Transparency) and determined that risk and ease of deployment play an important role in successful deployment of new security standards. After our study, Google announced that they would drop support for HPKP due to its lack of adoption, confirming our findings regarding the correlation between risk, deployment effort, and effective deployment. More details about the deployment of HTTPS security extensions can be found in Section 5.1.

Challenge 8

We overcame C 8 by performing an in-depth analysis of the Certificate Transparency (CT) ecosystem and the TLS certificates contained in CT logs. We showed that the CT ecosystem saw a large increase in participation just before the April 2018 deadline, after which Google's Chrome browser only accepted newly issued certificates if they were logged in CT logs. Moreover, we found that a few thousand certificates in CT logs exhibited severe security weaknesses (e. g. weak signature algorithms such as SHA1 or short TLS keys). By correlating the development of weak certificates over time, we showed that the situation was significantly improving when stricter guidelines from the CA or browser community were passed. For an in-depth analysis of the CT ecosystem we refer the reader to Section 5.2.

After tackling these two challenges we can provide an answer to RQ III, as we find that there are still significant numbers of HTTPS servers with lacking HTTPS security extensions and weak certificates which makes them vulnerable to Man-in-the-Middle attacks.

RQ IV: Are BACnet devices vulnerable to amplification attacks?

The goal of research question RQ IV was to evaluate the vulnerability of the building automation protocol BACnet to amplification attacks. During our work on this research question we tackled the following three challenges:

- C 9: Evaluate deployment of publicly reachable BACnet devices
- C 10: Evaluate amplification attack potential of BACnet devices
- C 11: Evaluate impact of notification campaign targeting BACnet devices

We now present results for each tackled challenge.

Challenge 9

We tackled challenge C 9 by performing multiple Internet-wide measurements for BACnet devices and by conducting a thorough analysis of the BACnet ecosystem. We found more than 16 k publicly reachable BACnet devices. These devices are clustered in geographical locations (mostly USA and Canada), network prefixes, and ASes. Moreover, the majority of devices are only from a handful of manufacturers. The analysis of the BACnet ecosystem laid the groundwork for further security analysis of the BACnet protocol. More specifics about the BACnet ecosystem can be read in Section 6.4.

Challenge 10

To provide an answer to C 10 we performed a theoretical analysis for the eligibility of the BACnet protocol for amplification attacks. After confirming that BACnet devices could indeed by exploited as amplifiers, we conducted several measurements to quantify the severity of such attacks by calculating the amplification factor. We found that BACnet devices offered a similar amplification factor as DNS-based amplifiers. Consequently, we demonstrated that BACnet devices could indeed be misused as amplifiers in amplification attacks. For further details on BACnet's vulnerability to amplification attacks we refer the reader to Section 6.5.

Challenge 11

To solve challenge C 11 we conducted a notification campaign in cooperation with a large European CERT, which notified affected parties of the amplification attack danger posed by publicly reachable BACnet devices. After our notification we saw a reduction in the number of publicly reachable BACnet devices, which confirmed that notification campaigns can drive measurable impact. Details on BACnet's notification campaign can be found in Section 6.7.

After tackling these three challenges we can provide an answer to RQ IV, as we find that there are publicly reachable BACnet devices which can be misused as amplifiers in amplification attacks. We can therefore positively answer this research question as BACnet devices are indeed vulnerable to amplification attacks.

RQ V: Are IPMI devices vulnerable to Man-in-the-Middle attacks?

With research question RQ V we wanted to evaluate the TLS security of the deployment of IPMI out-of-band management devices with regard to Man-in-the-Middle attacks. During our work on this research question we tackled the following two challenges:

- C 12: Evaluate deployment of publicly reachable IPMI devices
- C 13: Evaluate TLS security of IPMI devices

We now present results for each tackled challenge.

Challenge 12

We tackled challenge C 12 by developing a novel IPMI measurement technique with RMCP Ping requests and evaluating the IPMI deployment. This RMCP Ping technique identified IPMI devices even if they had IPMI-over-IP disabled. With this technique we drew a more accurate picture of the Internet's publicly reachable IPMI device deployment. Similar to the BACnet deployment, we found the IPMI deployment to be topologically clustered in certain prefixes and ASes. To learn more about IPMI's deployment in the public Internet see Sections 7.5.2 and 7.5.3.

Challenge 13

In addition to evaluating the IPMI deployment, we analyzed the security of IPMI devices in order to tackle challenge C 13. To this end we conducted an in-depth TLS security analysis of all found IPMI devices and found that many devices lacked basic security characteristics (e. g. weak keys or usage of vendor-default TLS certificates). Insecure IPMI devices allow attackers to completely control their hosts (e. g. restarting machines, changing the default boot device, eavesdropping on network traffic) using Man-in-the-Middle attacks. With this analysis we concluded that the security of publicly reachable IPMI devices was sub-par at best. For more details on the lack of security of publicly reachable IPMI devices we refer the reader to Section 7.5.4.

After tackling these two challenges we provide an answer to RQ V, as we find that the majority of IPMI devices lacks basic TLS security best practices and is therefore vulnerable to Man-in-the-Middle attacks.

8.2 Future Work

Even though a number of contributions towards making Internet measurements more effective for uncovering insecure devices and services were presented in this dissertation, there are still some open avenues for future research.

In the future we want to further integrate our developed scanning tool ZMapv6 with the detection of aliased prefixes. Aliased prefixes could be detected as part of ZMapv6 and we could immediately blacklist this prefix. This would allow to merge several measurement runs (aliased prefix detection at several levels before running the actual measurement run) into a single ZMapv6 invocation.

Another possibility would be to directly leverage IPv6 address learning techniques such as Entropy/IP or 6Gen in ZMapv6. Making use of these learning techniques ZMapv6 could send probes to additional addresses in prefixes with similar address allocation schemes as have been seen before.

Additionally, it would be desirable to add further protocols and measurement types to the IPv6 Hitlist Service to make it even more useful to fellow researchers.

With the continuing pervasion of Internet-of-Things (IoT) devices in our everyday lives [198], Internet researchers are faced with more and more Internet-connected devices. Due to their relatively long product lifespan of five, ten, or even more years (compared to e. g. two years average lifetime of smartphones in the US [82]) IoT devices pose new challenges. One of these challenges is update management (i. e., vendors ensuring that devices are kept up-to-date with regard to newly discovered security vulnerabilities) or the lack thereof. Devices lacking security patches are a valuable asset for attackers and botnet owners. These actors with malicious intent can exploit these vulnerabilities to get access to these IoT devices. Therefore it will become even more important to continuously monitor IoT devices, e. g. using active Internet measurements. Universities with researchers can be the independent body to tackle this task, instead of leaving it to government agencies [182].

Another aspect of these IoT devices will be their support of the IPv6 protocol. As some of these protocols (such as BACnet) sees almost no IPv6 deployment, due to the exhaustion of available IPv4 addresses this is bound to change. The support of an additional network protocol not only increases the attack surface, but also increases the security configuration management to ensure that devices can be accessed only by permitted parties. As has been shown, IPv6 security configurations are lacking compared to their IPv4 counterpart [59]. The adoption of IPv6 in the IoT world means that IPv6 measurements will further gain in importance and become even more essential to get an accurate picture of the Internet and identify security threats.

The dual-stack usage of IPv4 and IPv6 might be prevalent for a while until IPv6 is adopted by a critical mass of devices and services. Once this breaking point is reached and companies start to move to IPv6-only, the IPv4 Internet sees the threat of being abandoned. In turn, IPv4 deployments and configurations might lack updates and risk being out-of-date. This means that in the future we could see a reverse scenario as we see today [59], where the IPv6 Internet is better configured and more secure than its legacy counterpart. The IPv4 Internet could, however, still be used to stage attacks and must therefore remain an important target for security researchers.

Finally, to cope with the increasingly fast changing nature of the Internet (e. g. due to the increased usage of IPv6 privacy extensions), measurements need to be performed at a higher frequency as well. These high-frequent measurement campaigns pose new challenges to Internet researchers, as software and hardware needs to be adapted to perform measurements at 100+ Gbit/s without the loss of accuracy due to e. g. packet drops or ICMP throttling at routers.

Part V

Appendix

Chapter A

goscanner Project Interdependency

Figure A.1 visualizes goscanner's interdependencies. The most important interdependencies are visualized in Figure 3.2 in Section 3.2.3. Due to its large size, the files in the `unix` package are capped.

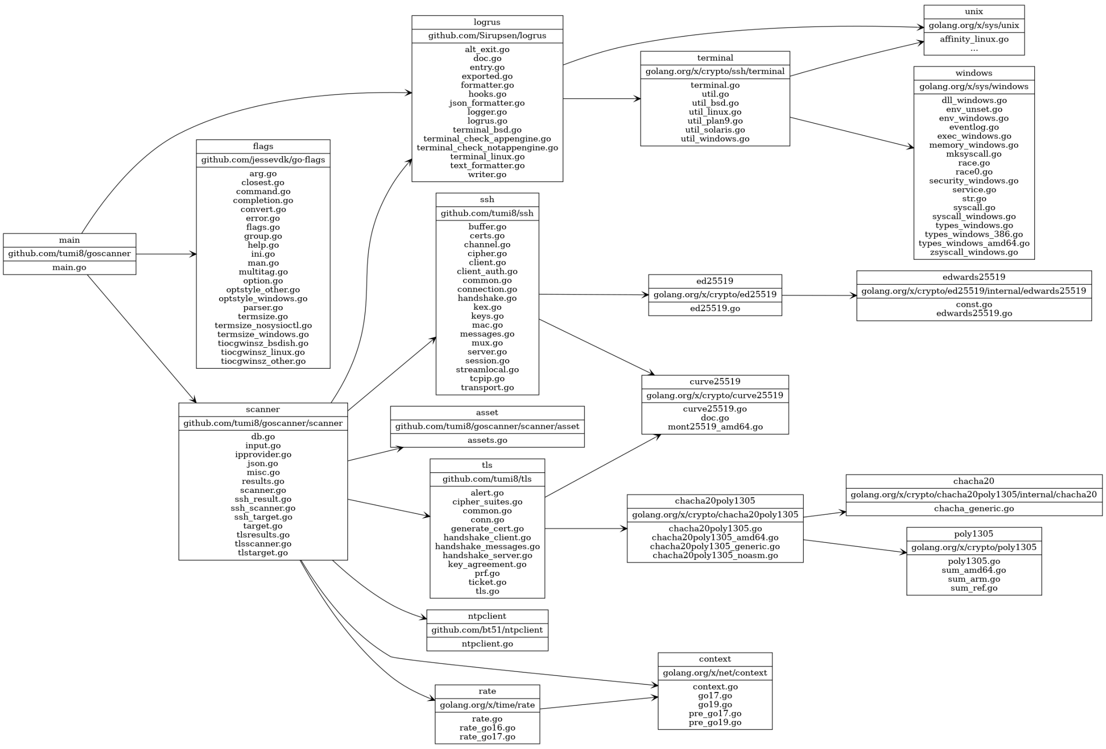


Figure A.1: Visualization of all goscanner interdependencies, created using goviz [205].

Chapter B

List of Publications

In the following we list all publications published during this research. Publications are grouped based on their affiliation to dissertation topics.

B.1 IPv6 Hitlists and Measurements

- *Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczynski, Stephen D. Strowes, Luuk Hendriks, Georg Carle*, “Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists” in Proceedings of the 2018 ACM Internet Measurement Conference [100].
- *Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, Georg Carle*, “Scanning the IPv6 Internet: Towards a Comprehensive Hitlist”, in Proceedings of the 2016 International Workshop on Traffic Monitoring and Analysis [104].

B.2 HTTPS and Certificate Transparency Measurements

- *Quirin Scheitle, Oliver Gasser, Theodor Nolte, Johanna Amann, Lexi Brent, Georg Carle, Ralph Holz, Thomas C. Schmidt, Matthias Wählisch*, “The Rise of Certificate Transparency and Its Implications on the Internet Ecosystem” in Proceedings of the 2018 ACM Internet Measurement Conference [218].
- *Oliver Gasser, Benjamin Hof, Max Helm, Maciej Korczynski, Ralph Holz, Georg Carle*, “In Log We Trust: Revealing Poor Security Practices with Certificate Transparency Logs and Internet Measurements”, in Proceedings of the 2018 Passive and Active Measurement Conference, **Best Paper Award** [101].

- *Johanna Amann, Oliver Gasser, Quirin Scheitle, Lexi Brent, Georg Carle, Ralph Holz*, "Mission Accomplished? HTTPS Security after DigiNotar", in Proceedings of the 2017 Internet Measurement Conference, **Community Contribution Award, IRTF Applied Networking Research Prize** [16].

B.3 BACnet Measurements

- *Oliver Gasser, Quirin Scheitle, Benedikt Rudolph, Carl Denis, Nadja Schricker, Georg Carle*, "The Amplification Threat Posed by Publicly Reachable BACnet Devices", in Journal of Cyber Security and Mobility 2017 [106].
- *Oliver Gasser, Quirin Scheitle, Carl Denis, Nadja Schricker, Georg Carle*, "Security Implications of Publicly Reachable Building Automation Systems", in Proceedings of the 2017 International Workshop on Traffic Measurements for Cybersecurity [105].
- *Oliver Gasser, Quirin Scheitle, Carl Denis, Nadja Schricker, Georg Carle*, "Öffentlich erreichbare Gebäudeautomatisierung: Amplification-Anfälligkeit von BACnet und Deployment-Analyse im Internet und DFN", in 24. DFN-Konferenz Sicherheit in vernetzten Systemen (2017) [107].

B.4 IPMI Measurements

- *Oliver Gasser, Felix Emmert, Georg Carle*, "Digging for Dark IPMI Devices: Advancing BMC Detection and Evaluating Operational Security", in Proceedings of the 2016 International Workshop on Traffic Monitoring and Analysis [98].

B.5 Other Publications

- *Quirin Scheitle, Taejoong Chung, Jens Hiller, Oliver Gasser, Johannes Naab, Roland van Rijswijk-Deij, Oliver Hohlfeld, Ralph Holz, Dave Choffnes, Alan Mislove, Georg Carle*, "A First Look at Certification Authority Authorization (CAA)", in ACM SIGCOMM Computer Communications Review [215].
- *Patricia Callejo, Connor Kelton, Narseo Vallina-Rodriguez, Rubén Cuevas, Oliver Gasser, Christian Kreibich, Florian Wohlfart, Ángel Cuevas*, "Opportunities and Challenges of Ad-based Measurements from the Edge of the Network", in Proceedings of the 16th ACM Workshop on Hot Topics in Networks [41].

- *Quirin Scheitle, Matthias Wählisch, Oliver Gasser, Thomas C. Schmidt, Georg Carle*, “Towards an Ecosystem for Reproducible Research in Computer Networking”, in ACM SIGCOMM Reproducibility Workshop 2017 [219].
- *Quirin Scheitle, Oliver Gasser, Minoo Rouhi, Georg Carle*, “Large-Scale Classification of IPv6-IPv4 Siblings with Variable Clock Skew”, in Proceedings of the 2017 Network Traffic Measurement and Analysis Conference [217].
- *Quirin Scheitle, Oliver Gasser, Patrick Sattler, Georg Carle*, “HLOC: Hints-Based Geolocation Leveraging Multiple Measurement Frameworks”, in Proceedings of the 2017 Network Traffic Measurement and Analysis Conference [216].
- *Timm Böttger, Lothar Braun, Oliver Gasser, Felix von Eye, Helmut Reiser, Georg Carle*, “DoS Amplification Attacks – Protocol-Agnostic Detection of Service Abuse in Amplifier Networks”, in Proceedings of the 2015 International Workshop on Traffic Monitoring and Analysis [37].
- *Johann Schlamp, Ralph Holz, Oliver Gasser, Andreas Korsten, Quentin Jacquemart, Georg Carle, Ernst W. Biersack*, “Investigating the Nature of Routing Anomalies: Closing in on Subprefix Hijacking Attacks”, in Proceeding of the 2015 International Workshop on Traffic Monitoring and Analysis [220].
- *Felix von Eye, Timm Böttger, Helmut Reiser, Lothar Braun, Oliver Gasser, Georg Carle*, “Detektion und Prävention von Denial-of-Service Amplification Attacken – Schutz des Netzes aus Sicht eines Amplifiers”, in 22. DFN-Konferenz Sicherheit in vernetzten Systemen [84].
- *Oliver Gasser, Ralph Holz, Georg Carle*, “A deeper understanding of SSH: results from Internet-wide scans”, in Proceedings of the 14th Network Operations and Management Symposium (2014) [99].

Chapter C

List of Student Theses

In the following we list all student theses advised by the author of this dissertation during this research under the supervision of Prof. Dr.-Ing. Georg Carle. Student theses are grouped based on their affiliation to dissertation topics. The primary advisors of each student thesis are denoted by an asterisk (*).

C.1 Network Measurement Methodology

- *Nils Mäurer*, "Efficient scans in a research network", Bachelor's thesis advised by Ralph Holz*, Oliver Gasser* [177].
- *Matthias Jaros*, "Distribution and Orchestration of Network Measurements on the Planetlab testbed", Bachelor's thesis advised by Ralph Holz*, Oliver Gasser* [141].
- *Zhechko Zhechev*, "Asymmetric Route Detection using Return TTLs", Bachelor's thesis advised by Oliver Gasser*, Quirin Scheitle [263].
- *Dmitry Chokovski*, "Is IPv6 faster than IPv4", Bachelor's thesis advised by Paul Emmerich*, Quirin Scheitle, Oliver Gasser [47].
- *Markus Sosnowski*, "Internet-Wide Assessment of TCP Options", Bachelor's thesis advised by Quirin Scheitle*, Oliver Gasser*, Minoo Rouhi, Paul Emmerich, Dominik Scholz [228].
- *Jan-Philipp Lauinger*, "Evaluating Client Discrimination in Anonymization Networks Using Active Network Scans", Forschungspraxis advised by Oliver Gasser*, Sree Harsha Totakura [155].

C.2 IPv6 Hitlists and Measurements

- *Sebastian Gebhard*, "IPv6 Scanning - Smart address selection and comparison to legacy IP", Master's thesis advised by Oliver Gasser*, Quirin Scheitle* [108].

C.3 HTTPS and TLS Measurements

- *Felix Beil*, "Long Term Analysis of HTTP Strict Transport Security", Bachelor's thesis advised by Quirin Scheitle*, Oliver Gasser [25].
- *Pirmin Blanz*, "IPv6 TLS Security Scanning", Master's thesis advised by Oliver Gasser*, Quirin Scheitle* [28].
- *Tobias Brunnwieser*, "A Framework for Detection and Analysis of HTTPS Interception ", Master's thesis advised by Oliver Gasser*, Sree Harsha Totakura, Florian Wohlfart [34].
- *Max Helm*, "Evaluating TLS Certificate Transparency Logs using Active Scans", Interdisciplinary project advised by Oliver Gasser*, Benjamin Hof [120].
- *Max Helm*, "Traceable Measurement Result Publication in Append-only Ledgers", Master's thesis advised by Oliver Gasser*, Benjamin Hof, Quirin Scheitle [121].

C.4 BACnet Measurements

- *Nadja Schricker*, "Active Security Evaluation with Network Scans", Bachelor's thesis advised by Oliver Gasser*, Quirin Scheitle [222].

C.5 IPMI Measurements

- *Felix Emmert*, "Messung und Evaluation der Verbreitung von IPMI-Geräten mit aktiven Scans", Bachelor's thesis advised by Oliver Gasser* [81].

C.6 Other Student Theses

- *Timm Böttger*, "Detection of Amplification Attacks in Amplifier Networks", Master's thesis advised by Oliver Gasser*, Lothar Braun*, Felix von Eye [36].
- *Michael Köpferl*, "Effective Visualization of Amplification Attacks in Amplifier Networks", Bachelor's thesis advised by Oliver Gasser*, Felix von Eye [152].

- *Michael Köpferl*, “Evaluation of amplification attacks in large-scale networks to improve detection performance”, Interdisciplinary project advised by Oliver Gasser*, Stefan Metzger [153].
- *Albert Khakimullin*, “Real-time Amplification Attack Detection”, Interdisciplinary project advised by Oliver Gasser*, Felix von Eye [145].
- *Johannes Naab*, “Scanning and Evaluating DNS Deployments in the Internet”, Master’s thesis **note** [178].
- *Michael Domke*, “DNS-in-a-Box: A Testing Framework for Reproducible Network Measurements”, Bachelor’s thesis advised by Johannes Naab*, Oliver Gasser [69].
- *Frank Schmidt*, “Evaluation of Trust Relationships in the Domain Name System”, Master’s thesis advised by Johannes Naab*, Oliver Gasser [221].
- *Hendrik Eichner*, “Revisiting SSH Security in the Internet”, Bachelor’s thesis advised by Oliver Gasser*, Minoo Rouhi [78].
- *Fabian Raab*, “Modeling and Analysis of BGP Community Attributes”, Bachelor’s thesis advised by Oliver Gasser*, Johann Schlamp* [196].
- *Jonas Heintzenberg*, “Browser-based measurement of NAT middleboxes”, Bachelor’s thesis advised by Florian Wohlfart*, Oliver Gasser [119].
- *Johannes Fischer*, “Browser-based Internet Connection Testing”, Master’s thesis advised by Florian Wohlfart*, Oliver Gasser [94].
- *Michael Brunner*, “Multi-protocol Path Topology Evaluation”, Bachelor’s thesis advised by Minoo Rouhi*, Dominik Scholz*, Oliver Gasser [33].
- *Katharina Wiegräbe*, “Identifying Web-enabled Devices on Internet Paths”, Bachelor’s thesis advised by Minoo Rouhi*, Dominik Scholz*, Oliver Gasser, Quirin Scheitle [256].
- *Christian Sturm*, “Detection of malicious packets through inbound Time to Live headers”, Bachelor’s thesis advised by Quirin Scheitle*, Oliver Gasser*, Felix von Eye [233].
- *Arno Hilke*, “IP Spoofing Detection Through Time to Live Header Analysis”, Bachelor’s thesis advised by Quirin Scheitle*, Oliver Gasser* [125].
- *Till Wickenheiser*, “Correlation of TTL data to network characteristics”, Bachelor’s thesis advised by Quirin Scheitle*, Oliver Gasser* [254].

- *Paulin Tchonin*, “TTL Analysis for DDoS Defense”, Master’s thesis advised by Quirin Scheitle*, Oliver Gasser*, Paul Emmerich [235].
- *Patrick Sattler*, “Parsing geographical locations from DNS names”, Bachelor’s thesis advised by Quirin Scheitle*, Oliver Gasser*, Johannes Naab [212].
- *Patrick Sattler*, “Parsing geographical locations from DNS names”, Guided research advised by Quirin Scheitle*, Oliver Gasser* [213].
- *Patrick Sattler*, “Parsing geographical locations from DNS names”, Interdisciplinary project advised by Quirin Scheitle*, Oliver Gasser* [214].
- *Victor Sosa*, “Large-Scale Flow Collection”, Interdisciplinary project advised by Oliver Gasser* [227].
- *Katharina Rudert*, “Botnet Detection using Machine Learning on Flow Data”, Master’s thesis advised by Johannes Naab*, Oliver Gasser* [209].
- *Minoo Rouhi Vejdani*, “Path tracing and validation of IPv4 and IPv6 siblings”, Master’s thesis advised by Quirin Scheitle*, Oliver Gasser*, Paul Emmerich [208].
- *Alexander Schulz*, “Identification of IPv6-IPv4 Sibling Pairs from Passive Observations”, Bachelor’s thesis advised by Quirin Scheitle*, Oliver Gasser*, Minoo Rouhi [223].
- *Samy El Deib*, “Detecting IPv6-IPv4 Sibling Pairs Based on few Data Points”, Bachelor’s thesis advised by Quirin Scheitle*, Oliver Gasser*, Minoo Rouhi [79].

Chapter D

List of Figures

Chapter E

List of Tables

Bibliography

[1] Mustafa Emre Acer, Emily Stark, Adrienne Porter Felt, Sascha Fahl, Radhika Bhargava, Bhanu Dev, Matt Braithwaite, Ryan Sleevi, and Parisa Tabriz. „Where the Wild Warnings Are: Root Causes of Chrome HTTPS Certificate Errors“. In: *Proceedings of the 2017 Conference on Computer and Communications Security.* ACM. Dallas, TX, USA, Nov. 2017.

[2] ACM. *Artifact Review and Badging.* 2016. URL: `https://www.acm.org/publications/policies/artifact-review-badging`.

[3] David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J. Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella-Béguelin, and Paul Zimmermann. „Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice“. In: *Proceedings of the 2015 Conference on Computer and Communications Security.* ACM. Denver, CO, USA, Oct. 2015. DOI: `10.1145/2810103.2813707`.

[4] David Adrian, Zakir Durumeric, Gulshan Singh, and J Alex Halderman. „Zippier ZMap: Internet-Wide Scanning at 10 Gbps“. In: *Proceedings of the 2014 USENIX Workshop on Offensive Technologies.* USENIX Association. San Diego, CA, USA, Aug. 2014.

[5] Maarten Aertsen, Maciej Korczyński, Giovane Moura, Samaneh Tajalizadehkhoob, and Jan van den Berg. „No domain left behind: is Let's Encrypt democratizing encryption?“ In: *Proceedings of the 2017 Applied Networking Research Workshop.* ACM. 2017, pp. 48–54.

[6] Akamai. *DDoS Resource Center.* URL: `https://www.akamai.com/us/en/security/ddos-resource-center.jsp`.

[7] Akamai Technologies Inc. *Entropy/IP open-source implementation.* URL: `https://github.com/akamai/entropy-ip`.

[8] Devdatta Akhawe, Bernhard Amann, Matthias Vallentin, and Robin Sommer. „Here's My Cert, So Trust Me, Maybe?: Understanding TLS Errors on the Web“.

In: *Proceedings of the 2013 Conference on World Wide Web*. ACM. May 2013, pp. 59–70.

[9] Alexa. *Alexa Top Sites*. URL: https://www.alexa.com/topsites.

[10] Alexa. *Alexa Top Sites By Country*. URL: https://www.alexa.com/topsites/countries.

[11] Andrew Allemann. *Network Solutions Displays Customers' Whois Queries to the Public*. Domain Name Wire. Sept. 1, 2009. URL: https://domainnamewire.com/2009/09/01/network-solutions-displays-customers-whois-queries-to-the-public/.

[12] Andrew Allemann. *Network Solutions Faces PR Nightmare Over Domain FrontRunning*. Domain Name Wire. Jan. 8, 2008. URL: https://domainnamewire.com/2008/01/08/network-solutions-faces-pr-nightmare-over-domain-frontrunning/.

[13] Rafael Almeida, Osvaldo Fonseca, Elverton Fazzion, Dorgival Guedes, Wagner Meira, and Ítalo Cunha. „A Characterization of Load Balancing on the IPv6 Internet“. In: *Proceedings of the 2017 Passive and Active Measurement Conference*. Springer. Sydney, Australia, Mar. 2017, pp. 242–254.

[14] Lance Alt, Robert Beverly, and Alberto Dainotti. „Uncovering Network Tarpits with Degreaser“. In: *Annual Computer Security Applications Conference*. 2014.

[15] Johanna Amann, Matthias Vallentin, Seth Hall, and Robin Sommer. *Extracting Certificates from Live Traffic: A Near Real-Time SSL Notary Service*. Tech. rep. TR-12-014. International Computer Science Institute, Nov. 2012.

[16] Johanna Amann, Oliver Gasser, Quirin Scheitle, Lexi Brent, Georg Carle, and Ralph Holz. „Mission Accomplished? HTTPS Security after DigiNotar“. In: *Proceedings of the 2017 Internet Measurement Conference*. Community Contribution Award, IRTF Applied Networking Research Prize. ACM. London, United Kingdom, Nov. 2017, pp. 325–340.

[17] Johanna Amann, Robin Sommer, Matthias Vallentin, and Seth Hall. „No Attack Necessary: The Surprising Dynamics of SSL Trust Relationships“. In: *Proceedings of the 2013 Computer Security Applications Conference*. ACM. 2013, pp. 179–188.

[18] APWG. *Anti-Phishing Working Group*. URL: http://antiphishing.org.

[19] ASHRAE. *BACnet - A Data Communication Protocol for Building Automation and Control Systems*. Ed. by Refrigerating American National Institute/American Society of Heating and Air-Conditioning Engineering. 1995.

[20] ASHRAE. *BACnet - A Data Communication Protocol for Building Automation and Control Systems Addendum 135-2012aj*. Ed. by Refrigerating American Na-

tional Institute/American Society of Heating and Air-Conditioning Engineering. 2016.

[21] Vaibhav Bajpai, Saba Ahsan, Jürgen Schönwälder, and Jörg Ott. „Measuring YouTube Content Delivery over IPv6“. In: *ACM SIGCOMM Computer Communication Review* 47.5 (Oct. 2017), pp. 2–11.

[22] D. Balenson. *Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers.* RFC 1423 (Historic). RFC. Fremont, CA, USA: RFC Editor, Feb. 1993. DOI: `10.17487/RFC1423`. URL: `https://www.rfc-editor.org/rfc/rfc1423.txt`.

[23] Shehar Bano, Philipp Richter, Mobin Javed, Srikanth Sundaresan, Zakir Durumeric, Steven J Murdoch, Richard Mortier, and Vern Paxson. „Scanning the Internet for Liveness“. In: *ACM SIGCOMM Computer Communication Review* 48.2 (Apr. 2018), pp. 2–9. DOI: `10.1145/3213232.3213234`.

[24] Elaine Barker, William Barker, William Burr, William Polk, and Miles Smid. „Recommendation for Key Management – Part 1: General (Revision 3)“. In: *NIST Special Publication* 800.57 (2012), pp. 1–147. URL: `https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-3/archive/2012-07-10`.

[25] Felix Beil. „Long Term Analysis of HTTP Strict Transport Security“. Advised by Quirin Scheitle, Oliver Gasser. Bachelor’s thesis. Technical University of Munich, July 2018.

[26] Benjamin Beurdouche, Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cédric Fournet, Markulf Kohlweiss, Alfredo Pironti, Pierre-Yves Strub, and Jean Karim Zinzindohoue. „A Messy State of the Union: Taming the Composite State Machines of TLS“. In: *Proceedings of the 2015 Symposium on Security and Privacy.* IEEE. San Jose, CA, USA, May 2015.

[27] Robert Beverly, Ramakrishnan Durairajan, David Plonka, and Justin P Rohrer. „In the IP of the Beholder: Strategies for Active IPv6 Topology Discovery“. In: *Proceedings of the 2018 Internet Measurement Conference.* ACM. Boston, MA, USA, Nov. 2018.

[28] Pirmin Blanz. „IPv6 TLS Security Scanning“. Advised by Oliver Gasser, Quirin Scheitle. Master’s thesis. Technical University of Munich, Feb. 2017.

[29] Birk Blechschmidt and Quirin Scheitle. *massdns, commit 4b3148a.* 2017. URL: `https://github.com/quirins/massdns`.

[30] Anthony Bonkoski, Russ Bielawski, and J. Alex Halderman. „Illuminating the Security Issues Surrounding Lights-Out Server Management“. In: *Proceedings of the 2014 USENIX Workshop on Offensive Technologies.* USENIX Association. San Diego, CA, USA, Aug. 2014.

[31] Kevin Borgolte, Shuang Hao, Tobias Fiebig, and Giovanni Vigna. „Enumerating Active IPv6 Hosts for Large-scale Security Scans via DNSSEC-signed Reverse Zones“. In: *Proceedings of the 2018 Symposium on Security and Privacy*. IEEE. San Francisco, CA, USA, May 2018.

[32] Jon Brodkin. *Domain registrar auto-enrolls customers into $1,850 security service (updated)*. Ars Technica. Jan. 22, 2014. URL: `https://arstechnica.com/information-technology/2014/01/domain-registrar-auto-enrolls-customers-into-1850-security-service/`.

[33] Michael Brunner. „Multi-protocol Path Topology Evaluation“. Advised by Minoo Rouhi, Dominik Scholz, Oliver Gasser. Bachelor's thesis. Technical University of Munich, Sept. 2017.

[34] Tobias Brunnwieser. „A Framework for Detection and Analysis of HTTPS Interception “. Advised by Oliver Gasser, Sree Harsha Totakura, Florian Wohlfart. Master's thesis. Technical University of Munich, Apr. 2018.

[35] Bundesamt für Sicherheit in der Informationstechnik. *Schutz Kritischer Infrastrukturen durch IT-Sicherheitsgesetz und UP KRITIS*. URL: `https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Broschueren/Schutz-Kritischer-Infrastrukturen-ITSig-u-UP-KRITIS.pdf`.

[36] Timm Böttger. „Detection of Amplification Attacks in Amplifier Networks“. Advised by Oliver Gasser, Lothar Braun, Felix von Eye. Master's thesis. Technical University of Munich, May 2014.

[37] Timm Böttger, Lothar Braun, Oliver Gasser, Felix von Eye, Helmut Reiser, and Georg Carle. „DoS Amplification Attacks – Protocol-Agnostic Detection of Service Abuse in Amplifier Networks“. In: *Proceedings of the 2015 Workshop on Traffic Monitoring and Analysis*. Springer. Barcelona, Spain, Apr. 2015.

[38] CA/Browser Forum. *Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates*. Version 1.5.0. Sept. 1, 2017. URL: `https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.5.0.pdf`.

[39] CAIDA. *Routeviews Prefix to AS mapping*. URL: `https://www.caida.org/data/routing/routeviews-prefix2as.xml`.

[40] CAIDA. *The IPv6 DNS Names Dataset*. URL: `https://www.caida.org/data/active/ipv6_dnsnames_dataset.xml`.

[41] Patricia Callejo, Connor Kelton, Narseo Vallina-Rodriguez, Rubén Cuevas, Oliver Gasser, Christian Kreibich, Florian Wohlfart, and Ángel Cuevas. „Opportunities and Challenges of Ad-based Measurements from the Edge of the Network“. In: *Proceedings of the 2017 Workshop on Hot Topics in Networks*. ACM. Nov. 2017.

[42] Frank Cangialosi, Taejoong Chung, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, and Christo Wilson. „Measurement and Analysis of Private Key

Sharing in the HTTPS Ecosystem“. In: *Proceedings of the 2016 Conference on Computer and Communications Security*. ACM. Vienna, Austria, Oct. 2016, pp. 628–640.

[43] Censys. *ICMP Echo Request Full IPv4 Scan Results*. URL: `https://censys.io/data/0-icmp-echo_request-full_ipv4`.

[44] Center for Measurement and Analysis of Network Data @ NPS. *Yarrp (Yelling at Random Routers Progressively)*. URL: `https://www.cmand.org/yarrp`.

[45] Nikolaos Chatzis, Georgios Smaragdakis, Jan Böttger, Thomas Krenc, and Anja Feldmann. „On the Benefits of Using a Large IXP as an Internet Vantage Point“. In: *Proceedings of the 2013 Internet Measurement Conference*. ACM. Barcelona, Spain, Oct. 2013, pp. 333–346.

[46] Yizheng Chen, Manos Antonakakis, Roberto Perdisci, Yacin Nadji, David Dagon, and Wenke Lee. „DNS noise: Measuring the Pervasiveness of Disposable Domains in Modern DNS Traffic“. In: *Proceedings of the 2014 Conference on Dependable Systems and Networks*. IEEE. 2014, pp. 598–609.

[47] Dmitry Chokovski. „Is IPv6 faster than IPv4“. Advised by Paul Emmerich, Quirin Scheitle, Oliver Gasser. Bachelor’s thesis. Technical University of Munich, Mar. 2016.

[48] Chromium. *HSTS Preload List Submission*. URL: `https://hstspreload.org/`.

[49] Chromium. *HSTS/HPKP Preload Lists, v389, commit 21f26e9*. URL: `https://cs.chromium.org/chromium/src/net/http/transport_security_state_static.json?rcl=21f26e9`.

[50] Chromium authors. *CT over DNS implementation in Chromium*. URL: `https://cs.chromium.org/chromium/src/components/certificate_transparency/log_dns_client.cc?type=cs&sq=package:chromium`.

[51] Taejoong Chung, Yabing Liu, David Choffnes, Dave Levin, Bruce MacDowell Maggs, Alan Mislove, and Christo Wilson. „Measuring and Applying Invalid SSL Certificates: The Silent Majority“. In: *Proceedings of the 2016 Internet Measurement Conference*. ACM. Santa Monica, CA, USA, Nov. 2016, pp. 527–541.

[52] Cisco. *Umbrella Popularity List*. URL: `http://s3-us-west-1.amazonaws.com/umbrella-static/index.html`.

[53] Jeremy Clark and Paul C van Oorschot. „SoK: SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements“. In: *Proceedings of the 2013 Symposium on Security and Privacy*. IEEE. San Francisco, CA, USA, May 2013, pp. 511–525.

[54] Cloudflare. *Advanced DDoS Attack Protection*. URL: `https://www.cloudflare.com/ddos/`.

[55] Common Vulnerabilities and Exposures. *CVE-2003-0931.* Nov. 2003. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-0931.

[56] 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.* RFC 5280 (Proposed Standard). RFC. Updated by RFCs 6818, 8398, 8399. Fremont, CA, USA: RFC Editor, May 2008. DOI: 10.17487/RFC5280. URL: https://www.rfc-editor.org/rfc/rfc5280.txt.

[57] Andrei Costin, Jonas Zaddach, Aurélien Francillon, and Davide Balzarotti. „A Large-Scale Analysis of the Security of Embedded Firmwares“. In: *Proceedings of the 2014 USENIX Security Symposium.* USENIX Association. San Diego, CA, USA, Aug. 2014, pp. 95–110.

[58] Creative Commons. *Attribution-ShareAlike 4.0 International (CC BY-SA 4.0).* URL: https://creativecommons.org/licenses/by-sa/4.0/.

[59] Jakub Czyz, Matthew J Luckie, Mark Allman, and Michael Bailey. „Don't Forget to Lock the Back Door! A Characterization of IPv6 Network Security Policy“. In: *Proceedings of the 2016 Symposium on Network and Distributed System Security.* Internet Society. San Diego, CA, USA, Feb. 2016.

[60] Jakub Czyz, Mark Allman, Jing Zhang, Scott Iekel-Johnson, Eric Osterweil, and Michael Bailey. „Measuring IPv6 adoption“. In: vol. 44. 4. New York, NY, USA: ACM, Oct. 2014, pp. 87–98.

[61] E. Davies and J. Mohacsi. *Recommendations for Filtering ICMPv6 Messages in Firewalls.* RFC 4890 (Informational). RFC. Fremont, CA, USA: RFC Editor, May 2007. DOI: 10.17487/RFC4890. URL: https://www.rfc-editor.org/rfc/rfc4890.txt.

[62] S. Deering and R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification.* RFC 1883 (Proposed Standard). RFC. Obsoleted by RFC 2460. Fremont, CA, USA: RFC Editor, Dec. 1995. DOI: 10.17487/RFC1883. URL: https://www.rfc-editor.org/rfc/rfc1883.txt.

[63] S. Deering and R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification.* RFC 8200 (Internet Standard). RFC. Fremont, CA, USA: RFC Editor, July 2017. DOI: 10.17487/RFC8200. URL: https://www.rfc-editor.org/rfc/rfc8200.txt.

[64] Christoph Dietzel, Anja Feldmann, and Thomas King. „Blackholing at IXPs: On the Effectiveness of DDoS Mitigation in the Wild“. In: *Proceedings of the 2016 Passive and Active Measurement Conference.* Springer. Heraklion, Greece, Apr. 2016, pp. 319–332.

[65] Christoph Dietzel, Georgios Smaragdakis, Matthias Wichtlhuber, and Anja Feldmann. „Stellar: Network Attack Mitigation using Advanced Blackholing“. In:

Proceedings of the 2018 International Conference on emerging Networking EXperiments and Technologies. ACM. 2018, pp. 152–164.

[66] Peter van Dijk. *Finding v6 hosts by efficiently mapping ip6.arpa*. Mar. 2012. URL: `https://web.archive.org/web/20170603234058/http://7bits.nl/blog/posts/finding-v6-hosts-by-efficiently-mapping-ip6-arpa`.

[67] David Dittrich, Erin Kenneally, et al. „The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research". In: *US Department of Homeland Security* (2012).

[68] Trinh Viet Doan, Ljubica Pajevic, Vaibhav Bajpai, and Jorg Ott. „Tracing the Path to YouTube: A Quantification of Path Lengths and Latencies Toward Content Caches". In: *IEEE Communications Magazine* 57.1 (2019), pp. 80–86.

[69] Michael Domke. „DNS-in-a-Box: A Testing Framework for Reproducible Network Measurements". Advised by Johannes Naab, Oliver Gasser. Bachelor's thesis. Technical University of Munich, Nov. 2015.

[70] Zakir Durumeric. „Fast Internet-Wide Scanning: A New Security Perspective". PhD thesis. University of Michigan, 2017.

[71] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. „ZMap: Fast Internet-wide Scanning and Its Security Applications". In: *Proceedings of the 2013 USENIX Security Symposium*. USENIX Association. Washington, D.C., USA, Aug. 2013, pp. 47–53.

[72] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. „A Search Engine Backed by Internet-wide Scanning". In: *Proceedings of the 2015 Conference on Computer and Communications Security*. ACM. Denver, CO, USA, Oct. 2015.

[73] Zakir Durumeric, James Kasten, Michael Bailey, and J Alex Halderman. „Analysis of the HTTPS certificate ecosystem". In: *Proceedings of the 2013 Internet Measurement Conference*. ACM. Barcelona, Spain, Oct. 2013, pp. 291–304.

[74] Zakir Durumeric, David Adrian, Ariana Mirian, James Kasten, Elie Bursztein, Nicolas Lidzborski, Kurt Thomas, Vijay Eranti, Michael Bailey, and J Alex Halderman. „Neither Snow Nor Rain Nor MITM...: An Empirical Analysis of Email Delivery Security". In: *Proceedings of the 2015 Internet Measurement Conference*. ACM. Tokyo, Japan, Oct. 2015.

[75] Zakir Durumeric, Frank Li, James Kasten, Johanna Amann, Jethro Beekman, Mathias Payer, Nicolas Weaver, David Adrian, Vern Paxson, Michael Bailey, et al. „The Matter of Heartbleed". In: *Proceedings of the 2014 Internet Measurement Conference*. ACM. Vancouver, BC, Canada, Oct. 2014, pp. 475–488.

[76] D. Eastlake 3rd. *Transport Layer Security (TLS) Extensions: Extension Definitions*. RFC 6066 (Proposed Standard). RFC. Updated by RFCs 8446, 8449.

Fremont, CA, USA: RFC Editor, Jan. 2011. DOI: 10.17487/RFC6066. URL: https://www.rfc-editor.org/rfc/rfc6066.txt.

[77] W. Eddy. *TCP SYN Flooding Attacks and Common Mitigations*. RFC 4987 (Informational). RFC. Fremont, CA, USA: RFC Editor, Aug. 2007. DOI: 10.17487/RFC4987. URL: https://www.rfc-editor.org/rfc/rfc4987.txt.

[78] Hendrik Eichner. „Revisiting SSH Security in the Internet“. Advised by Oliver Gasser, Minoo Rouhi. Bachelor's thesis. Technical University of Munich, Sept. 2017.

[79] Samy El Deib. „Detecting IPv6-IPv4 Sibling Pairs Based on few Data Points“. Advised by Quirin Scheitle, Oliver Gasser, Minoo Rouhi. Bachelor's thesis. Technical University of Munich, Aug. 2017.

[80] C. Ellison and B. Schneier. „Ten Risks of PKI: what You're not Being Told about Public Key Infrastructure“. In: *Computer Security Journal* 16.1 (2000), pp. 1–7.

[81] Felix Emmert. „Messung und Evaluation der Verbreitung von IPMI-Geräten mit aktiven Scans“. Advised by Oliver Gasser. Bachelor's thesis. Technical University of Munich, Nov. 2015.

[82] Roger Entner. *International Comparisons: The Handset Replacement Cycle*. Tech. rep. Recon Analytics, Feb. 2013. URL: http://mobilefuture.org/wp-content/uploads/2013/02/mobile-future.publications.handset-replacement-cycle.pdf.

[83] C. Evans, C. Palmer, and R. Sleevi. *Public Key Pinning Extension for HTTP*. RFC 7469 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Apr. 2015. DOI: 10.17487/RFC7469. URL: https://www.rfc-editor.org/rfc/rfc7469.txt.

[84] Felix von Eye, Timm Böttger, Helmut Reiser, Lothar Braun, Oliver Gasser, and Georg Carle. „Detektion und Prävention von Denial-of-Service Amplification Attacken – Schutz des Netzes aus Sicht eines Amplifiers“. In: *Konferenzband zur 22. DFN-Konferenz Sicherheit in vernetzten Systemen*. Ed. by Christian Paulsen. Norderstedt, Deutschland, Feb. 2015, H–1–H–13. ISBN: 978-3-7347-5309-1.

[85] Dan Farmer. „IPMI: Freight Train to Hell“. In: (2013). DOI: 10.1016/S0022-3913(12)00047-9. URL: http://fish2.com/ipmi/itrain.pdf.

[86] Dan Farmer. „Sold Down the River“. In: (2013). URL: http://fish2.com/ipmi/river.pdf.

[87] Farsight Security. *DNSDB*. URL: https://www.dnsdb.info/.

[88] Federal Trade Commission. *FTC Obtains Settlement From Network Solutions LLC for Misleading Consumers About Refunds*. Apr. 7, 2015. URL: https://www.ftc.gov/news-events/press-releases/2015/04/ftc-obtains-settlement-network-solutions-llc-misleading-consumers.

[89] Adrienne Porter Felt, Richard Barnes, April King, Chris Palmer, Chris Bentzel, and Parisa Tabriz. „Measuring HTTPS Adoption on the Web“. In: *Proceedings of the 2017 USENIX Security Symposium*. USENIX Association. Vancouver, BC, Canada, Aug. 2017.

[90] Xuan Feng, Qiang Li, Haining Wang, and Limin Sun. „Characterizing Industrial Control System Devices on the Internet“. In: *Proceedings of the 2016 International Conference on Network Protocols*. IEEE. 2016, pp. 1–10.

[91] Tobias Fiebig. „An Empirical Evaluation of Misconfiguration in Internet Services“. PhD thesis. Technische Universität Berlin, 2017.

[92] Tobias Fiebig, Kevin Borgolte, Shuang Hao, Christopher Kruegel, Giovanni Vigna, and Anja Feldmann. „In rDNS We Trust: Revisiting a Common Data-Source's Reliability“. In: *Proceedings of the 2018 Passive and Active Measurement Conference*. Springer. Berlin, Germany, Mar. 2018.

[93] Tobias Fiebig, Kevin Borgolte, Shuang Hao, Christopher Kruegel, and Giovanni Vigna. „Something from Nothing (There): Collecting Global IPv6 Datasets from DNS“. In: *Proceedings of the 2017 Passive and Active Measurement Conference*. Springer. Sydney, Australia, Mar. 2017, pp. 30–43.

[94] Johannes Fischer. „Browser-based Internet Connection Testing“. Advised by Florian Wohlfart, Oliver Gasser. Master's thesis. Technical University of Munich, July 2017.

[95] Pawel Foremski. *New Entropy/IP generator*. URL: `https://github.com/pforemski/eip-generator`.

[96] Pawel Foremski, David Plonka, and Arthur Berger. „Entropy/IP: Uncovering Structure in IPv6 Addresses“. In: *Proceedings of the 2016 Internet Measurement Conference*. ACM. Santa Monica, CA, USA, Nov. 2016, pp. 167–181.

[97] Oliver Gasser. *bacnet.py: BACnet python module to parse BACnet response packets*. Nov. 2016. URL: `https://github.com/tumi8/bacnet.py`.

[98] Oliver Gasser, Felix Emmert, and Georg Carle. „Digging for Dark IPMI Devices: Advancing BMC Detection and Evaluating Operational Security“. In: *Proceedings of the 2016 Workshop on Traffic Monitoring and Analysis*. IFIP. Louvain-la-Neuve, Belgium, Apr. 2016.

[99] Oliver Gasser, Ralph Holz, and Georg Carle. „A deeper understanding of SSH: Results from Internet-wide scans“. In: *Proceedings of the 2014 Network Operations and Management Symposium*. IEEE. Krakow, Poland, May 2014.

[100] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczynski, Stephen D. Strowes, Luuk Hendriks, and Georg Carle. „Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists“. In: *Proceedings of the 2018 Internet*

Measurement Conference. ACM. Boston, MA, USA, Nov. 2018. DOI: 10.1145/3278532.3278564.

[101] Oliver Gasser, Benjamin Hof, Max Helm, Maciej Korczynski, Ralph Holz, and Georg Carle. „In Log We Trust: Revealing Poor Security Practices with Certificate Transparency Logs and Internet Measurements“. In: *Proceedings of the 2018 Passive and Active Measurement Conference*. Best Paper Award. Springer. Berlin, Germany, Mar. 2018, pp. 173–185.

[102] Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle. *IPv6 Hitlist Collection*. 2016. URL: https://www.net.in.tum.de/projects/gino/ipv6-hitlist.html.

[103] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczynski, Stephen D. Strowes, Luuk Hendriks, and Georg Carle. *IPv6 Hitlist Service*. 2018. URL: https://ipv6hitlist.github.io/.

[104] Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle. „Scanning the IPv6 Internet: Towards a Comprehensive Hitlist“. In: *Proceedings of the 2016 Workshop on Traffic Monitoring and Analysis*. IFIP. Louvain-la-Neuve, Belgium, Apr. 2016.

[105] Oliver Gasser, Quirin Scheitle, Carl Denis, Nadja Schricker, and Georg Carle. „Security Implications of Publicly Reachable Building Automation Systems“. In: *Proceedings of the 2nd International Workshop on Traffic Measurements for Cybersecurity*. San Jose, CA, USA, May 2017.

[106] Oliver Gasser, Quirin Scheitle, Benedikt Rudolph, Carl Denis, Nadja Schricker, and Georg Carle. „The Amplification Threat Posed by Publicly Reachable BACnet Devices“. In: *Journal of Cyber Security and Mobility* 6.1 (Oct. 2017).

[107] Oliver Gasser, Quirin Scheitle, Carl Denis, Nadja Schricker, and Georg Carle. „Öffentlich erreichbare Gebäudeautomatisierung: Amplification-Anfälligkeit von BACnet und Deployment-Analyse im Internet und DFN“. In: *Konferenzband zur 24. DFN-Konferenz Sicherheit in vernetzten Systemen*. Hamburg, Germany, Feb. 2017.

[108] Sebastian Gebhard. „IPv6 Scanning - Smart address selection and comparison to legacy IP“. Advised by Oliver Gasser, Quirin Scheitle. Master’s thesis. Technical University of Munich, Mar. 2016.

[109] Andrew Gerrand. *Share Memory By Communicating*. July 2010. URL: https://blog.golang.org/share-memory-by-communicating.

[110] Go Project. *The Go Programming Language*. URL: https://golang.org/.

[111] F. Gont and T. Chown. *Network Reconnaissance in IPv6 Networks*. RFC 7707 (Informational). RFC. Fremont, CA, USA: RFC Editor, Mar. 2016. DOI: 10.17487/RFC7707. URL: https://www.rfc-editor.org/rfc/rfc7707.txt.

[112] Google. *Google IPv6 adoption.* URL: https://www.google.com/intl/en/ipv6/statistics.html.

[113] Google. *Project Shield.* URL: https://projectshield.withgoogle.com/.

[114] Robert David Graham. *Masscan: Mass IP port scanner.* 2014. URL: https://github.com/robertdavidgraham/masscan.

[115] Christopher Grayson. *IPv666 – Address of the Beast.* Nov. 2018. URL: https://l.avala.mp/?p=285.

[116] Josef Gustafsson, Gustaf Overier, Martin Arlitt, and Niklas Carlsson. „A First Look at the CT Landscape: Certificate Transparency Logs in Practice“. In: *Proceedings of the 2017 Passive and Active Measurement Conference.* Springer. Sydney, Australia, Mar. 2017. DOI: 10.1007/978-3-319-54328-4_7.

[117] P. Hallam-Baker and R. Stradling. *DNS Certification Authority Authorization (CAA) Resource Record.* RFC 6844 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Jan. 2013. DOI: 10.17487/RFC6844. URL: https://www.rfc-editor.org/rfc/rfc6844.txt.

[118] HD Moore. „A Penetration Tester's Guide to IPMI and BMCs“. In: (Jan. 2013). URL: https://community.rapid7.com/community/metasploit/blog/2013/07/02/a-penetration-testers-guide-to-ipmi/.

[119] Jonas Heintzenberg. „Browser-based measurement of NAT middleboxes“. Advised by Florian Wohlfart, Oliver Gasser. Bachelor's thesis. Technical University of Munich, May 2017.

[120] Max Helm. „Evaluating TLS Certificate Transparency Logs using Active Scans“. Advised by Oliver Gasser, Benjamin Hof. Interdisciplinary project. Technical University of Munich, Sept. 2017.

[121] Max Helm. „Traceable Measurement Result Publication in Append-only Ledgers“. Advised by Oliver Gasser, Benjamin Hof, Quirin Scheitle. Master's thesis. Technical University of Munich, June 2018.

[122] Luuk Hendriks. „Measuring IPv6 Resilience and Security“. PhD thesis. University of Twente, 2019.

[123] Nadia Heninger, Zakir Durumeric, Eric Wustrow, and J Alex Halderman. „Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices.“ In: *Proceedings of the 2012 USENIX Security Symposium.* USENIX Association. Bellevue, WA, USA, Aug. 2012.

[124] Highsoft. *Highcharts.* URL: https://www.highcharts.com.

[125] Arno Hilke. „IP Spoofing Detection Through Time to Live Header Analysis“. Advised by Quirin Scheitle, Oliver Gasser. Bachelor's thesis. Technical University of Munich, Apr. 2016.

[126] R. Hinden and S. Deering. *IP Version 6 Addressing Architecture*. RFC 4291 (Draft Standard). RFC. Updated by RFCs 5952, 6052, 7136, 7346, 7371, 8064. Fremont, CA, USA: RFC Editor, Feb. 2006. DOI: 10.17487/RFC4291. URL: https://www.rfc-editor.org/rfc/rfc4291.txt.

[127] J. Hodges, C. Jackson, and A. Barth. *HTTP Strict Transport Security (HSTS)*. RFC 6797 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Nov. 2012. DOI: 10.17487/RFC6797. URL: https://www.rfc-editor.org/rfc/rfc6797.txt.

[128] P. Hoffman and J. Schlyter. *The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA*. RFC 6698 (Proposed Standard). RFC. Updated by RFCs 7218, 7671. Fremont, CA, USA: RFC Editor, Aug. 2012. DOI: 10.17487/RFC6698. URL: https://www.rfc-editor.org/rfc/rfc6698.txt.

[129] Ralph Holz, Lothar Braun, Nils Kammenhuber, and Georg Carle. „The SSL Landscape – A Thorough Analysis of the X.509 PKI Using Active and Passive Measurements“. In: *Proceedings of the 2011 Internet Measurement Conference*. ACM. Berlin, Germany, Nov. 2011, pp. 427–444.

[130] Ralph Holz, Johanna Amann, Olivier Mehani, Matthias Wachs, and Mohamed Ali Kaafar. „TLS in the Wild: An Internet-wide Analysis of TLS-based Protocols for Electronic Communication“. In: *Proceedings of the 2016 Symposium on Network and Distributed System Security*. Internet Society. San Diego, CA, USA, Feb. 2016.

[131] Ralph-Günther Holz. „Empirical Analysis of Public Key Infrastructures and Investigation of Improvements“. PhD thesis. Technical University of Munich, 2014.

[132] Lin-Shung Huang, Shrikant Adhikarla, Dan Boneh, and Collin Jackson. „An Experimental Study of TLS Forward Secrecy Deployments“. In: *IEEE Internet Computing* 18.6 (2014), pp. 43–51.

[133] IANA. *IPv4 Special-Purpose Address Registry*. URL: http://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml.

[134] IANA. *TLS cipher suite parameters CSV file*.

[135] ICANN. *Centralized Zone Data Service*. URL: https://czds.icann.org/.

[136] IEEE. *MAC Vendor List*. URL: http://standards-oui.ieee.org/oui.txt.

[137] Imperva. *DDoS Protection*. URL: https://www.incapsula.com/ddos-protection-services.html.

[138] *Intelligent Platform Management Interface Specification Second Generation*. E6 Markup. Revision 1.1. Intel, Hewlett-Packard, NEC, and Dell. Feb. 2014. URL:

http://www.intel.com/content/www/us/en/servers/ipmi/ipmi-v2-rev1-1-spec-errata-6-markup.html.

[139] Internet Society. *World IPv6 Launch.* URL: https://www.worldipv6launch.org/.

[140] *IP2Location IP Address Geolocation Database.* IP2Location.com. URL: https://ip2location.com/database/ip2location.

[141] Matthias Jaros. „Distribution and Orchestration of Network Measurements on the Planetlab testbed“. Advised by Ralph Holz, Oliver Gasser. Bachelor's thesis. Technical University of Munich, Apr. 2015.

[142] Juniper. *Out of Cycle Security Bulletin: ScreenOS: Multiple Security issues with ScreenOS (CVE-2015-7755, CVE-2015-7756).* Juniper. URL: http://kb.juniper.net/InfoCenter/index?page=content&id=JSA10713.

[143] B. Kaliski. *Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services.* RFC 1424 (Historic). RFC. Fremont, CA, USA: RFC Editor, Feb. 1993. DOI: 10.17487/RFC1424. URL: https://www.rfc-editor.org/rfc/rfc1424.txt.

[144] S. Kent. *Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management.* RFC 1422 (Historic). RFC. Fremont, CA, USA: RFC Editor, Feb. 1993. DOI: 10.17487/RFC1422. URL: https://www.rfc-editor.org/rfc/rfc1422.txt.

[145] Albert Khakimullin. „Real-time Amplification Attack Detection“. Advised by Oliver Gasser, Felix von Eye. Interdisciplinary project. Technical University of Munich, July 2016.

[146] K. Kinnear, M. Stapp, R. Desetti, B. Joshi, N. Russell, P. Kurapati, and B. Volz. *DHCPv4 Bulk Leasequery.* RFC 6926 (Proposed Standard). RFC. Updated by RFC 7724. Fremont, CA, USA: RFC Editor, Apr. 2013. DOI: 10.17487/RFC6926. URL: https://www.rfc-editor.org/rfc/rfc6926.txt.

[147] Lukas Krämer, Johannes Krupp, Daisuke Makita, Tomomi Nishizoe, Takashi Koide, Katsunari Yoshioka, and Christian Rossow. „AmpPot: Monitoring and Defending Against Amplification DDoS Attacks“. In: *Proceedings of the 2015 Workshop on Recent Advances in Intrusion Detection.* Springer. 2015, pp. 615–636.

[148] Michael Kranch and Joseph Bonneau. „Upgrading HTTPS in Mid-Air: An Empirical Study of Strict Transport Security and Key Pinning“. In: *Proceedings of the 2015 Symposium on Network and Distributed System Security.* Internet Society. San Diego, CA, USA, Feb. 2015.

[149] Brian Krebs. *Hacked Cameras, DVRs Powered Today's Massive Internet Outage.* Oct. 2016. URL: https://krebsonsecurity.com/2016/10/hacked-cameras-dvrs-powered-todays-massive-internet-outage/.

[150] Brian Krebs. *KrebsOnSecurity Hit With Record DDoS.* Sept. 2016. URL: http://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/.

[151] Deepak Kumar, Zhengping Wang, Matthew Hyder, Joseph Dickinson, Gabrielle Beck, David Adrian, Joshua Mason, Zakir Durumeric, J. Alex Halderman, and Michael Bailey. „Tracking Certificate Misissuance in the Wild“. In: *Proceedings of the 2018 Symposium on Security and Privacy.* IEEE. San Francisco, CA, USA, May 2018.

[152] Michael Köpferl. „Effective Visualization of Amplification Attacks in Amplifier Networks“. Advised by Oliver Gasser, Felix von Eye. Bachelor's thesis. Technical University of Munich, Sept. 2015.

[153] Michael Köpferl. „Evaluation of amplification attacks in large-scale networks to improve detection performance“. Advised by Oliver Gasser, Stefan Metzger. Interdisciplinary project. Technical University of Munich, Mar. 2018.

[154] NLnet Labs. *Unbound 1.6.0 DNS resolver.* 2017. URL: https://www.unbound.net.

[155] Jan-Philipp Lauinger. „Evaluating Client Discrimination in Anonymization Networks Using Active Network Scans“. Advised by Oliver Gasser, Sree Harsha Totakura. Forschungspraxis. Technical University of Munich, Nov. 2017.

[156] B. Laurie, A. Langley, and E. Kasper. *Certificate Transparency.* RFC 6962 (Experimental). RFC. Fremont, CA, USA: RFC Editor, June 2013. DOI: 10.17487/RFC6962. URL: https://www.rfc-editor.org/rfc/rfc6962.txt.

[157] Ben Laurie, Pierre Haneuf, and Adam Eijdenberg. *Certificate Transparency RFCs.* 2017. URL: https://github.com/google/certificate-transparency-rfcs.

[158] Frank Li, Zakir Durumeric, Jakub Czyz, Mohammad Karami, Michael Bailey, Damon McCoy, Stefan Savage, and Vern Paxson. „You've Got Vulnerability: Exploring Effective Vulnerability Notifications“. In: *Proceedings of the 2016 USENIX Security Symposium.* USENIX Association. Austin, TX, USA, Aug. 2016.

[159] J. Linn. *Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures.* RFC 1421 (Historic). RFC. Fremont, CA, USA: RFC Editor, Feb. 1993. DOI: 10.17487/RFC1421. URL: https://www.rfc-editor.org/rfc/rfc1421.txt.

[160] Yabing Liu, Will Tome, Liang Zhang, David Choffnes, Dave Levin, Bruce Maggs, Alan Mislove, Aaron Schulman, and Christo Wilson. „An End-to-End Measure-

ment of Certificate Revocation in the Web's PKI". In: *Proceedings of the 2015 Internet Measurement Conference*. ACM. Tokyo, Japan, Oct. 2015, pp. 183–196.

[161] Matthew Luckie. „Scamper: a Scalable and Extensible Packet Prober for Active Measurement of the Internet". In: *Proceedings of the 2010 Internet Measurement Conference*. ACM. Melbourne, Australia, Nov. 2010.

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

[163] Marek Majkowski. *Abusing Linux's firewall: the hack that allowed us to build Spectrum*. URL: `https://blog.cloudflare.com/how-we-built-spectrum/`.

[164] David Malone. „Observations of IPv6 addresses". In: *Proceedings of the 2008 Passive and Active Network Measurement Conference*. Springer. Cleveland, OH, USA, Apr. 2008.

[165] Gervase Markham. *Mailing List: Mozilla dev.sec.policy: PROCERT decision*. URL: `https://groups.google.com/forum/#!topic/mozilla.dev.security.policy/Ymrpsm7s5_I`.

[166] Matthew Bryant. *TLD AXFR transfers*. URL: `https://github.com/mandatoryprogrammer/TLDR`.

[167] Eran Messeri. *Mailing List: IETF Trans: Privacy analysis of the DNS-based protocol for obtaining inclusion proof*. URL: `https://www.ietf.org/mail-archive/web/trans/current/msg02617.html`.

[168] Ariana Mirian, Zane Ma, David Adrian, Matthew Tischer, Thasphon Chuenchujit, Tim Yardley, Robin Berthier, Joshua Mason, Zakir Durumeric, J Alex Halderman, and Michael Bailey. „An Internet-Wide View of ICS Devices". In: *Proceedings of the 2016 Conference on Privacy, Security and Trust*. IEEE. 2016, pp. 96–103.

[169] B. Moeller and A. Langley. *TLS Fallback Signaling Cipher Suite Value (SCSV) for Preventing Protocol Downgrade Attacks*. RFC 7507 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Apr. 2015. DOI: `10.17487/RFC7507`. URL: `https://www.rfc-editor.org/rfc/rfc7507.txt`.

[170] Mozila. *OneCRL*. Oct. 2017. URL: `https://firefox.settings.services.mozilla.com/v1/buckets/blocklists/collections/certificates/records`.

[171] Mozilla. *HPKP Preload List*. URL: `https://wiki.mozilla.org/SecurityEngineering/Public_Key_Pinning/Implementation_Details`.

[172] Mozilla. *HSTS Preload List*. URL: `https://wiki.mozilla.org/SecurityEngineering/HTTP_Strict_Transport_Security_(HSTS)_Preload_List`.

[173] Mozilla. *Mailing List: Mozilla dev.sec.policy: Mozilla's Plan for Symantec Roots*. URL: `https://www.mail-archive.com/dev-security-policy@lists.mozilla.org/msg08290.html`.

[174] Mozilla. *Revoking Trust in Two TurkTrust Certificates.* URL: https://blog.mozilla.org/security/2013/01/03/revoking-trust-in-two-turktrust-certficates/.

[175] Mro. *File:Ipv4-exhaust.svg on Wikimedia Commons.* URL: https://commons.wikimedia.org/wiki/File:Ipv4-exhaust.svg.

[176] Austin Murdock, Frank Li, Paul Bramsen, Zakir Durumeric, and Vern Paxson. „Target Generation for Internet-wide IPv6 Scanning“. In: *Proceedings of the 2017 Internet Measurement Conference.* ACM. London, United Kingdom, Nov. 2017.

[177] Nils Mäurer. „Efficient scans in a research network“. Advised by Ralph Holz, Oliver Gasser. Bachelor's thesis. Technical University of Munich, Feb. 2015.

[178] Johannes Naab. „Scanning and Evaluating DNS Deployments in the Internet“. Master's thesis. Technical University of Munich, Mar. 2014.

[179] T. Narten, R. Draves, and S. Krishnan. *Privacy Extensions for Stateless Address Autoconfiguration in IPv6.* RFC 4941 (Draft Standard). RFC. Fremont, CA, USA: RFC Editor, Sept. 2007. DOI: 10.17487/RFC4941. URL: https://www.rfc-editor.org/rfc/rfc4941.txt.

[180] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. *Neighbor Discovery for IP version 6 (IPv6).* RFC 4861 (Draft Standard). RFC. Updated by RFCs 5942, 6980, 7048, 7527, 7559, 8028, 8319, 8425. Fremont, CA, USA: RFC Editor, Sept. 2007. DOI: 10.17487/RFC4861. URL: https://www.rfc-editor.org/rfc/rfc4861.txt.

[181] Michael Newman. *BACnet: The Global Standard for Building Automation and Control Networks.* Momentum Press, 2013, p. 44.

[182] NHK World Japan. *Govt. to access home devices in security survey.* Jan. 25, 2019. URL: https://www3.nhk.or.jp/nhkworld/en/news/20190125_44/.

[183] Johnathan Nightingale. *Mozilla Security Blog: DigiNotar Removal Follow Up.* Sept. 2011. URL: https://blog.mozilla.org/security/2011/09/02/diginotar-removal-follow-up/.

[184] Linus Nordberg, Daniel Gillmor, and Tom Ritter. *Gossiping in CT.* Internet-Draft draft-ietf-trans-gossip-04. URL: https://tools.ietf.org/html/draft-ietf-trans-gossip-04.

[185] Carl Nykvist, Linus Sjöström, Josef Gustafsson, and Niklas Carlsson. „Server-Side Adoption of Certificate Transparency“. In: *Proceedings of the 2018 Passive and Active Measurement Conference.* Springer. Berlin, Germany, Mar. 2018.

[186] Craig Partridge and Mark Allman. „Ethical Considerations in Network Measurement Papers“. In: *Communications of the ACM* 59.10 (2016), pp. 58–64.

[187] PhishTank. *A Nonprofit Anti-phishing Organization.* URL: http://www.phishtank.com.

[188] David Plonka and Arthur Berger. „kIP: a Measured Approach to IPv6 Address Anonymization“. In: *arXiv preprint arXiv:1707.03900* (2017).

[189] David Plonka and Arthur Berger. „Temporal and Spatial Classification of Active IPv6 Addresses“. In: *Proceedings of the 2015 Internet Measurement Conference*. ACM. Tokyo, Japan, Oct. 2015.

[190] Ingmar Poese, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, and Bamba Gueye. „IP Geolocation Databases: Unreliable?“ In: *ACM SIGCOMM Computer Communication Review* 41.2 (Apr. 2011), pp. 53–56.

[191] J. Postel. *Internet Protocol*. RFC 791 (Internet Standard). RFC. Updated by RFCs 1349, 2474, 6864. Fremont, CA, USA: RFC Editor, Sept. 1981. DOI: 10.17487/RFC0791. URL: https://www.rfc-editor.org/rfc/rfc791.txt.

[192] PremiumDrops. *Domain Lists*. 2017. URL: http://premiumdrops.com/zones.html.

[193] Matthew Prince. *Technical Details Behind a 400Gbps NTP Amplification DDoS Attack*. Feb. 2014. URL: https://blog.cloudflare.com/technical-details-behind-a-400gbps-ntp-amplification-ddos-attack/.

[194] Matthew Prince. *The DDoS That Almost Broke the Internet*. Mar. 2013. URL: https://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet/.

[195] Ronald Prins. *DigiNotar Certificate Authority Breach "Operation Black Tulip"*. Interim Report. Fox-IT, Sept. 2012. URL: https://www.rijksoverheid.nl/binaries/rijksoverheid/documenten/rapporten/2011/09/05/diginotar-public-report-version-1/rapport-fox-it-operation-black-tulip-v1-0.pdf.

[196] Fabian Raab. „Modeling and Analysis of BGP Community Attributes“. Advised by Oliver Gasser, Johann Schlamp. Bachelor's thesis. Technical University of Munich, Sept. 2015.

[197] Rapid7. *Project Sonar Datasets*. URL: https://opendata.rapid7.com.

[198] Alison DeNisco Rayome. *There will soon be more IoT devices in the world than people, security risks abound*. TechRepublic. Feb. 7, 2017. URL: https://www.techrepublic.com/article/there-will-soon-be-more-iot-devices-in-the-world-than-people-security-risks-abound/.

[199] Philipp Richter, Georgios Smaragdakis, David Plonka, and Arthur Berger. „Beyond Counting: New Perspectives on the Active IPv4 Address Space“. In: *Proceedings of the 2016 Internet Measurement Conference*. ACM. Santa Monica, CA, USA, Nov. 2016, pp. 135–149.

[200] Roland van Rijswijk-Deij, Mattijs Jonker, Anna Sperotto, and Aiko Pras. „A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measure-

ments“. In: *IEEE Journal on Selected Areas in Communications* 34.6 (2016), pp. 1877–1888.

[201] RIPE NCC. *IPMap.* URL: `https://ftp.ripe.net/ripe/ipmap/`.

[202] RIPE NCC. *RIPE Atlas.* URL: `https://atlas.ripe.net/`.

[203] Ivan Ristić. *SSL/TLS and PKI History.* URL: `https://www.feistyduck.com/ssl-tls-and-pki-history/`.

[204] Tom Ritter. *An Experimental "RequireCT" Directive for HSTS.* Feb. 2015. URL: `https://ritter.vg/blog-require_certificate_transparency.html`.

[205] Robots and Pencils. *goviz.* URL: `https://github.com/RobotsAndPencils/goviz`.

[206] Jim Roskind. *Quic UDP Internet Connections.* URL: `https://docs.google.com/document/d/1RNHkx_VvKWyWg6Lr8SZ-saqsQx7rFV-ev2jRFUoVD34`.

[207] Christian Rossow. „Amplification Hell: Revisiting Network Protocols for DDoS Abuse“. In: *Proceedings of the 2014 Symposium on Network and Distributed System Security.* Internet Society. San Diego, CA, USA, Feb. 2014.

[208] Minoo Rouhi Vejdani. „Path tracing and validation of IPv4 and IPv6 siblings“. Advised by Quirin Scheitle, Oliver Gasser, Paul Emmerich. Master’s thesis. Technical University of Munich, Aug. 2016.

[209] Katharina Rudert. „Botnet Detection using Machine Learning on Flow Data“. Advised by Johannes Naab, Oliver Gasser. Master’s thesis. Technical University of Munich, July 2016.

[210] Sergio de los Santos, Carmen Torrano, Yaiza Rubio, and Félix Brezo. „Implementation State of HSTS and HPKP in Both Browsers and Servers“. In: *Proceedings of the 2016 International Conference on Cryptology and Network Security.* Springer. Milan, Italy, Nov. 2016, pp. 192–207.

[211] Matthew Sargent, John Kristoff, Vern Paxson, and Mark Allman. „On the Potential Abuse of IGMP“. In: *ACM SIGCOMM Computer Communication Review* 47.1 (Jan. 2017), pp. 27–35.

[212] Patrick Sattler. „Parsing geographical locations from DNS names“. Advised by Quirin Scheitle, Oliver Gasser, Johannes Naab. Bachelor’s thesis. Technical University of Munich, Feb. 2016.

[213] Patrick Sattler. „Parsing geographical locations from DNS names“. Advised by Quirin Scheitle, Oliver Gasser. Guided research. Technical University of Munich, Nov. 2016.

[214] Patrick Sattler. „Parsing geographical locations from DNS names“. Advised by Quirin Scheitle, Oliver Gasser. Interdisciplinary project. Technical University of Munich, Mar. 2018.

[215] Quirin Scheitle, Taejoong Chung, Jens Hiller, Oliver Gasser, Johannes Naab, Roland van Rijswijk-Deij, Oliver Hohlfeld, Ralph Holz, Dave Choffnes, Alan Mislove, and Georg Carle. „A First Look at Certification Authority Authorization (CAA)“. In: *ACM SIGCOMM Computer Communication Review* 48.2 (Apr. 2018), pp. 10–23.

[216] Quirin Scheitle, Oliver Gasser, Patrick Sattler, and Georg Carle. „HLOC: Hints-Based Geolocation Leveraging Multiple Measurement Frameworks“. In: *Proceedings of the 2017 Network Traffic Measurement and Analysis Conference*. IFIP. Dublin, Ireland, June 2017.

[217] Quirin Scheitle, Oliver Gasser, Minoo Rouhi, and Georg Carle. „Large-Scale Classification of IPv6-IPv4 Siblings with Variable Clock Skew“. In: *Proceedings of the 2017 Network Traffic Measurement and Analysis Conference*. IFIP. Dublin, Ireland, June 2017.

[218] Quirin Scheitle, Oliver Gasser, Theodor Nolte, Johanna Amann, Lexi Brent, Georg Carle, Ralph Holz, Thomas C. Schmidt, and Matthias Wählisch. „The Rise of Certificate Transparency and Its Implications on the Internet Ecosystem“. In: *Proceedings of the 2018 Internet Measurement Conference*. ACM. Boston, MA, USA, Nov. 2018, pp. 343–349. ISBN: 978-1-4503-5619-0. DOI: `10.1145/3278532.3278562`.

[219] Quirin Scheitle, Matthias Wählisch, Oliver Gasser, Thomas C Schmidt, and Georg Carle. „Towards an Ecosystem for Reproducible Research in Computer Networking“. In: *Proceedings of the 2017 SIGCOMM Reproducibility Workshop*. ACM. Los Angeles, CA, USA, Aug. 2017, pp. 5–8.

[220] Johann Schlamp, Ralph Holz, Oliver Gasser, Andreas Korsten, Quentin Jacquemart, Georg Carle, and Ernst W. Biersack. „Investigating the Nature of Routing Anomalies: Closing in on Subprefix Hijacking Attacks“. In: *Proceedings of the 2015 Workshop on Traffic Monitoring and Analysis*. Springer. Barcelona, Spain, Apr. 2015.

[221] Frank Schmidt. „Evaluation of Trust Relationships in the Domain Name System“. Advised by Johannes Naab, Oliver Gasser. Master's thesis. Technical University of Munich, June 2017.

[222] Nadja Schricker. „Active Security Evaluation with Network Scans“. Advised by Oliver Gasser, Quirin Scheitle. Bachelor's thesis. Technical University of Munich, Oct. 2016.

[223] Alexander Schulz. „Identification of IPv6-IPv4 Sibling Pairs from Passive Observations“. Advised by Quirin Scheitle, Oliver Gasser, Minoo Rouhi. Bachelor's thesis. Technical University of Munich, July 2017.

[224] Shodan. *Map of Industrial Control Systems on the Internet*. URL: `https://icsmap.shodan.io/`.

[225] Ryan Sleevi. *Certificate Transparency in Chrome - Change to Enforcement Date – Google Groups*. Apr. 21, 2017. URL: `https://groups.google.com/a/chromium.org/forum/#!msg/ct-policy/sz_3W_xKBNY/6jq2ghJXBAAJ`.

[226] Ryan Sleevi and Eran Messeri. *Certificate Transparency in Chrome: Monitoring CT Logs Consistency*. May 1, 2015. URL: `https://docs.google.com/document/d/1FP5J5Sfsg0OR9P4YT0q1dM02iavhi8ix1mZlZe_z-ls`.

[227] Victor Sosa. „Large-Scale Flow Collection“. Advised by Oliver Gasser. Interdisciplinary project. Technical University of Munich, May 2016.

[228] Markus Sosnowski. „Internet-Wide Assessment of TCP Options“. Advised by Quirin Scheitle, Oliver Gasser, Minoo Rouhi, Paul Emmerich, Dominik Scholz. Bachelor’s thesis. Technical University of Munich, July 2017.

[229] Spamhaus. *The Spamhaus Project*. URL: `https://www.spamhaus.org`.

[230] Emily Stark. *Expect-CT Extension for HTTP*. Internet-Draft draft-ietf-httpbis-expect-ct-02. URL: `https://tools.ietf.org/html/draft-ietf-httpbis-expect-ct-02`.

[231] Statvoo. *Statvoo Top Websites*. URL: `https://statvoo.com/top`.

[232] Stephen D Strowes. „Bootstrapping Active IPv6 Measurement with IPv4 and Public DNS“. In: *arXiv preprint arXiv:1710.08536* (2017).

[233] Christian Sturm. „Detection of malicious packets through inbound Time to Live headers“. Advised by Quirin Scheitle, Oliver Gasser, Felix von Eye. Bachelor’s thesis. Technical University of Munich, Dec. 2015.

[234] Symantec. *Update on Test Certificate Incident*. 2016. URL: `https://www.symantec.com/page.jsp?id=test-certs-update`.

[235] Paulin Tchonin. „TTL Analysis for DDoS Defense“. Advised by Quirin Scheitle, Oliver Gasser, Paul Emmerich. Master’s thesis. Technical University of Munich, Oct. 2016.

[236] TechCrunch. *Network Solutions Hijacking Unassigned Sub-Domains*. Apr. 8, 2008. URL: `https://techcrunch.com/2008/04/08/network-solutions-hijacking-unassigned-sub-domains/`.

[237] TUM. *cablint*. URL: `https://github.com/tumi8/certlint`.

[238] TUM. *Certificate Transparency Go Tool*. URL: `https://github.com/google/certificate-transparency-go`.

[239] TUM. *Forked crypto/ssh package used by goscanner*. URL: `https://github.com/tumi8/ssh`.

[240] TUM. *Forked crypto/tls package used by goscanner*. URL: `https://github.com/tumi8/tls`.

[241] TUM. *goscanner*. URL: https://github.com/tumi8/goscanner.

[242] TUM. *IPv6 Hitlist Collection*. URL: https://www.net.in.tum.de/projects/gino/ipv6-hitlist.html.

[243] TUM. *Modified ZMap version for IPMI measurements*. URL: https://www.net.in.tum.de/pub/zmap/zmap-ipmi.tar.gz.

[244] TUM. *ZMapv6*. URL: https://github.com/tumi8/zmap.

[245] TUM. *ZMapv6: Add IPv6 capability including probe modules*. URL: https://github.com/tumi8/zmap/commit/1f34faa3f14e3924d6c5790d758d19952482d43c.

[246] TUM. *ZMapv6: Add stdin option for ipv6-target-file parameter*. URL: https://github.com/tumi8/zmap/commit/39fdfef55e1e149c1a3543111130eabd388ef686.

[247] TUM. *ZMapv6: IPv6: add scanning address to pcap filter*. URL: https://github.com/tumi8/zmap/commit/7c1fbda1ac642092b330403e08ab5c87895e5b5a.

[248] Tune The Web. *OPINION - Dangerous Web Security Features*. Sept. 2015. URL: https://www.tunetheweb.com/blog/dangerous-web-security-features/.

[249] Johanna Ullrich, Peter Kieseberg, Katharina Krombholz, and Edgar Weippl. „On Reconnaissance with IPv6: A Pattern-Based Scanning Approach“. In: *Proceedings of the 2015 Conference on Availability, Reliability and Security*. IEEE. Toulouse, France, Aug. 2015, pp. 186–192.

[250] Benjamin VanderSloot, Johanna Amann, Matthew Bernhard, Zakir Durumeric, Michael Bailey, and J Alex Halderman. „Towards a Complete View of the Certificate Ecosystem“. In: *Proceedings of the 2016 Internet Measurement Conference*. ACM. Santa Monica, CA, USA, Nov. 2016.

[251] Verisign. *Domain Name Industry Brief Q1'17*. July 2017. URL: https://www.verisign.com/assets/domain-name-report-Q12017.pdf.

[252] Stefan Viehböck. *House of Keys: Industry-Wide HTTPS Certificate and SSH Key Reuse Endangers Millions of Devices Worldwide*. SEC Consult. URL: http://blog.sec-consult.com/2015/11/house-of-keys-industry-wide-https.html.

[253] ViewDNS. *ccTLD Zone Files / Domain Name Lists*. 2017. URL: http://viewdns.info/data/.

[254] Till Wickenheiser. „Correlation of TTL data to network characteristics“. Advised by Quirin Scheitle, Oliver Gasser. Bachelor's thesis. Technical University of Munich, Apr. 2016.

[255] WIDE Project. *MAWI Working Group Traffic Archive*. URL: http://mawi.wide.ad.jp/mawi/.

[256] Katharina Wiegräbe. „Identifying Web-enabled Devices on Internet Paths". Advised by Minoo Rouhi, Dominik Scholz, Oliver Gasser, Quirin Scheitle. Bachelor's thesis. Technical University of Munich, Aug. 2017.

[257] Addy Yeow. *Bitnodes*. URL: `https://bitnodes.earn.com/`.

[258] Scott Yilek, Eric Rescorla, Hovav Shacham, Brandon Enright, and Stefan Savage. „When Private Keys Are Public: Results from the 2008 Debian OpenSSL Vulnerability". In: *Proceedings of the Internet Measurement Conference*. ACM. Chicago, IL, USA, Nov. 2009.

[259] Sebastian Zander, Lachlan LH Andrew, and Grenville Armitage. „Capturing Ghosts: Predicting the Used IPv4 Space by Inferring Unobserved Addresses". In: *Proceedings of the 2014 Internet Measurement Conference*. ACM. Vancouver, BC, Canada, Oct. 2014, pp. 319–332.

[260] ZGrab authors. *ZGrab*. URL: `https://github.com/zmap/zgrab`.

[261] Jing Zhang, Zakir Durumeric, Michael Bailey, Mingyan Liu, and Manish Karir. „On the Mismanagement and Maliciousness of Networks". In: *Proceedings of the 2014 Symposium on Network and Distributed System Security*. Internet Society. San Diego, CA, USA, Feb. 2014.

[262] Liang Zhang, David Choffnes, Dave Levin, Tudor Dumitras, Alan Mislove, Aaron Schulman, and Christo Wilson. „Analysis of SSL Certificate Reissues and Revocations in the Wake of Heartbleed". In: *Proceedings of the 2014 Internet Measurement Conference*. ACM. Vancouver, BC, Canada, Oct. 2014.

[263] Zhechko Zhechev. „Asymmetric Route Detection using Return TTLs". Advised by Oliver Gasser, Quirin Scheitle. Bachelor's thesis. Technical University of Munich, Feb. 2016.

[264] Liang Zhu, Johanna Amann, and John Heidemann. „Measuring the Latency and Pervasiveness of TLS Certificate Revocation". In: *Proceedings of the 2016 Passive and Active Measurement Conference*. Springer. Heraklion, Greece, Apr. 2016. DOI: `10.1007/978-3-319-30505-9_2`.

ISBN 978-3-937201-65-8
9 783937 201658

ISBN 978-3-937201-65-8
DOI 10.2313/NET-2019-05-1

ISSN 1868-2634 (print)
ISSN 1868-2642 (electronic)