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

An Evaluation of Architectural Threats to Internet Routing

Johann Schlamp

Dissertation

Network Architectures and Services

NET-2016-07-2

TECHNISCHE UNIVERSITÄT MÜNCHEN
Institut für Informatik
Lehrstuhl für Netzarchitekturen und Netzdienste

An Evaluation of Architectural Threats to Internet Routing

Johann Schlamp

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

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. Jörg Ott
Prüfer der Dissertation:	1. Univ.-Prof. Dr.-Ing. Georg Carle
	2. Prof. Dr. Ernst W. Biersack

Die Dissertation wurde am 05.04.2016 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 04.07.2016 angenommen.

Cataloging-in-Publication Data

Johann Schlamp
An Evaluation of Architectural Threats to Internet Routing
Dissertation, July 2016

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

ISBN 978-3-937201-53-5
DOI 10.2313/NET-2016-07-2

ISSN 1868-2634 (print)
ISSN 1868-2642 (electronic)

Network Architectures and Services NET-2016-07-2
Series Editor: Georg Carle, Technical University of Munich, Germany

ABSTRACT

The Internet is an integral part of today's way of life and a critical business infrastructure at the same time. Since its early beginnings, resilience has been embedded deeply into the fabric of Internet routing. It is highly resistant against random outages and, in theory, immune to major blackouts. To sustain this quality, routers constantly exchange network reachability information to identify cost-efficient routes and to re-route around failures. A distributed routing table updated by individual peers serves to disseminate the necessary information. In practice, this process lacks a reliable route validation scheme and is based on mutual trust. As a consequence, any Internet participant can, accidentially or deliberately, advertise false routes and globally attract traffic destined to arbitrary networks.

In April 1997, a minor operating error committed by a small Internet service provider led to the first global Internet failure. Since then, much effort has been invested to study and prevent such accidential events. A huge body of related work emerged to detect and analyze attacks that aim to disrupt connectivity of individual networks. However, state-of-the-art techniques focus on common attack scenarios and mostly neglect more sophisticated variants. Even though a comprehensive solution to secure Internet routing has been under development for nearly a decade, its completion and adoption is not yet on the horizon.

In this thesis, we address research questions to evaluate the actual threat of routing attacks. We establish a solid background on Internet operations and derive a rigorous routing model to study different types of attacks. To close the detection gap in related work, we develop novel techniques that draw on a variety of data sources. More importantly, we apply these concepts under realistic conditions and assess the risk potential of different attacks in great detail. We learn that there is a real threat and conduct forensic case studies on specific incidents to put the results into perspective. Based on a rich set of lessons learned, we design a comprehensive framework to monitor the global routing system in real-time.

The work presented with this thesis offers great potential for future development. It opens up new research directions towards a reliable mitigation of attacks and can support the development of a secure routing architecture for the Internet. At the same time, the presented routing model is of general nature and applicable to a broader area of research. This thesis may thus foster new and innovative techniques to analyze Internet routing.

ZUSAMMENFASSUNG

Das Internet ist ein zentraler Bestandteil der heutigen Gesellschaft und bildet eine kritische Infrastruktur für nahezu alle Wirtschaftssektoren. Eine wesentliche Eigenschaft des Routings im Internet ist dessen Ausfallsicherheit. Entsprechende Routing-Protokolle sind resistent gegen zufällige Störungen und schließen flächendeckende Ausfälle praktisch aus. Router tauschen in diesem Zusammenhang kontinuierlich Informationen über die Erreichbarkeit von Netzwerken aus, um kostengünstige Routen zu identifizieren und defekte Routen zu umgehen. Für diesen Zweck wird eine verteilte Routing-Tabelle aufgebaut und von einzelnen Teilnehmern individuell aktualisiert. In der Praxis fehlt diesem auf gegenseitigem Vertrauen basierenden Prozess jedoch eine zuverlässige Möglichkeit zur Validierung von aktuellen Routen-Informationen. Dementsprechend können Routing-Teilnehmer globale Verkehrsströme zu beliebigen Netzwerken versehentlich oder mit Vorsatz anziehen.

Im April 1997 führte ein geringfügiger Fehler bei einem kleineren Internetanbieter zur ersten weltweiten Störung des Internets. Seitdem wurden viele Sicherheitsvorkehrungen eingeführt und umfangreiche Arbeiten zur Erkennung und Analyse von gezielten Angriffen durchgeführt. Der Stand der Technik beschränkt sich allerdings auf einfache Angriffsszenarien und vernachlässigt fortschrittlichere Varianten. Trotz der seit einem Jahrzehnt andauernden Arbeiten an einer umfassenden Sicherheitslösung für das Internet-Routing zeichnet sich dessen Fertigstellung und flächendeckender Einsatz in näherer Zukunft nicht ab.

In der vorliegenden Arbeit werden Forschungsfragen zur Bedrohung des Internets durch Routing-Angriffe behandelt. Mit Hilfe eines präzisen Routing-Modells und unter Berücksichtigung gängiger Betriebspraxis werden verschiedene Angriffstypen diskutiert. Auf Basis zahlreicher Datenquellen werden neuartige Erkennungstechniken entwickelt und dabei Defizite verwandter Arbeiten behoben. Durch deren Einsatz in der Praxis und ergänzt durch spezifische Fallstudien wird das tatsächliche Gefährdungspotential bestimmt. Daraus resultiert der Entwurf eines kombinierten Systems zur Echtzeitanalyse des Internet-Routings.

Die erarbeiteten Konzepte bieten weitreichendes Potential für zukünftige Forschung. Neben neuen Ansätzen zur Bekämpfung von Angriffen können diese auch zum Aufbau einer sicheren Routing-Architektur beitragen. Gleichzeitig eignet sich das präsentierte Routing-Modell als allgemeiner Ansatz auch für die Entwicklung neuartiger Routing-Analysen.

ACKNOWLEDGEMENTS

I would like to express my gratitude to many individuals who supported my work over the years. First and foremost, I am very grateful to my supervisor Prof. Dr.-Ing. Georg Carle. You allowed me all the freedoms one could possibly wish for, and you gave me advice for whatever issues that bothered my mind. It was an exciting period at your chair, in which I learned a great deal about work, life, and not least about myself. I would equally like to thank Prof. Dr. Ernst W. Biersack for his continuous guidance. You inspired me in many ways, and I greatly appreciate your challenging support, not to mention your tolerance for my deadline-driven working style. I very much enjoyed my stays with you in Southern France.

I had great colleagues at the Technical University of Munich, who significantly influenced my work. Möpi (a.k.a Dr.-Ing. Stephan M. Günther), you showed me that there is always a more rigorous solution and that perfection is a quest approaching infinity. I really enjoyed our mathematical excursions. Dr. Lothar Braun, I learned a great deal from your most efficient time management. No matter how much work you had on the desk, you always delivered right on time showing off an astounding calm. Dr. Ralph Holz, you taught me much about upholding scientific integrity. With Dr. Marc Fouquet and Dr. Heiko Niedermayer, I had exemplary advisors for my diploma thesis. You encouraged me to stay in academia. Last, but not least, I took great pleasure and benefit from supervising the work of numerous students. Please bear with me if I pushed you too hard sometimes. A special thanks is due to Leonhard Rabel, who spent countless hours proof-reading this thesis.

I also appreciate the support of fellow researchers at other institutes. Prof. Dr. Matthias Wählisch, thank you for constantly challenging my ideas. I will fondly recall our heated debates on technical and culinary matters. Prof. Dr. Thomas C. Schmidt, I enjoyed the serenity and professionalism you brought into our discussions. I further want to thank Dr. Pierre-Antoine Vervier and Dr. Quentin Jacquemart for providing me with valuable background.

I am very grateful to my parents for giving me unconditional support at every stage of my life. My fellow Schafkopf guys, you constantly provided for the distraction I needed after long weeks of technical work. And to my love Sarah, thank you for being so patient with me all the time and for showing me a life beyond work. You truly are the best part of me.

Garching, December 2016

Johann Schlamp

TABLE OF CONTENTS

1 CHAPTER ONE
Introduction

The Internet can be considered the largest and most complex system that has ever been built. Comprising millions of hosts [10] and billions of users [11], there is no comparable development in human history. With its forthcoming expansion to physical objects, the Internet of things [12] may soon approach the complexity of the human brain [13, 14]. Given its brief but explosive history, future evolutions for the next decades can hardly be foreseen. The Internet is a *disruptive technology* with massive impact on markets [15], society [16], and politics [17]. It successfully withstood attempts to shut down nation-wide [18] and played an important role in the fall of governments [19]. Online connectivity has become ubiquitous [20, 21] and the Internet a critical asset [22] to protect. In this regard, it is surprising to see that global communication is heavily tied to a routing protocol that was first developed in 1982 [23] and is, even more astonishingly, based on mutual trust.

Internet routing today is in a constantly shifting state [24] due to temporary network outages, load balancing efforts, and a growing number of new entrants [25, 26]. Network routes need to be continually adjusted to reflect such changes. As an open federation of networks under independent control, the Internet is neither steered by a central authority, nor does a global map of available network routes exist. The *Border Gateway Protocol* [27], a distributed routing protocol, enables routers to select optimal routes, e.g. shortest or most cost-efficient routes, by disseminating the necessary update information. Each participant advertises a local view of network reachability to his neighbors, who redistribute corresponding routes to theirs. But without having a reliable route validation scheme, routing decisions are generally based on unverified information. As a consequence, an attacker can easily inject manipulated routes into the routing system to globally attract traffic. In 2008, for instance, Pakistan Telecom began advertising a favorable route to the YouTube network [28]. Within a few minutes, and lasting for several hours, major parts of the Internet directed their YouTube traffic towards Pakistan, thereby effectively disrupting its service.

In this thesis, we address research questions regarding such architectural threats to Internet routing. We first study current network operations and derive a comprehensive at-

tacker model. Then, we discuss different techniques to discover routing attacks by leveraging a variety of unique data sets. We consequently evaluate the actual threat imposed by routing attacks in practice and conduct forensic case studies to obtain a broader view on the potential risk. Our insights lead us to the design of a comprehensive monitoring framework to assess routing attacks in real-time. The work presented in this thesis is an important step to provide network operators with the necessary tools to counter attacks. More importantly, this work can support the development of a secure interdomain routing system in the future. It further opens up new perspectives on Internet routing analysis in general.

During his research, the author significantly advanced the state-of-the-art on attack detection and regularly contributed to the scientific community. Most notably, he published articles in the high-ranking journals *IEEE Journal on Selected Areas in Communications (JSAC)* [8] and *ACM SIGCOMM Computer Communication Review (CCR)* [4]. His conference paper at the *International Workshop on Traffic Monitoring and Analysis (TMA)* [7] won the best paper award. The author further submitted a paper to the top-tier journal *IEEE/ACM Transactions on Networking (TON)* [9], which is pending for review.

2 CHAPTER TWO
Problem Statement

It is a well-known fact that interdomain routing exhibits shortcomings with respect to security. Practical experience shows that sudden outages [29, 30] may be the result. In addition, anecdotal evidence [28, 31, 32] exists that attackers deliberately exploit flaws in the interdomain routing protocol. In this respect, invalid route updates can be injected into the routing system to illegitimately attract traffic. The prevalence of such attacks, however, is still not fully perceived. It is thus a worthwhile task to evaluate the actual threat in thorough detail. To this end, the architectural concepts behind Internet routing need to be put under close scrutiny. More importantly, novel techniques have to be developed to conduct empirical analyses, which may help to assess the risk potential in practice. The ultimate goal of this work is to foster and support the development of a secure routing system for the future of the Internet. In the following, we dissect this long-term task into several research objectives and discuss them in more detail. At the end of this chapter, we revisit the author's scientific contributions and discuss the outline of this thesis.

2.1 Research Objectives

The author's work is structured around three major objectives. First, a thorough analysis of the problem statement with respect to flaws in Internet routing is to be carried out. A formalized routing model can help to rigorously study different types of attacks and to infer an attacker's tactics. Second, routing attacks that are not yet addressed by previous work need to be identified. To close the detection gap, innovative techniques that take into account data from various sources are to be developed. Finally, these detection systems have to be deployed in practical environments. In this respect, a thorough evaluation of ongoing attacks as well as studies on incidents from the past are to be carried out. Based on lessons learned from operational practice, a comprehensive framework can be derived to continuously monitor the routing landscape.

O–1 Security Flaws in Interdomain Routing

In the literature, circumstancial evidence indicates that attacks on interdomain routing are feasible and carried out in practice. To obtain a broader picture, several aspects of today's Internet routing have to be taken into account. First, current practices regarding the management of Internet resources by non-profit registries are to be studied. Second, shortcomings in the design of the Border Gateway Protocol (BGP) need to be assessed. To this end, its historic origins as well as previously proposed extensions to improve security have to be considered. A comprehensive attacker model is to be devised that accounts for an attacker's motivation and for the topological changes imposed by attacks. With such a model, strengths and weaknesses of state-of-the-art detection techniques can be evaluated.

O-1.1 *Studying the management of Internet resources.*
To establish a basis for the evaluation of threats to Internet routing, it is helpful to study the architecture of the Internet with respect to the management of Internet resources. Based on current practices and implemented policies regarding the allocation and utilization of Internet resources, suitable data sources for subsequent analyses can be identified.

O-1.2 *Analyzing shortcomings in the Border Gateway Protocol.*
An analysis of the design principles of BGP in due consideration of operational aspects enables the assessment of inherent security flaws. Despite the fact that protocol extensions that aim to improve security may not bet widely adopted, studying such concepts can complement the initial findings leading to a new set of techniques for the detection and mitigation of attacks.

O-1.3 *Devising a comprehensive attacker model.*
Based on a thorough evaluation of BGP, an elaborate model to formalize Internet routing is to be devised. This process serves to develop a comprehensive attacker model that allows for the anticipation of new attack vectors. By taking into account the motivation behind attacks, eligible attack tactics can be infered.

O-1.4 *Assessing state-of-the-art attack detection.*
A huge body of related work exists on common routing attacks. However, attack variants that prove to be more beneficial for an attacker have been neglected for the largest part. Based on an exhaustive attacker model, corresponding detection techniques need to be assessed and classified according to their applicability.

O–2 Detection and Analysis of Routing Attacks

State-of-the-art approaches to detect routing attacks do not tap the full potential with respect to leveraged data sources and corresponding analysis techniques. As a matter of fact, specific types of attacks are still beyond the scope of most techniques. It is thus worthwile to look into new directions to complete the set of analysis tools. To this end, it is imperative to draw on orthogonal data sources in order to narrow down the search space for routing attacks and to provide rich evidential data for the investigation of suspicious events. In addition, it appears reasonable to question continued and hardened assumptions about the analysis of topological characteristics of the Internet.

O-2.1 *Assessing legitimate routing anomalies.*
The global routing system regularly exhibits a large number of routing anomalies. In related work, these anomalies often serve to infer potential candidates for an attack. It is an unsolved task, however, to reduce high rates of false positives. Hence, new concepts to assess the legitimacy of routing anomalies need to be developed, which ideally leverage orthogonal data sources like active measurements and publicly available databases on Internet resources. Supplemental data such as traffic recordings can be supportive to such analyses as well.

O-2.2 *Uncovering path manipulation and traffic interception.*
It is common belief that graph data structures are best suitable for modeling the Internet topology. Consequently, most control-plane techniques to detect routing attacks are based on such models. These approaches, however, are not capable to detect the manipulation of routing policies. To improve on this situation, new techniques should go beyond common graph analyses in order to uncover more elaborate incidents like man-in-the-middle attacks.

O-2.3 *Identifying abandoned Internet resources.*
In literature, routing attacks are mostly addressed from a technical point of view. However, attackers may also operate on an administrative level in order to take ownership of a victim's Internet resources. Such attacks can be enabled by inactive victims leaving behind abandoned resources. To assess the actual threat, the utilization of Internet resources needs to be studied in detail. As a result, conclusive activity metrics can be derived in order to identify and monitor idle victims, and their vulnerable resources respectively.

O–3 Empirical Threat Analyses

With a new set of detection techniques, the imminent threat to Internet routing can be assessed. To this end, empirical analyses need to be conducted in order to identify the risk potential and to study individual incidents. Thorough case studies are helpful to grasp operational details of different types of attacks. Based on the lessons learned from such analyses, a combined real-time monitoring framework can be drafted for continuous operation. A conclusive design that is capable to connect various detection schemes offers great potential for future work. The knowledge gained with this thesis can be used to develop new and ambitious research directions in order to reliably prevent routing attacks in the future.

O-3.1 *Looking for evidence that attacks on interdomain routing occur.*
By applying the newly developed analysis techniques under realistic conditions, their performance can be analyzed and the actual threat posed by routing attacks can be assessed. Ideally, further insights into suspicious routing anomalies can be gained and discussed in greater detail in order to put the threat into context. In this respect, a thorough evaluation of evidence for and against attacks is to be carried out and lessons learned should be derived.

O-3.2 *Designing a real-time monitoring framework.*
In order to continuously monitor threats to Internet routing, an extensible detection system to combine both existing and future approaches is to be developed. To this end, requirements and intended analysis capabilities need to be discussed and supported by a comprehensive system architecture that allows for flexible integration of various detection techniques. Conclusive findings resulting from continuous operation of such a system might establish a basis for researchers to secure the interdomain routing system in the future.

2.2 Contributions and Outline

This thesis presents the author's contributions in the area of modeling, detection, and analysis of routing attacks. Major results have been peer-reviewed in prior publications. In the following, the author's scientific contributions that constitute this thesis are presented and aligned with the research objectives discussed above. A detailed overview of the structure of this thesis is given at the end of this section.

2.2.1 Prior Publications

Significant parts of this thesis have been previously published, either in conference proceedings or in scientific journals. A complete list of the author's prior work related to this thesis can be found in chronological order in the List of Prior Publications. These publications comprise nine scientific papers with major relevance to the detection and analysis of routing attacks. In [1], a methodology to estimate the geographic location of arbitrary systems connected to the Internet is devised. A framework to study and visualize generic traffic flows is presented in [2]. Both systems can generally help with studying attacks in more detail. These publications, however, are not directly related to the evaluation of threats posed by routing attacks, and are thus of supplemental nature. Besides, the author's contribution was secondary in each case. For reference purposes, the original publications are attached to this thesis in Appendix A and Appendix B.

The work presented in [3] and [4] focuses on a forensic case study of a specific routing attack, and discusses lessons learned for future detection and prevention of such attacks. These publications are original work of the author. For the collaborative detection system presented in [5], the author contributed a technique for traffic analysis and means to study the utilization of Internet resources from a management perspective. An approach to identify abandoned Internet resources derived and realized by the author is published in [6]. His co-authors provided support with respect to methodological refinement and writing, and played an important role in outlining future research directions. Another collaborative system to assess the legitimacy of routing anomalies is given with [7]. The author contributed significant parts of methodology and implementation, in particular with respect to using SSL/TLS measurements and resource-based inference rules to assess suspicious routing events. His co-authors contributed a scanning tool for SSL/TLS hosts and a topology-based inference algorithm. It is worth mentioning that this paper received the *best paper award* after presentation by the author at the conference venue. Consequently, the author strived to extend the approach with a formalized routing model to classify attacks and conducted more detailed analyses based on richer data sets. His co-authors broadened their contributions as well. An accordingly extended paper is published in a high-ranking journal [8]. The author subsequently transfered his formal routing model into practice by developing so-called route automata. These highly efficient data structures allow for studying Internet routing in a rigorous way and were applied to evaluate route leaks and interception attacks. Supported in writing by his co-authors, the author submitted a paper to a high-ranking journal [9].

2.2.2 Structure of this Thesis

Subsequent chapters begin with a note on prior publication as indicated below, and, if applicable, contain further details about changes that were made to the original publication:

> **NOTE** *This chapter contains prior publication.*

Each chapter further provides a short **SUMMARY** including an outline to get acquainted with the contents presented thereinafter. The thesis is structured as follows.

Part I provides relevant background knowledge, a comprehensive attacker model, and an assessment of related work (**Objective O-1**). Chapter 3 starts with a constructive analysis of current practices for the management of Internet resources and evaluates publicly available data sources *(Objective O-1.1)*. A detailed treatment of security flaws in the Border Gateway Protocol in a historical context including recently proposed security extensions is presented in Chapter 4 *(Objective O-1.2)*. In Chapter 5, a novel model for Internet routing is derived based on concepts of formal languages, which leads to a classification of attacks and a discussion of suitable attack tactics *(Objective O-1.3)*. Related work regarding empirical studies on routing anomalies and techniques for detecting attacks is discussed and assessed in Chapter 6 *(Objective O-1.4)*.

Part II presents three distinct techniques that are suitable to discover different types of routing attacks (**Objective O-2**). With Chapter 7, an elaborate approach to assess the legitimacy of routing anomalies based on the combination of multiple data sources is developed *(Objective O-2.1)*. Chapter 8 introduces a novel concept to study routing data, which is applied to the detection of man-in-the-middle attacks *(Objective O-2.2)*. A technique to identify abandoned Internet resources in order to anticipate potential victims of a hidden takeover attack is derived in Chapter 9 *(Objective O-2.3)*.

Part III evaluates the presented detection techniques in practice, provides insights into real-world incidents, and outlines a combined concept for a real-time framework to continuously monitor the threat of routing attacks (**Objective O-3)**. Chapters 10 to 12 document the empirical evaluation of individual detection techniques *(Objective O-3.1)*. Given the insights from operational practice, the design of a flexible monitoring framework that is capable to incorporate various detection schemes is derived in Chapter 13 *(Objective O-3.2)*. It includes a detailed description of requirements and analysis capabilities as well as an implementation roadmap for future work.

Objective	**Contributions of this thesis**			*Analysis*	*Modeling*	*Detection*	*Evaluation*
O-1	**Part I**	*Background and Theory*					
O-1.1	p. 13	Chapter 3	Management of Internet Resources	X			
O-1.2	p. 25	Chapter 4	The Border Gateway Protocol	X			
O-1.3	p. 33	Chapter 5	A Comprehensive Attacker Model		X	X	
O-1.4	p. 49	Chapter 6	Related Work and State-of-the-Art	X			
O-2	**Part II**	*Discovering Routing Attacks*					
O-2.1	p. 61	Chapter 7	Legitimate Routing Anomalies			X	X
O-2.2	p. 79	Chapter 8	Interception and Path Manipulation	X	X	X	
O-2.3	p. 93	Chapter 9	Abandoned Internet Resources			X	X
O-3	**Part III**	*Monitoring the Threat*					
O-3.1	p. 107	Chapter 10	Evaluation of *HEAP*		X		X
O-3.1	p. 133	Chapter 11	Evaluation of *CAIR*		X	X	X
O-3.1	p. 147	Chapter 12	Evaluation of *PHEW*				X
O-3.2	p. 157	Chapter 13	A Real-Time Monitoring System			X	
	Part IV	*Discussion and Conclusion*					
	p. 167	Chapter 14	Author's Contributions				
	p. 173	Chapter 15	Conclusion and Perspectives				

Table 2.1: Structure and contributions of this thesis.

Part IV provides a retrospect on the author's accomplishments and scientific contributions including a comparison with the state-of-the art in Chapter 14. An outlook on future research directions set forth by the author's work in Chapter 15 concludes the thesis.

Table 2.1 presents the structure of this thesis in relation to the research objectives as discussed in Section 2.1. The table shows the methodological approach behind each of the tasks that were addressed by the author's work. *Analysis* refers to work on technical background with respect to the problem statement in order to provide a solid basis for subsequent improvements. The term *Modeling* subsumes novel work to formalize and classify attacks in order to establish a comprehensive threat model. The development of innovative techniques to identify different types of routing attacks and the resulting design of a combined monitoring system is refered to as *Detection*. Empirical studies to assess the risk potential of routing attacks including the conduction of individual case studies are summarized under the term *Evaluation*.

PART ONE

Background and Theory

3 CHAPTER THREE
Management of Internet Resources

NOTE *This chapter does not contain prior publication.*

SUMMARY The following analysis provides background knowledge on Internet resources. Section 3.1 discusses the role of the non-profit Internet Assigned Numbers Authority in the distribution of such resources. It is hierarchically organized, and governed by a community-driven approach. A different type of authority, namely trusted third parties, which issue cryptographic Internet resources, is discussed in Section 3.2. A variety of network protocols make use of corresponding trusted certificates. In order to gather evidential data for the further course of this thesis, public databases providing supplemental information are evaluated in Section 3.3. Moreover, the measurement-based collection of additional data sets is discussed. Practical examples for obtainable data sets are presented and explained.

3.1 Internet Assigned Numbers Authority

The distribution of Internet resources began with a single individual handing out network-related numbers on an as-needed basis. In 1972, Jon Postel, working at the University of Southern California, started to assign socket numbers for an early host-to-host communication protocol [33]. Soon, his efforts comprised the stewardship for a variety of network protocols and, with the Internet Protocol (IP) emerging, ultimately spanned the assignment of IP addresses in 1981 [34]. At that time, a total of 47 class A networks were already assigned to governmental and academical institutions. A decade later, hundreds of class B networks followed, and Postel's function evolved into a non-profit corporation that already had come to be known as IANA [35].

Today, IANA is a department of the non-profit Internet Corporation for Assigned Names and Numbers (ICANN). Initially established by the U.S. Department of Commerce, ICANN quickly became an international, community-driven institution. Based on a democratic process, ICANN now governs the administration of globally unique names and numbers. Executive power lies with IANA [36], whose key functions still comprise the management of number resources, domain names, and protocol parameters [37]. In addition to allocation tasks, IANA also implements technical standards as specified by the Internet Engineering Task Force (IETF) and the Internet Research Task Force (IRTF) [36]. For instance, IANA maintains the reserved address space for IP multicast [38] as well as special-purpose addressing schemes like private networks [39].

3.1.1 Regional Internet Registries

While IANA represents the top of the hierarchy for number resource allocation, five so-called Regional Internet Registries (RIRs) have been formed to partition responsibility. The American Registry for Internet Numbers (ARIN) administers North America and parts of the Caribbean region. Latin America and the remaining parts of the Caribbean region are managed by the Latin America and Caribbean Network Information Centre (LACNIC). The Asia-Pacific region is served by the Asia Pacific Network Information Centre (APNIC). European number resources are managed by the Réseaux IP Européens (RIPE) Network Coordination Centre. The African Network Information Center (AfriNIC) serves Africa and the Indian Ocean region. Meanwhile, all five RIRs joined forces in the Number Resource Organization (NRO), a non-profit coordinating body that represents its members' interests towards ICANN.

Each RIR develops individual number assignment policies in accordance to the needs of its geopolitical region with a consensus-based approach. While ARIN has practically depleted its pool of IPv4 addresses at the time of writing, it is assumed, in contrast, that RIPE is able to sustain the supply of IPv4 resources for new market entrants for at least several more years due to more restrictive policies. In the light of the imminent depletion of the IPv4 address pool, some RIRs operate an IP address transfer listing service for its members. Beside IP addresses, RIRs further assign numbers for autonomous systems (ASes) and delegate reverse DNS zones within the `.arpa` domain, which is in turn centrally administered by IANA.

Organizations that receive number resource allocations like blocks of IP addresses are called Local Internet Registries (LIRs). For the most part, these organizations are Internet service providers (ISPs) and governmental or academical institutions. The intended purpose of an LIR is to assign number resources to customers, i.e. to other ISPs or end users. While it is mandatory for LIRs to be a member of an RIR and to adhere to assignment policies, LIRs have sole responsibility for the administration of their allocated resources. At the time of writing, RIPE, for instance, counts more than 11,500 members from 76 countries.

3.1.2 Network Information Centers

IANA is also responsible for maintaining the root zone database for the Domain Name System (DNS) [40]. This database holds a list of all generic and country-code top-level domains and information about their delegation to designated manager bodies [41]. Nowadays, these managers are often called registry operators or Network Information Centers (NICs). Similar to RIRs, a NIC's purpose is to allocate second-level domains to their members. DENIC, the German registry operating the `.de` top-level domain, has more than 300 members at the moment, mostly comprising ISPs and hosting providers. These members have direct access to DENIC's registration system and provide domain registration services to other organizations or end users. Some NICs only registrate compulsory third or even fourth-level domains. The British registry Nominet UK, for instance, required new domains to be registered under specific second-level domains like `.co.uk` up until recently.

IANA further maintains an authoritative list of third-party operated DNS root servers and a list of name servers for all top-level domains, including infrastructure-related domains like `.arpa` and other special-purpose domains. In addition, IANA provides root zone trust anchors for DNSSEC [42].

3.2 Trusted Third Parties

Another type of Internet resources that is fundamental for today's Internet is given with cryptographic certificates. Typically, cryptography serves to establish secure communication channels over the Internet, thereby ensuring confidentiality and integrity for the information exchanged between two parties. In addition, trusted third parties can be employed to ensure authenticity by confirming the communicating parties' identities. Decentralized trust concepts like the web of trust in OpenPGP [43] mostly focus on the secure exchange of sensitive documents and data, most notably emails, within a circle of acquaintances. To this end, a trusted third party for two individuals is given by a mutual friend who attests authenticity. A more prevalent approach to authenticate arbitrary communication partners is based on hierarchical trust. A central authority with a priori trust confirms the identity of intermediaries, who in turn attest authenticity for clients. With such an infrastructure, a vast majority of network protocols can be secured.

3.2.1 Public Key Infrastructure

In practice, most facilities to provide authentication for secure communication channels are based on asymmetric cryptography. These so-called Public Key Infrastructures (PKIs) provide technical means and procedures to manage cryptographic certificates. The most prominent representative is the X.509 PKI [44], which is the de facto standard for securing popular Internet services like the world wide web. In order to establish a secure channel, asymmetric encryption serves to negotiate a symmetric session key between two parties, which is then used to encrypt the subsequent communication. To this end, the initiator of the communication obtains the receiver's public key to encrypt a randomly generated symmetric session key. With his corresponding private key, the receiver is able to decrypt the session key and the encrypted channel can be established.

To ensure authenticity, the ownership of public keys needs to be verifiable. In more specific terms, public keys need to be cryptographically signed by trusted third parties. The resulting documents are called trusted certificates, which contain a so-called *subject* that uniquely identifies a client, his public key, and an appropriate signature added by the *issuer* of the certificate. In order to validate the signature, the issuer's public key obtained from his own certificate is to be used. Note that trust in the latter needs to be established as well, in general with the help of another authenticating body further up in the trust hierarchy. Hence, trusted certificates in fact constitute a chain of certificates. The final link in such a

chain is a self-signed certificate, also called a *root certificate*, owned by a central authority, which requires implicit and ultimate trust.

In client-server communication, authentication is often applied unilaterally to confirm the server's identity only. The reason for that is that trusted certificates are bound to unique identifiers, e.g. to domain names of web servers [45]. Such immutable identifiers are often not available for dynamically connecting clients, let alone the technical knowledge to obtain and use a corresponding certificate.

3.2.2 Certificate Authorities

In the context of PKIs, entities that issue certificates are called Certificate Authorities (CAs). The process of a CA issuing a trusted certificate naturally requires proper authentication of the recipient. The most basic form of authentication is the demonstration of control over a domain name for which a certificate is to be issued. The ability to receive emails for such a domain is considered a proof of ownership. A more sophisticated authentication scheme called Extended Validation further requires the confirmation of individuals' identities and legally binding documents to be signed. Beyond that, CAs generally provide means to revoke certificates, e.g. in case of theft or loss. For this purpose, Certificate Revocation Lists (CRLs) [44] can be queried from dedicated servers, if specified in a CA's certificate. Nowadays, the revocation status of certificates can also be obtained with queries to an Online Certificate Status Protocol (OCSP) [46] server, either by the user himself or by the certificate holder on behalf [47].

The trustworthiness of a CA can always be established by explicitly trusting its public key, and its self-signed certificate respectively. For smaller CAs, e.g. within the scope of a single company, the process of installing the CA's certificate to all of the company's devices is manageable in some cases. However, if a CA issues certificates for public services like a corporate web site, outsiders have no means to validate the certificate. In this case, a higher-ranking CA in the trust hierarchy needs to be commissioned to sign the subordinate CA's certificate. Multiple intermediate CAs can be involved in a certificate chain, its final link, however, is given by certificates of so-called root CAs. For such root CAs, trust anchors are hard-coded into popular software like web browsers or email clients. Given their implicitly trusted root certificates, users are enabled to fully validate certificate chains and can thus reliably assess the authenticity of web sites, email servers etc. Only few root CAs exist world-wide. The Mozilla CA Certificate Store, for instance, holds 180 root certificates from 66 different CAs at the time of writing.

protocol	default port	STARTTLS	SSL/TLS port
FTP [51]	21	*yes* [52]	990
HTTP [53]	80	*yes* [54]	443
IMAP [55]	143	*yes* [56]	993
IRC [57]	6667	*yes* [58]	6697
LDAP [59]	389	*yes* [60]	636
NNTP [61]	119	*yes* [62]	563
POP3 [63]	110	*yes* [56]	995
SMTP [64]	25	*yes* [65]	465
SUBMISSION [66]	587	*yes* [65]	–
XMPP/CLIENT [67]	5222	*yes* [67]	5223
XMPP/SERVER [67]	5269	*yes* [67]	5270

Table 3.1: STARTTLS and SSL/TLS support in popular network protocols.

3.2.3 Secure Network Protocols

A variety of communication protocols is able to make use of trusted certificates. One of the more prominent examples is the Secure Sockets Layer (SSL) Protocol [48] developed by Netscape Communications since 1994, which meanwhile has been replaced by the Transport Layer Security (TLS) Protocol [49]. SSL/TLS operates on top of a reliable transport protocol like the Transmission Control Protocol (TCP) [50] and is application-neutral. It employs X.509 certificates to provide transparent encryption of an application's otherwise unencrypted content. In addition, SSL/TLS offers optional PKI-based authentication of the communicating parties.

In principle, any connection-oriented service can be readily enhanced with SSL/TLS by switching to an SSL/TLS-enabled transport stream replacement. However, this implies that a different port must be used for the secured protocol variant. In the past, IANA assigned many ports for such encapsulated traffic, e.g. port 80 for the Hypertext Transfer Protocol (HTTP) [53] and port 443 for HTTP over TLS [45]. Beside an increased consumption rate for the limited number of assignable ports, this approach introduces a variety of limitations with respect to security policies. The eligible policy to *use TLS when available*, for instance, is cumbersome to implement. To rectify the situation, several protocols were updated around the millenium change to provide so-called STARTTLS functionality. With this approach, an initially unencrypted connection can be upgraded to a secure SSL/TLS connection. Such protocol extensions enable opportunistic use of SSL/TLS, e.g. mail relays that

start to exchange mails over encrypted transports whenever possible. Table 3.1 lists widely deployed network protocols and their support for SSL/TLS and STARTTLS.

Other encryption protocols that employ trusted certificates exist. The Datagram Transport Layer Security (DTLS) Protocol [68] facilitates SSL/TLS functionality for connectionless protocols. IPsec [69], a protocol suite that adds cryptographic protection to the IP layer, supports the use of X.509 certificates for authentication within the Internet Key Exchange (IKE) Protocol [70]. The widely used Secure Shell (SSH) Transport Layer Protocol [71] features various public key-based authentication schemes. Another example for the use of X.509 certificates is given by the Secure/Multipurpose Internet Mail Extensions (S/MIME) [72], which provide authentication and encryption features for email correspondence.

3.3 Public Databases

The management of Internet resources naturally involves databases that hold operational data and administrative information about corresponding resource holders. Since the Internet is generally considered a public good, such databases are publicly accessible to a large extent. In fact, some of the most vital parts of the Internet, like DNS, critically depend on this circumstance. Hence, gathering supplemental information about Internet resources, possibly even in its entirety, is feasible in general. As a consequence, operational and scientific analyses conducted on technical aspects of the Internet can greatly benefit in this respect.

3.3.1 WHOIS System

The first directory service for the Internet, named WHOIS [73], dates back to the early 1980s. A simple query-response protocol, WHOIS provided little more than the name and contact details of persons who were *capable of passing traffic* [74] across the network at that time. Since then, little has changed for the WHOIS system though. Its intended purpose is still to provide informational services for and about Internet participants, albeit this notion does not include end users anymore.

In general, WHOIS queries yield a set of text records in a human-readable representation. Beside a resource holder's contact details, this may include further administrative information about the state of resources like registration dates or intended use. In addition, WHOIS records often contain operational data such as configuration specifics or applicable

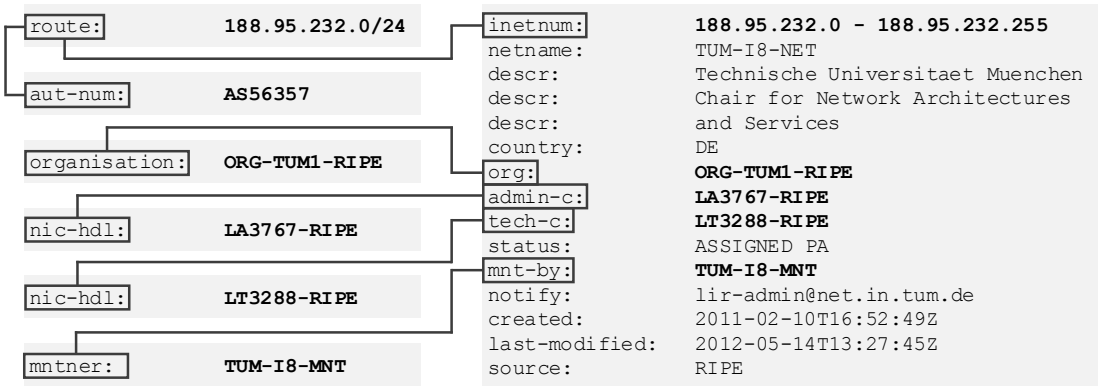


Figure 3.1: WHOIS query for an IP block.

policies defined by the resource holders. In the past, updates to these data sets were applied manually upon request. Nowadays, automated processes are available to a certain extent.

The WHOIS system evolved into a global federation of loosely connected servers hosted by RIRs and NICs, each providing access to their individual resource database. The special-purpose server `whois.iana.org`, operated by IANA, can be used to determine the responsible WHOIS server for any given resource object. Note that in the case of NICs, these servers may in turn further delegate responsibility to subordinate servers. In common opinion, any Internet user is entitled to freely use the WHOIS system. Meanwhile, ICANN has initiated a program to redesign WHOIS from scratch due to severe security concerns particularly with respect to indifferent disclosure of personally identifiable information.

Regional Internet Registries

RIRs provide manifold information about assigned Internet resources and their users. The RIPE database, for instance, holds nearly 9.5 million data objects comprising about 4 million IP blocks of varying size, 29,000 AS numbers, and more than 90,000 organisations. According to RIPE, the monthly query rate of its database exceeds 17,000 requests on average. While all RIRs provide a convenient web interface to query their database, WHOIS servers are still operated to provide traditional command-line based access as well.

Naturally, the aforementioned resource objects are connected with each other. Figure 3.1 shows the result of an exemplary WHOIS query for an IP block in the RIPE database. The retrieved object is shown on the right hand side, unique identifiers that link to other objects are highlighted. The data record includes an organization associated to the IP block as well as administrative and technical contacts. In addition, information about the entity

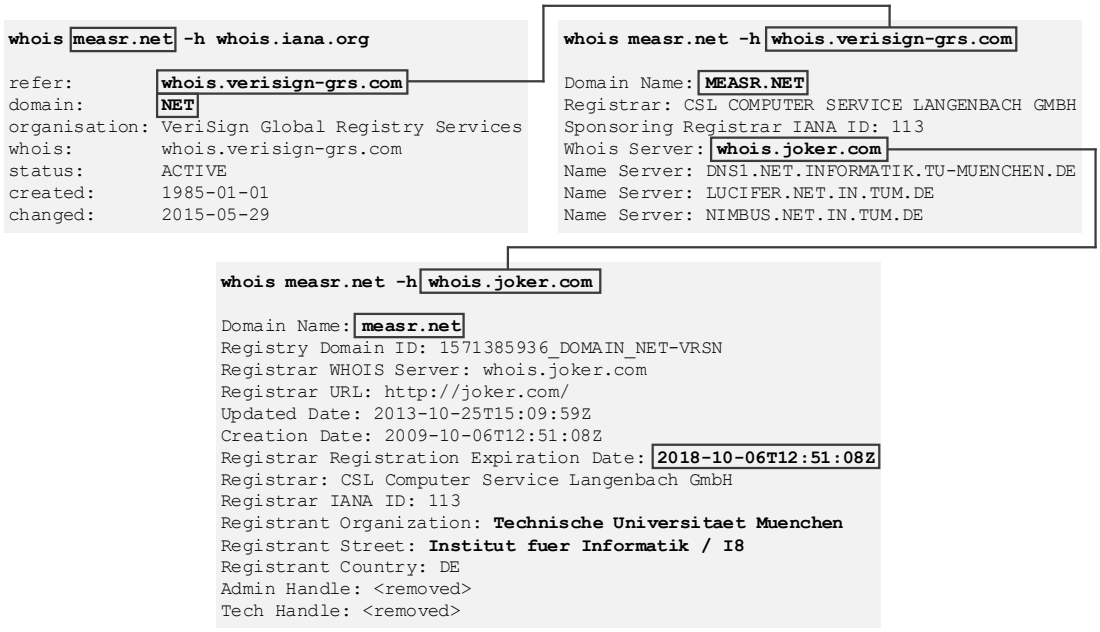


Figure 3.2: WHOIS query for a domain name.

authorized to maintain the object is shown. On the left hand side, related objects linked to the IP block are indicated. Further details for these objects can be obtained by subsequent WHOIS queries. In essence, this data forms a connected graph representing the utilization of resources from both an administrative and an operational point of view. Updates to RIR database objects can be applied via web or email interfaces. Authentication is implemented based on passwords or PGP keys as specified in according maintainer objects. Some RIRs also facilitate single-sign-on systems to consolidate access for various services.

Note that the RIR databases are also embedded in a larger ecosystem formed by so-called Internet Routing Registries (IRRs) [75]. Beside mirroring the RIR data sets, these registries provide means to document routing policies and further operational details. For all five RIRs and the majority of IRRs, bulk WHOIS data is available for download, albeit some registries require prior registration to access it. In most cases, however, personally identifiable information is removed from these data sets to a certain extent.

Network Information Centers

NICs operate WHOIS servers to allow querying for domain name information. The available data sets greatly vary between NICs, but at least include details about the NIC member involved in a registration process, i.e. the *registrar*, and the client who registered a

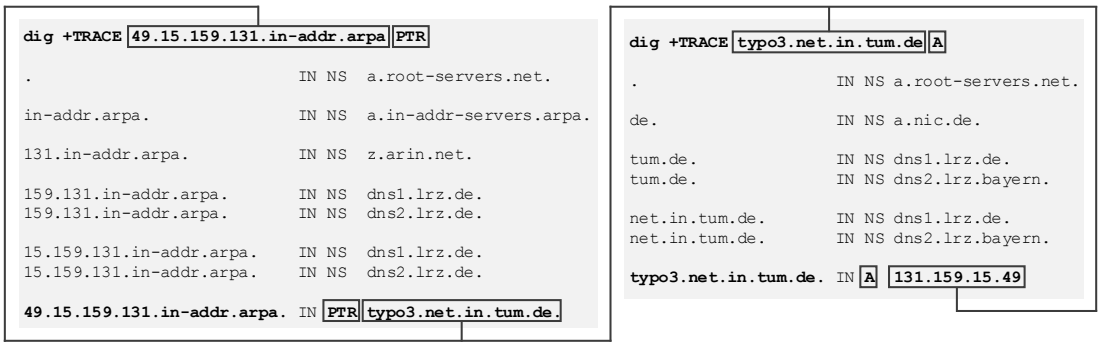


Figure 3.3: Implementation of a FCrDNS check.

domain name, also called the *registrant*. In addition, name servers that handle DNS queries for a domain name, either operated by the registrar or the registrant himself, are specified. Most NICs further provide administrative and technical contact details as well as registration and expiry dates.

The manual pages for the standard Linux `whois` implementation state that this client *tries to guess the right server*. This process is illustrated in Figure 3.2, with an exemplary WHOIS query for the domain `measr.net`. Note that, although this process actually involves three WHOIS queries, the user is usually presented with the final result only. First, the IANA server is queried in order to determine the responsible server for the `.net` zone, which is operated by the *VeriSign Global Registration Service*. A subsequent WHOIS query to the VeriSign server yields that `whois.joker.com` has the necessary information for the domain name in question. This server ultimately provides the full set of details, including rich information about the registrar and registrant, of which some has been removed in Figure 3.2 for the sake of clarity. Note that the expiration date is of particular relevance for the further course of this thesis. It is interesting to see that all three queries yield data records in differing yet human-readable formats. Thus, it is often cumbersome to automate the processing of WHOIS records, which explains the obscure notice in the aforementioned manual pages.

3.3.2 Domain Name System

NIC-operated WHOIS servers do not provide operational information beside a non-authoritative list of name servers for registered domain names. Instead, this information is stored in the Domain Name System, which is a highly distributed public database in itself. Each domain name requires authoritative name servers to answer queries for it. In the case of Figure 3.2, three redundant name servers serve queries for the so-called *forward* mapping

of `measr.net` to an IP address. The *reverse* process, i.e. to map an IP address to a domain name, is supported via the special-purpose zone `in-addr.arpa`, which is operated by IANA. Both types of address resolution are illustrated in Figure 3.3.

Note that according to common opinion [76], forward and reverse mappings should be aligned, i.e. the name obtained for an IP address via a reverse lookup should in turn yield this IP address in a forward lookup, which is the case in Figure 3.3. Otherwise, an operational or configuration error is to be assumed. In contrast, the converse requirement, i.e. that a domain name should match the reverse name of its forward-resolved IP address, would be misguided. The reason for that lies in multiple names that address a single server. A company name registered under several top level domains, for instance, may point to the same web server. In contrast, server IP addresses are typically configured with a single reverse name, and thus do not match all—or any—of the publicly advertised names.

A variety of service implementations, e.g. for mail, SSH, FTP and IRC, perform so-called Forward-confirmed reverse DNS (FCrDNS) checks in accordance to the aforementioned observation. The intended purpose is to confirm ownership of both an IP address, i.e. control over its authoritative reverse DNS server, and a domain name that resolves to that address. Hence, such checks are carried out by doing a reverse lookup on a client's IP address to obtain a list of domain names, also called PTR records, and a subsequent forward lookup on each of these names. If the client's IP address can be mapped to one of the resulting IP addresses, the check is passed. Failures are often reported to system administrators[1] or users, for instance via mail headers indicating possible spam [77]. Some firewalls might even block connections. Although the usefulness of such checks with respect to improving security has been critized [78] for various reasons, FCrDNS is still widely deployed especially in the field of spam detection and mitigation.

3.3.3 X.509 PKI

Arguably not a real database in the traditional sense, each server that provides X.509-based authentication holds a trusted certificate, which can be obtained by an arbitrary client. Hence, it is generally possible to carry out comprehensive scans in order to generate an exhaustive database. To this end, SSL/TLS-capable hosts are characterized by open ports according to Table 3.1. By performing a regular SSL/TLS handshake with any such host, its X.509 certificate can be obtained.

[1] e.g. in SSH logs like »`sshd: reverse mapping checking failed - POSSIBLE BREAK-IN ATTEMPT!`«

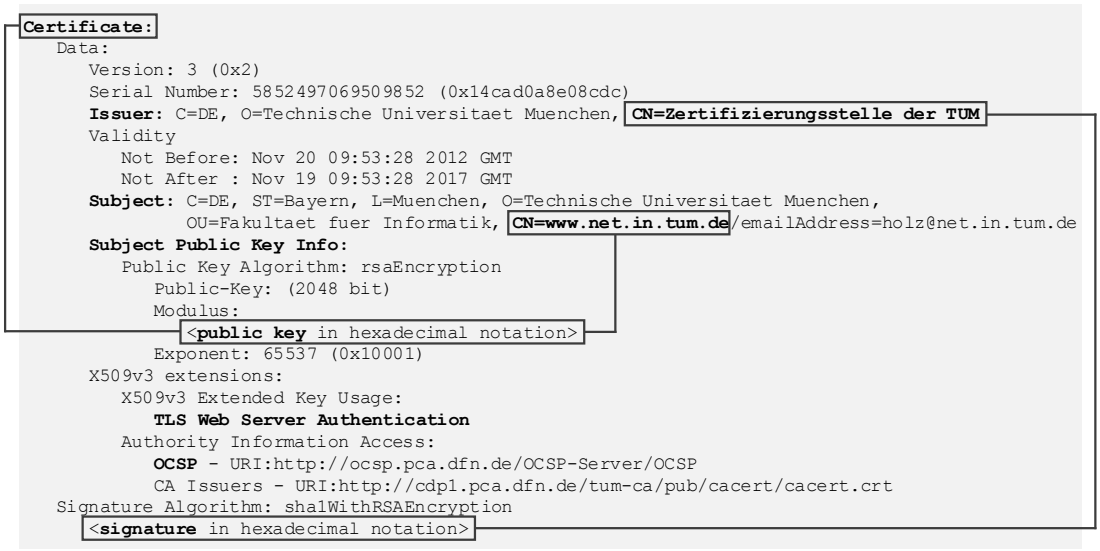


Figure 3.4: Structure of an X.509 certificate.

Figure 3.4 provides an example for a trusted certificate presented by a web server. First, the highlighted fields show the relationship between the certificate, the subject, and the corresponding public key. Note that the *Common Name (CN)* of the subject is authenticated in general, in this case a domain name addressing the web server. Second, a signature that attests the validity of the certificate and thus the validitiy of the public key for the domain name is included. The certificate is issued by the *Zertifizierungsstelle der TUM*, i.e. the certificate authority of the *Technische Universität München (TUM)*.

In the actual SSL/TLS handshake, a complete certificate chain is presented to the client, which is not shown in Figure 3.4 for the sake of brevity. In practice, when connecting to the web server as discussed above, the user receives two additional certificates as part of such a chain. The first is a trusted certificate for the TUM CA issued by the *DFN-Verein*, i.e. the association for a German research network. Naturally, the second certificate received is held by the DFN-Verein, which is in turn signed by *Deutsche Telekom*. The latter is implicitly trusted as a root certificate is shipped with the certificate store of common browsers. Given this certificate chain, a client connecting to `www.net.in.tum.de` is thus enabled to reliably validate the identity of the server.

4 CHAPTER FOUR
The Border Gateway Protocol

NOTE *This chapter does not contain prior publication.*

SUMMARY The Border Gateway Protocol (BGP) is the omnipresent interdomain routing protocol. Its purpose is to exchange global routing information between border routers of autonomous systems (AS). In spite of being a vital part of today's Internet, BGP still lacks adequate security mechanisms. As a consequence, BGP is prone to being attacked. Extensions to BGP have been proposed to provide cryptographic protection, which have not been deployed widely due to increased complexity and resource consumption. In the recent past, efforts to secure BGP with a light-weight public key infrastructure showed remarkable progress. Even so, a comprehensive deployment of a secure interdomain routing protocol is still out of reach for many years.

Section 4.1 contains an appraisal of the historic origins of BGP. An integral part of the protocol, the route update process and its implications on operational practice are described in Section 4.2. Inherent security flaws are exposed in Section 4.3, followed by a discussion of best practices and protocol extensions to improve security. In Section 4.4, BGPsec, the latest approach to secure BGP, is presented.

4.1 Historic Origins

The Internet as of today evolved from ARPANET, a small research network, which was initially funded by the U.S. Department of Defense. It introduced a new concept of packet switching and was the first network to implement the TCP/IP protocol suite. Within three decades, the ARPANET grew into a world-wide network of networks, thereby providing connection to millions of hosts. Today's shortcomings of BGP with respect to security are the product of its pivotal design goal to enable connection for everyone.

4.1.1 Exterior Gateway Protocol

The ARPANET was first brought online in 1969 by connecting four so-called Interface Message Processors (IMP) at universities located on the U.S. east coast. Within ten years, more than 100 IMPs were added to this network, and the need for standardized connection and routing protocols arose. In 1978, the concept of Gateway Routing was introduced [79], followed by a detailed specification of a gateway implementation one year later [80]. In conjunction with a comprehensive specification of the DARPA Internet Gateway [81] in 1983, a Gateway-to-Gateway Protocol (GGP) was defined. GGP implemented a shortest-path routing algorithm based on a distance metric measured in gateway hops. More importantly, the gateway design also included an Exterior Gateway Protocol (EGP) [23] to permit the interconnection of arbitrary gateways. The exchange of gloabl routing information was designed to allow any participant to *perceive all of the networks and gateways as part of one total Internet system*, without imposing explicit requirements on routing protocols or gateway hardware. A standardized format for network reachability information ensured that any participant was able to globally advertise his networks. With more and more gateways being added, the EGP was further extended by the concept of autonomous systems [82] and interoperability with Interior Gateway Protocols (IGP).

4.1.2 BGP-4

Based on five years of operational experience [83, 84] with the EGP, and with commercial service providers entering the Internet market, a new Border Gateway Protocol [85] was drafted, which finally evolved into a path-vector-protocol [86, 87]. In 1994, a fourth version of the protocol was specified [27, 88]. It provides a new concept of classless interdomain routing, i.e. IP prefix based routing, together with route aggregation features and support

for routing policies. The latest revision of BGP-4 [89] stems from 2006 and is the de-facto standard in today's interdomain routing. Due to the depletion of the 16-bit number pool for globally unique Autonomous System Numbers (ASNs), BGP was extended to support 32-bit ASNs [90] in 2012.

4.2 Route Update Process

BGP routers exchange Network Layer Reachability Information (NLRI), i.e. reachability of IP prefixes, via update messages. These messages include various path attributes, most importantly the `AS_PATH` that an update message has traversed. All available routing information is stored in the Routing Information Base (RIB) of routers. Upon reception of new update messages, the included routing information is validated. If no IP-level route to the `NEXT_HOP` IP address is available, or if the `AS_PATH` attribute already contains the receiver's ASN, which indicates a routing loop, the routing information is considered invalid. Otherwise, a decision process is triggered to determine if the received route is in fact the best route available to the advertised IP prefix. This decision process – also known as the Best Path Selection Algorithm – is influenced by a variety of parameters including local policies. In general, routes with shorter AS paths are preferred. If a new best route is selected and no local policies contradict, it is installed into the Forwarding Information Base (FIB) and advertised to neighboring routers. In this case, the `AS_PATH` attribute is prepended by the sender's ASN. With respect to IP packet forwarding, the selection of a FIB entry is generally based on a longest prefix match such that longer prefixes, i.e. networks of smaller size, are preferred.

4.3 Security Considerations

Since the early 1990s, the Internet experienced a significant shift in nature. In the beginning, it was designed to primarily connect networks among trusted parties like universities or governmental institutions. With its unexpectedly rapid growth and increasing economic impact, original assumptions on trust became obsolete. Nowadays, many threats to routing protocols exist [91, 92, 93, 94, 95, 96, 97]. The focus of this thesis lies on threats arising from attackers who inject falsified routing information into the global routing system. Attacks targeting individual BGP routers in order to manipulate existing BGP sessions are beyond the scope of this thesis.

In Section 3.1 we discussed how Internet resources like IP blocks and ASNs are administered by RIRs and assigned to LIRs for operational use. In BGP, however, there is no corresponding concept to validate or enforce resource ownership. Every BGP participant may advertise reachability of any network, in particular with arbitrary path attributes. Without any mechanism in BGP to semantically validate route advertisements, upstream providers are generally compelled to trust a customer's update messages. Hence, an attacker may exploit his upstream providers to propagate falsified routing information. In particular, this includes the potential to originate IP prefixes without authorization and to manipulate attributes like the `AS_PATH` of a BGP message.

4.3.1 IRR-Based Filtering

Some ISPs filter customer route updates based on external knowledge. IRR databases can be queried to create filter lists that incorporate RIR ownership information. A route update is often considered valid if a corresponding `route` database object exists, which binds the originating AS to the advertised IP prefix. To generalize the documentation of routing policies, IRR databases support the Routing Policy Specification Language (RPSL) [98, 99]. With RPSL, ISPs are enabled to document their routing policies in greater detail. Furthermore, low-level router configurations can be generated from RPSL statements to ease the process of creating filter lists. However, since routing documentation is manually maintained by resource holders, the information in IRR databases is prone to being incomplete or outdated. This fact hinders the implementation of strict filtering policies. In addition, such filter lists can grow quickly for a larger number of customers, and may thus consume a significant amount of resources on BGP routers. Moreover, upstream providers form a multi-tier network, in which higher-tier ISPs provide connection for lower-tier ISPs. While a provider might be able to assemble filter lists for his direct customers, filtering for his customers' customers is not feasible since this information is hard to collect and might change quickly. As a consequence, RPSL-based validation of AS paths is impractical. The same applies to so-called peering connections, in which ISPs exchange customer routes on a horizontal agreement. Other practical problems result from ISPs operating in multiple RIR regions, such that none of the RIR databases is able to sustain exhaustive documentation. Despite the fact that IRRs aim to improve this situation by mirroring and consolidating RIR databases, conflicting information may exist nonetheless. More importantly, the process of updating IRR databases is open to anyone by design.

4.3.2 Proposed Protocol Extensions

Much effort has been put into the development of security extensions to BGP. In general, such extensions have to 1) provide means for authentication of AS numbers, 2) support ownership validation of IP prefixes for authenticated ASNs, and 3) allow for integrity checks on AS paths. With the first requirement, impersonation of ASes is addressed. The second requirement ensures that ASes cannot originate prefixes of other ASes. The manipulation of AS paths is eliminated with the third requirement.

An early approach to extend BGP is Secure BGP (S-BGP) [100]. It proposes a hierarchical public key infrastructure to provide different types of trusted certificates. These certificates enable ownership validation for IP blocks and AS numbers and authentication between BGP routers. The approach also provides certificates to verify that a BGP router is authorized to represent an AS. With this set of certificates, two types of so-called attestations can be signed. Address attestations are used to legitimate ASes to originate a given prefix. Route attestations guarantee that every AS along a given AS path has been authorized by the preceding AS to advertise the route, which is called forward attestation. These attestions are to be included in BGP messages via additional path attributes. Any BGP router who receives and supports such extended messages is able to cryptographically verify the authenticity of its neighboring routers and corresponding update messages. This includes the integrity of the whole AS path and the authority of advertising ASes to originate an IP prefix.

S-BGP suggests a root CA operated by IANA and a hierarchy of CAs given by RIRs and LIRs. S-BGP thus introduces significant computational overhead since BGP routers need to validate a full certificate chain for every extended BGP message. To cope with this problem, the Secure Path Vector [101] protocol reduces computational complexity by utilizing more efficient cryptographic primitives. Secure Origin BGP (soBGP) [102] introduces a web of trust model that does not require signatures for individual update messages. Similar to PGP, trusted ASes are entitled to authenticate other ASes, and to sign their certificates respectively. Given a mutually trusted third-party AS, the authenticity of neighboring routers can be verified. Additional authorization certificates are employed to prove legitimization to originate IP prefixes. soBGP thereby introduces a new BGP security message to exchange certificates. In contrast to S-BGP, there is no need to sign individual update messages since authentication is based on stored certificates. Thus, soBGP is more flexible in terms of incremental deployment and introduces less computational overhead. However, it provides a considerably weaker check for AS path integrity and is arguably less secure than S-BGP due to its distributed trust model.

The concept of Pretty Secure BGP (psBGP) [103] aims at combining the strengths of both approaches to provide a balance between security and practicality. psBGP proposes a centralized PKI to authenticate AS numbers and a distributed endorsement scheme for IP prefix ownership validation. To this end, ASes create so-called prefix assertion lists for themselves and for neighboring ASes. AS path validation is handled in a similiar way as with S-BGP. Due to an increased complexity as compared to S-BGP and soBGP and weaker security compared to a fully hierarchical solution, psBGP did not achieve broad acceptance. Trusted BGP (TBGP) [104] proposes a light-weight attestation service running on all BGP routers to build transitive trust relationships without the need for architectural extensions like a PKI or a web of trust. TBGP postulates authenticity and integrity of routing information, if its scheme is enforced on all involved routers along a routing path. An oppositional approach is the deployment of an Interdomain Routing Validator (IRV) [105] as a distributed query system to implement missing features in BGP. The use of BGP community attributes to enrich BGP updates with a list of legitimate origin ASes is presented in [106]. With [107], a dynamic route selection scheme based on insights into general routing dynamics is proposed. Measurement-based approaches to identify anomalous BGP messages are suggested in [108, 109]. A hybrid approach that introduces BGP signatures and the monitoring of TCP flows to detect reachability problems is proposed in [110]. The Neighborhood Watch Program [111] utilizes careful inference rules based on IRR databases to assess the validity of origin ASes in BGP messages. PGBGP [112] and QBGP [113] propose to slow down the propagation of suspicious routes to limit the impact of illicit BGP messages. Finally, the use of the existing DNS infrastructure to validate origins with the help of new BGP resource records signed by DNSSEC is proposed in [114].

4.4 Securing BGP with BGPsec

BGPsec [115] is the latest approach to secure BGP. It is developed by the IETF Secure Inter-Domain Routing (SIDR) working group. An integral part is the Resource Public Key Infrastructure (RPKI), which provides so-called resource certificates. These certificates attest the right-of-use for IP resources, which is granted by the issuer of a certificate. Corresponding CAs that can issue such certificates comprise IANA as well as RIRs and LIRs or any other entity that is capable to delegate resources. The RPKI is designed as a distributed repository of resource certificates. In practice, however, it is essential for a participating party to built up a local cache of the entire RPKI information space. Signed manifests are used to ease this synchronization process.

To secure route origins, so-called Route Origin Authorizations (ROA) signed by resource certificates are used. A ROA represents an AS and a list of IP prefixes it is allowed to originate. ROAs can be validated in the RPKI and multiple ROAs per prefix are allowed. To support incremental deployment, the validation scheme allows for *valid*, *invalid* and *unknown* results. It is recommended to prefer valid routes over unknown and unknown routes over invalid. Prefixes that are currently not in use should be bound to the special AS number 0, such that the validation outcome will always be invalid. Note that RPKI-based origin validation is designed without imposing changes to BGP itself. To secure route propagation, BGPsec further proposes router certificates stored in the RPKI and a new attribute for BGP update messages. This attribute holds a signature of the outgoing BGP messages, which covers the local and neighboring AS as well as the signature of the original incoming message. Such an interlocking signature scheme together with forward attestation of the receiving AS provides an effective protection against the manipulation of AS paths.

Today, the RPKI is well advanced from an implementational point of view and already deployed by RIRs in production environments. ROAs can be created by LIRs with little effort at RIR-provided management portals, and tools for validation readily exist [116, 117, 118]. Although the entry threshold for RPKI, i.e. to create ROAs or to deploy origin validation, is relatively low compared to previous approaches, still no ROAs exist for 98% of all BGP updates [119]. More recent figures [120] confirm this finding. A major reason for slow adoption is missing economic incentives. The deployment of origin validation by any particular ISP does not improve security for that ISP but rather improves security for all other ISPs. Moreover, with little deployment, there is in turn no incentive to create ROAs in the first place. A more detailed discussion on the adoptability of security extensions to BGP is given in [121]. An oppositional study [122] provides implicit arguments against RPKI due to prevalent incentives for dishonest path announcements. Constructive ways to drive the deployment based on a set of early adoptors are discussed in [123]. A mathematical examination on the security of BGP extensions is carried out in [124]. Further, the authors of [125] anticipate new challenges for BGP security after partial deployment. More fundamental concerns have been expressed by operators [126] that a fully deployed RPKI might enable nation-state adversaries to revoke certificates and thus to disconnect arbitrary autonomous systems permanently. Beyond that, both the specification and implementation of path validation in BGPsec are still at an early stage [127].

5 CHAPTER FIVE
A Comprehensive Attacker Model

NOTE *This chapter contains prior publication.*

Submitted to **IEEE/ACM Transactions on Networking (TON), 2016.**
Sections 5.1 and 5.2.2 are based on previous work [9]. The formal definitions in Section 5.1 are extended to allow for the discussion of additional types of attacks.

Published in **IEEE Journal on Selected Areas in Communications–Special Issue on Measuring and Troubleshooting the Internet (JSAC-SI-MT), 2016.**
Sections 5.1, 5.2.1 and 5.3 are based on previous work [8]. The motivation behind attacks in Section 5.3 is discussed in greater detail.

SUMMARY So far, the discussion of routing attacks in BGP relies on informal wording and imprecise formulations. In Section 5.1, the author improves on this situation by introducing the novel concept of *finite route languages*. Based on formal languages, this rigorous model provides a complete description of the observable routing system and can be used to study various aspects of Internet routing. It is applied to formalize a generic routing attack and to assess its characteristics with respect to requirements and impact. A comprehensive classification of hijacking attacks is presented in Section 5.2. It includes (sub)prefix and AS hijacking, man-in-the-middle, and hidden takeover attacks, which are rigorously formalized and studied in great detail. In Section 5.3, the author further assesses the motivation behind different types of attacks and thoroughly discusses eligible attack tactics.

5.1 A Formal Model for Internet Routing

A formal language is a set of strings of symbols constrained by specific rules. The global Internet routing system can be represented as all active BGP routes, i.e. a set of AS paths from all vantage points towards all advertised IP prefixes. By adopting the immediate analogy, we define the *Finite Route Language (FRL)*. In this model, a routing attack is then defined by an attacker extending the set of active routes by forged routes.

5.1.1 Finite Route Languages

Let Σ_{AS} be the set of all ASes, Π the set of all IP addresses, $p \subset \Pi$ an IP prefix, and $p' \subset p$ a more specific prefix of p. Let further be $(w, p) \in \Sigma_{AS}^* \times \mathscr{P}(\Pi)$ a route with an AS path $w \in \Sigma_{AS}^*$, i.e. an arbitrary concatenation of ASes, to a prefix $p \subset \Pi$, in the following denoted $r = wp$. Then, we define $\mathscr{L} \subset \Sigma_{AS}^* \times \mathscr{P}(\Pi)$ as the set of *active* routes to all advertised prefixes in the global routing system, i.e. the set of all observable routes. $\mathscr{L}(p) \subset \mathscr{L}$ denotes the subset of routes to a given prefix $p \subset \Pi$, such that

$$\mathscr{L}(p) \;=\; \{wuop \in \mathscr{L} \mid w \in \Sigma_{AS}^*\,;\; u \in \Sigma_{AS}\,;\; o \in \Sigma_{AS}\}$$

with w being an AS subpath and u the upstream AS of the origin AS o. For a given route r and a subprefix $p' \subset p$, we postulate $r \in \mathscr{L}(p) \Rightarrow r \in \mathscr{L}(p')$ as a corollary, since routes to less specific prefixes also cover more specific prefixes. Note that the converse is false. Further, $\mathscr{L}_P \subset \mathscr{L}$ denotes the set of all observable routes from a set of observation points $P \subset \Sigma_{AS}$.

Π_o denotes the set of IP addresses advertised by an AS $o \in \Sigma_{AS}$, i.e. the union of its advertised prefixes. Then, the set of origin ASes for a prefix $p \subset \Pi$ is given by

$$O(p) \;=\; \{o \in \Sigma_{AS} \mid p \subset \Pi_o\}\,.$$

Consistently, the set of origin ASes $O(p')$ for a subprefix $p' \subset p$ also comprises the origin ASes for less specific prefixes such that

$$O(p') \;=\; \{o \in \Sigma_{AS} \mid p' \subset \Pi_o\} \;=\; \bigcup_{p \supseteq p'} O(p)\,,$$

since all of these ASes effectively originate routes to the particular network p'. Note again that the converse is false. The set of upstream AS neighbors for an AS $o \in \Sigma_{AS}$ is given by

$$U(o) \;=\; \{u \in \Sigma_{AS} \mid wuop \in \mathscr{L} \text{ such that } w \in \Sigma_{AS}^*\,;\; p \subset \Pi_o\}\,.$$

Note that due to best path selection in BGP (see Section 4.2), the number of routes from an observation point $s \in \Sigma_{AS}$ to a particular prefix $p \subset \Pi$ is generally bound by the number of neighboring ASes of s, i.e. $|\mathscr{L}_s(p)| \leq |U(s)|$ holds. In the followiong, we reuse the unary operator $|\,.\,|$ to indicate the number of routes in a set $\mathscr{O} \subseteq \mathscr{L}$, the length of a route $r \in \mathscr{L}$ or a subpath $w \in \Sigma_{AS}^*$, and the number of ASes in a set $S \subseteq \Sigma_{AS}$.

5.1.2 Formalization of Routing Attacks

We define routing attacks as an attacker extending the global set of BGP routes $\mathscr{L}$ by forged routes. Their purpose is to manipulate existing routes in order to re-route traffic flows or to take over a victim's Internet resources. In general, such incidents are called *hijacking attacks*. Typically, these attacks lead to topological changes in the Internet, which can be observed by neutral BGP speakers.

Attacker Model

Our attacker is assumed capable of injecting arbitrary BGP messages into the global routing system, i.e. he operates a BGP router and maintains a BGP session to at least one upstream provider. We assume that the attacker is not hindered by local filters or other validation mechanisms employed by his upstream provider. Instead, the upstream AS indifferently redistributes all update messages to its peers, which thus may propagate throughout the Internet. An observation point shall be in place to monitor the propagation of BGP messages. It is worth mentioning that data packets do not necessarily traverse all ASes in a given path since an attacker may craft BGP messages with a forged AS path. Further, route updates with less attractive paths may not reach a particular observation point due to best path selection in BGP (Section 4.2). Without loss of generality, an omnipresent observation point to observe the set of all active routes $\mathscr{L}$ is assumed for the following definitions.

Generic Routing Attacks

In the following, we denote an attacker's AS $a \in \Sigma_{AS}$ and his victim's AS $v \in \Sigma_{AS}$. Further, a victim's prefix is given by $p_v \subset \Pi_v$. Then, a generic routing attack on p_v is defined by an attacker injecting forged routes $\mathscr{F}_a$ into the routing system, such that the altered set of globally visible routes $\hat{\mathscr{L}}(p'_v)$ is given by

$$\hat{\mathscr{L}}(p'_v) \;=\; \mathscr{L}(p_v) \cup \mathscr{F}_a(p'_v) \;\text{ with }\; p'_v \subseteq p_v \,.$$

Requirements and Impact

In BGP, the impact of a hijacking attack generally depends on a best path selection process. In particular, shorter AS paths are preferred over longer ones, although policy-induced exceptions on a case-by-case basis may exist. With respect to packet forwarding, routes to longer IP prefixes prevail. Assuming the ambition to forge globally accepted routes, an attacker thus succeeds if his routes towards a victim's network are considered best by a vast majority of Internet participants. In practice, an attacker needs to ensure that his bogus routes $\mathscr{F}_a(p'_v)$ are either

1) unrivaled in terms of competitive routes, i.e. $|\mathscr{L}(p'_v)| = 0$,
2) shortest from a global perspective, i.e. $\forall r \in \mathscr{L}(p'_v),\ r_a \in \mathscr{F}_a(p'_v) :\ |r_a| < |r|$, or
3) more specific than all others, i.e. $\forall p''_v \subseteq p'_v \subset p_v :\ \mathscr{L}(p''_v) = \mathscr{L}(p'_v) = \mathscr{L}(p_v)$.

As a consequence, the prospects of identifying an attack naturally depend on the significance of topological changes in $\hat{\mathscr{L}}$, i.e. on abnormal changes to the sets of origin ASes $\hat{O}$ and upstream ASes $\hat{U}$ for a victim's network, and on the pecularities of the forged routes.

5.2 Classification of Attacks

BGP-based attacks aim to inject falsified protocol messages into the global routing system, which may lead to topological changes in the Internet. Depending on the characteristics of such changes, hijacking attacks can be classified into several subtypes with differing tactical value.

Examples

In addition to the formalized model, the topology in Figure 5.1 is used to exemplify different types of attacks. *Mallory* thereby denotes an attacker, and her autonomous system respectively. *Oscar* and *Paul* are *Mallory's* upstream providers who do not validate BGP messages. The observation point *OP* receives route updates that propagate through the Internet, which are herein after simplified to the topology-relevant attributes of BGP messages, namely the IP prefix, also called `NLRI`, and the AS path, refered to as `AS_PATH`. The following expression illustrates such an update observed at *OP:*

$$OP: \qquad AS_*^K \ \leftarrow\ Carol \ \leftarrow\ Alice \ \ll\ P \qquad legitimate$$

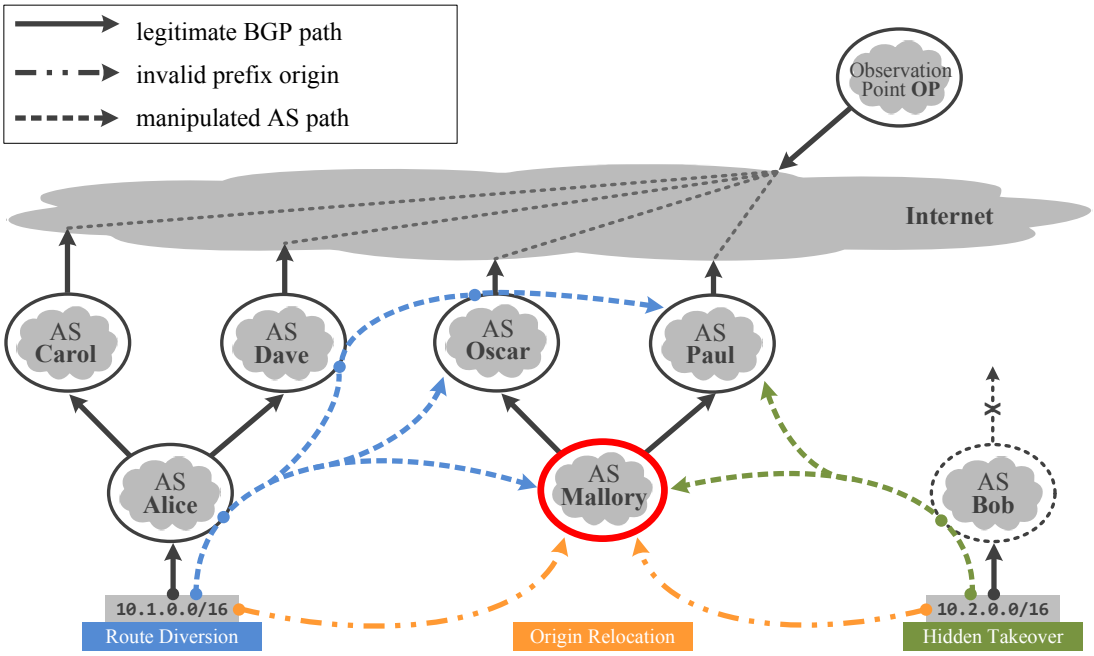


Figure 5.1: Attack vectors for BGP attacks.

A route update for the prefix P originates ($\ll$) at *Alice* and traverses *Carol* and a series of K ASes AS_*^K to reach the observation point *OP*. For the sake of clarity, temporary convergence effects within BGP are ignored. *Alice* and *Bob* serve as victims for different kinds of attacks, *Carol* and *Dave* provide upstream connectivity for *Alice*.

Hijacking Attacks in Practice

Given a standard router with an already established point-to-point IP connection to the router of an upstream provider, it is surprisingly easy to participate in BGP and to originate an IP prefix. Consider the following BGP configuration of *Alice's* router named `rtr`, which actually represents a minimal working example:

```
rtr(config)#  router bgp alice
rtr(config-router)#  neighbor X.X.X.X remote-as carol
rtr(config-router)#  network 10.1.0.0/16
```

In this example, *Alice* opens a BGP session to her neighbor router `X.X.X.X`, which is operated by *Carol*, and advertises direct reachability of `10.1.0.0/16`. This information will be redistributed by *Carol* to her neighbors and may eventually propagate to all BGP routers connected to the Internet.

5.2.1 Origin Relocation Attacks

It is easy for an attacker to originate arbitrary IP prefixes, irrespectively of whether legitimate routes to these IP prefixes already exist. Depending on the specifics of the attacker's approach, the original routes will be partially or entirely overridden in the global routing system. Hence, this type of attack results in an *origin relocation* of a victim's network.

Prefix Hijacking

The most basic form of origin relocation attacks is *prefix hijacking*. An attacker thereby originates a victim's prefix at his own AS, in principle in the same way as illustrated above. The resulting forged routes compete with the victim's concurrent announcements. The following definition formulates this attack scenario:

$$\begin{aligned} \hat{\mathscr{L}}(p_v) \;=\; & \{wuvp_v \mid w \in \Sigma_{AS}^*\,;\; u \in U(v)\} \cup && \textit{legitimate} \\ & \{wuap_v \mid w \in \Sigma_{AS}^*\,;\; u \in U(a)\} && \textbf{forged} \\ \text{with}\quad \hat{O}(p_v) \;=\; & O(p_v) \cup \{a\} \\ \text{and}\quad \hat{U}(v) \;=\; & U(v) \end{aligned} \tag{5.1}$$

The set of origin ASes $\hat{O}(p_v)$ for the prefix p_v now comprises two ASes, while the set of the victim's upstream ASes $\hat{U}(v)$ remains unchanged. In literature, a situation with $|O(p)| > 1$ is often called a *multi-origin AS (MOAS)*. Given the exemplary topology in Figure 5.1, *Mallory* may craft a BGP update message as listed below. At the same time, legitimate paths advertised by *Alice* are present in the global routing table.

OP:	AS_*^K	$\leftarrow$	*Carol*	$\leftarrow$	*Alice*	$\ll$	`10.1.0.0/16`	*legitimate*
OP:	AS_*^L	$\leftarrow$	*Dave*	$\leftarrow$	*Alice*	$\ll$	`10.1.0.0/16`	*legitimate*
OP:	AS_*^M	$\leftarrow$	*Oscar*	$\leftarrow$	*Mallory*	$\ll$	`10.1.0.0/16`	**forged**

Example 5.1: Prefix Hijacking.

As shorter AS paths are generally preferred over longer ones, the attack is likely to succeed for observation points s where $M < \min(K, L)$ holds, and for

$$\{s \in \Sigma_{AS} \mid \forall\, w_v p_v \in \mathscr{L}_s(p_v)\; \exists\; w_a p_v \in \mathscr{F}_a(p_v) :\; |w_a| < |w_v|\}$$

respectively. However, it is safe to assume that clients that are topologically close to *Carol* or *Dave* still reach the victim *Alice*, since shorter legitimate routes take precedence. The Internet thus decomposes into two disjoint parts: one part that is affected by the forged announcement, in literature often called the *poisoned* part, and one that remains unaffected.

Subprefix Hijacking

To overcome the limited impact inherent to prefix hijacking, an attacker can leverage longest prefix matching in IP routing with so-called *subprefix hijacking*. To this end, the attacker originates a subprefix $p'_v \subset p_v$ at his AS, thereby injecting a new set of routes $\mathscr{F}_a(p'_v)$ into the global routing system as given by:

$$\begin{aligned} \hat{\mathscr{L}}(p'_v) &= \{wuvp_v \mid w \in \Sigma^*_{AS}\,;\, u \in U(v)\} \cup && \textit{legitimate} \\ &\quad\ \{wuap'_v \mid w \in \Sigma^*_{AS}\,;\, u \in U(a)\} && \textbf{forged} \\ \text{with}\quad \hat{O}(p_v) &= O(p_v), \\ \hat{O}(p'_v) &= O(p_v) \cup \{a\} \\ \text{and}\quad \hat{U}(v) &= U(v) \end{aligned} \tag{5.2}$$

Subprefix hijacking attacks generally have global impact, since routes to the more specific prefix $p' \subset p$ dominate. In accordance with prefix hijacking, such an incident with $|O(p)| > 0$ and $|O(p') \setminus O(p)| > 0$ is called *subprefix multi-origin AS (subMOAS)*.

Note that the victim might readily advertise routes to a prefix and a corresponding subprefix concurrently to the attacker's subprefix route, i.e. condition 3) in Section 5.1.2 does not hold since $\mathscr{L}(p_v) \subset \mathscr{L}(p'_v)$. In this case, the event can also be considered a MOAS event. Otherwise, it is called a *strict* subMOAS, and subprefix hijacking respectively. We assume this variant in the following. While virtually all Internet participants are affected by strict subprefix hijacking, it may be tempting to conclude that only part of the victim's network, i.e. subprefixes, can be taken over. As a matter of fact, this is not the case. An attacker can easily craft multiple update messages such that the set of forged routes $\mathscr{F}_a(p_v)$ fully covers the prefix p_v with more specific routes:

$$\mathscr{F}_a(p_v) = \bigcup_{p'_v \subset \Pi} \mathscr{F}_a(p'_v) \text{ such that } p_v = \bigcup p'_v\,.$$

With respect to Figure 5.1, *Mallory* could thus inject the following BGP routes:

OP:	AS^K_*	←	*Carol*	←	*Alice*	≪	`10.1.0.0/16`	*legitimate*
OP:	AS^L_*	←	*Dave*	←	*Alice*	≪	`10.1.0.0/16`	*legitimate*
OP:	AS^M_*	←	*Oscar*	←	*Mallory*	≪	`10.1.0.0/17`	**forged**
OP:	AS^M_*	←	*Oscar*	←	*Mallory*	≪	`10.1.128.0/17`	**forged**

Example 5.2: Subprefix Hijacking.

By advertising a victim's network with BGP updates split up into multiple longer prefixes, as given by `10.1.0.0/17` and `10.1.128.0/17` in the example above, subprefix hijack-

ing can be as effective as regular prefix hijacking, with the advantage of globally preferred routes at the same time. Notwithstanding this possibility, an attacker might be satisfied with hijacking individual subnets of high value only.

5.2.2 Route Diversion Attacks

Origin relocation attacks lead to noticeable changes in the set of origins for a victim's prefix. To further disguise an attack, the AS path, i.e. the `AS_PATH` attribute of forged BGP messsages, can be manipulated as well. Since this attribute is essential for loop detection in BGP, standard BGP routers do not provide means for such alteration, except for prepending a BGP speaker's own AS to the AS path multiple times for traffic shaping purposes. However, it is common practice with route servers [128], and open-source software tools like ExaBGP [129] have been developed to provide this functionality as well, albeit with different use cases in mind.

AS Hijacking

In general, the `AS_PATH` attribute can be set to any subpath $w \in \Sigma_{AS}^*$. It is constructive, however, to keep AS paths short, and, more importantly, to avoid abnormal MOAS incidents. To this end, an attacker could add his victim's AS v to the AS path, which is thus called an *AS hijacking* attack:

$$\begin{aligned} \hat{\mathscr{L}}(p_v) &= \{wuvp_v \mid w \in \Sigma_{AS}^*\,;\, u \in U(v)\} \cup && \textit{legitimate} \\ &\quad\ \{wuavp_v \mid w \in \Sigma_{AS}^*\,;\, u \in U(a)\} && \textbf{forged} \\ \text{with}\quad \hat{O}(p_v) &= O(p_v) \\ \text{and}\quad \hat{U}(v) &= U(v) \cup \{a\} \end{aligned}$$

This approach provides a remarkable benefit: The set of origin ASes $\hat{O}(p_v)$ for the prefix p_v remains unchanged, which effectively disguises the attack. The only noticeable effect is an allegedly new upstream provider for v given by the attacker's AS a. Furthermore, the attacker could manipulate BGP messages as follows to even hide his own AS:

$$\begin{aligned} \hat{\mathscr{L}}(p_v) &= \{wuvp_v \mid w \in \Sigma_{AS}^*\,;\, u \in U(v)\} \cup && \textit{legitimate} \\ &\quad\ \{wuvp_v \mid w \in \Sigma_{AS}^*\,;\, u \in U(a)\} && \textbf{forged} \\ \text{with}\quad \hat{O}(p_v) &= O(p_v) \\ \text{and}\quad \hat{U}(v) &= U(v) \cup U(a) \end{aligned} \tag{5.3}$$

With this approach, the only trace left is the attackers' upstream provider $u \in U(a)$, which seemingly acts as a new upstream provider $u \in \hat{U}(v)$ for the victim. We assume this variant in the remainder of this thesis. The example below (compare to Figure 5.1) illustrates both variants of this compelling approach:

OP:			AS_*^K	$\leftarrow$	*Carol*	$\leftarrow$	*Alice*	$\ll$	`10.1.0.0/16`	*legitimate*
OP:			AS_*^L	$\leftarrow$	*Dave*	$\leftarrow$	*Alice*	$\ll$	`10.1.0.0/16`	*legitimate*
OP:	AS_*^M	$\leftarrow$	*Oscar*	$\leftarrow$	*Mallory*	$\leftarrow$	*Alice*	$\ll$	`10.1.0.0/16`	**forged**
OP:			AS_*^M	$\leftarrow$	*Oscar*	$\leftarrow$	*Alice*	$\ll$	`10.1.0.0/16`	**forged**

Example 5.3: AS Hijacking.

In this scenario, either the attacker *Mallory* or her upstream provider *Oscar* appear as a third upstream provider for the victim *Alice*, which makes it hard to be recognized as an attack as such. Especially for the latter case, no actual offender can be identified in any of the corresponding BGP messages, a fact that greatly hinders mitigation.

AS hijacking per se still has the disadvantage of limited impact, though, since the attacker's routes do not dominate all parts of the Internet. To compensate, an attacker can easily combine his attack with subprefix hijacking to profit from both high impact and little traceability.

Man-in-the-Middle Attacks

A devastating attack based on AS path manipulation aims at stealthily intercepting a victim's traffic. Such an attack succeeds if data packets sent to the victim are re-routed via the attacker's AS, who in turn needs to have a stable backhaul path to the victim to forward the eavesdropped packets. Such attacks are unfeasible to detect from a topological point of view since only new paths over already existing links emerge.

The challenge with this approach is that the attacker is generally affected by his hijacking attacks as well. During the attack, there is no reliable way for him to send IP packets to his victim, since global routes to the victim's network now lead to his own AS. A solution to this problem could be to establish a BGP-agnostic link to the victim, for instance a layer 2 connection or an MPLS tunnel. In most cases, however, this approach is not feasible without physical access to the victim's network. Interestingly, BGP itself can be leveraged to solve this problem. To prevent routing loops, BGP discards update messages that already include a router's own AS in the `AS_PATH` attribute. Hence, the attacker needs to fabricate

a message that includes all ASes on the path from his own AS a to the victim's AS v. As a consequence, all of these ASes discard the attacker's BGP messages, and thus a stable link to the victim's AS remains intact. As a BGP speaker, the attacker can easily identify such a path to p_v in $\mathscr{L}_a(p_v)$. The set of AS paths that represent a suitable backhaul link via an arbitrary upstream AS $u \in U(a)$ thus reads

$$W_v^a(u) \;=\; \{w_v^a \in \Sigma_{AS}^* \mid u w_v^a p_v \in \mathscr{L}_a(p_v)\}\,.$$

Note that in practice, exactly one such path w_v^a exists per upstream AS, i.e. $|W_v^a(u)| = 1$, since each neighbor AS redistributes its prefered routes only. For the attack to succeed, the attacker needs to employ at least two upstream providers. In the following, $t \in U(a)$ serves to establish the backhaul link, while the remaining upstream providers $s \in U(a) \setminus \{t\}$ are utilized to launch a particular (sub)prefix hijacking attack against $p'_v \subseteq p_v$ such that

$$\begin{aligned}
\hat{\mathscr{L}}(p'_v) \;=\; & \{w u v p_v \mid w \in \Sigma_{AS}^*\,;\; u \in U(v)\} \cup && \textit{legitimate} \\
& \{w s a t w_v^a p'_v \mid w \in \Sigma_{AS}^*\,;\; w_v^a \in W_v^a(t)\} && \textbf{forged} \\
\text{with}\quad \hat{O}(p_v) \;=\; & \hat{O}(p'_v) \;=\; O(p_v) \\
\text{and}\quad \hat{U}(v) \;=\; & U(v)
\end{aligned} \tag{5.4}$$

Most notably, the attacker can hide his AS number while still forwarding between the upstreams s and t. In any case, the attack does not lead to suspicious topological changes. The observable sets of both origin and upstream ASes remain unchanged, while the forged route to p'_v generates the impression of originating from the victim's AS v and propagating via his legitimate upstream providers $U(v)$ (as well as theirs). With the attacker forwarding all traffic received for p'_v to the victim, no loss in connectivity can be observed either.

Although technically a strict subprefix hijacking with global impact, it is worth mentioning that a small part of the Internet remains unaffected. Since all ASes along the subpath $w_v^a \in W_v^a(t)$ ignore the attacker's update message, these ASes are effectively immune to the attack. Thus, customers of these ASes that do not employ multiple upstream providers do not receive the forged routes $\mathscr{F}_a(p'_v)$. However, the number of so-called *single-homed* ASes is in fact very small in general, as is the number of affected ASes along the subpath w_v^a, i.e. the limitation outlined above applies to a negligible number of ASes.

With respect to the example in Figure 5.1, assume that an attacker's goal is to eavesdrop all traffic destined to a fictitious webserver hosted on `10.1.119.68`. According to *Mallory's* BGP table, a series of N ASes AS_i is traversed on the path from *Oscar* to *Alice*. By advertising `10.1.119.0/24` with a manipulated AS path as follows, a man-in-the-middle attack of global impact can be launched:

$$
\begin{array}{lllll}
OP: & & AS_*^K \leftarrow \textit{Carol} \leftarrow & \textit{Alice} \ll \texttt{10.1.0.0/16} & \textit{legitimate} \\
OP: & & AS_*^L \leftarrow \textit{Dave} \leftarrow & \textit{Alice} \ll \texttt{10.1.0.0/16} & \textit{legitimate} \\
OP: & AS_*^M \leftarrow \boxed{\textit{Paul}} \leftarrow \boxed{\textit{Oscar}} \leftarrow AS_i^N \leftarrow \textit{Dave} \leftarrow & & \textit{Alice} \ll \texttt{10.1.119.0/24} & \textbf{forged}
\end{array}
$$

Example 5.4: Man-in-the-Middle Attack.

This approach induces a stable yet unobservable backhaul link from *Mallory* via *Oscar* and *Dave* to *Alice*, since none of the ASes inbetween accept the forged announcement. At the same time, *Paul* redistributes the announcement to all of his peers. Apart from a negligible number of intermediate single-homed customers as discussed above, the new route via *Mallory* is propagated globally. From a BGP perspective, the only noticeable change beside *Alice's* supposedly new subprefix announcement is a new link between *Paul* and *Oscar*. Due to the dynamic nature of peering agreements in the Internet, such minor routing updates occur countless times per day.

5.2.3 Hidden Takeover Attacks

In the following, we discuss an attack to take over unused Internet resources, i.e. network prefixes with $|\mathscr{L}(p_v)| = 0$ as given by condition 1) in Section 5.1.2. To hijack such a prefix, the attacker may choose among any of the aforementioned hijacking techniques. In the following, we assume that the attacker carries out AS hijacking, since it promises highest impact with lowest traceability:

$$
\begin{aligned}
\hat{\mathscr{L}}(p_v) &= \{\} \cup && \textit{legitimate} \\
&\quad \{wuvp_v \mid w \in \Sigma_{AS}^* \,;\, u \in U(a)\} && \textbf{forged} \\
\text{with} \quad \hat{O}(p_v) &= \{\} \cup \{v\} \\
\text{and} \quad \hat{U}(v) &= \{\} \cup U(a)
\end{aligned}
\tag{5.5}
$$

No conflicting sets of origin or upstream ASes arise, since the initial situation is defined by $\mathscr{L}(p_v) = O(p_v) = U(v) = \{\}$. Although the attacker's actions are observable as such, they appear to be legitimate actions of the victim. Especially in the case of AS hijacking, the attack merely creates the impression that the victim is re-establishing BGP connectivity via the upstream providers $U(a)$. Moreover, due to the nature of unused resources, possibly no one takes notice of an attack. The following example shows the unobtrusiveness of a hidden takeover attack (compare to Figure 5.1):

$$OP: \qquad AS_*^M \leftarrow \boxed{Paul} \leftarrow Bob \ll \texttt{10.2.0.0/16} \quad \textbf{forged}$$

Example 5.5: Hidden Takeover Attack.

The attacker *Mallory* fully acts on behalf of her victim *Bob*. The upstream ISP *Paul* is misled and unknowingly connects *Bob's* AS as instructed by *Mallory*, who successfully hides her true identity in complete absence of formal inconsistencies or topological conflicts.

5.3 Attack Tactics

In order to apprehend the attacking techniques as discussed so far, their applicability with respect to an attacker's goals has to be taken into account. Naturally, the field of application for different hijacking attacks depends on the attacker's motivation. We can differentiate between *destructive* and *abusive* variants.

5.3.1 Motivation behind Hijacking Attacks

Hijacking attacks may serve a variety of purposes. An obvious intent is to inflict damage to a victim's operations by disrupting network connectivity. Furthermore, hijacked networks can be abused for malicious activities, for instance to send large amounts of spam emails. More sophisticated attacks aim at compromising a victim's reputation by carrying out illicit actions on behalf of the victim. Lastly, attacks can be tailored to break confidentiality or integrity of a particular victim's communications. Some of these attacks require the attacker to cover his tracks or to deliberately impersonate a handpicked victim. In these cases, it is essential for an attacker to anticipate evidential changes to the Internet topology. Other attacks are launched in a *fire-and-forget* fashion without the need for deception.

Blackholing

The process of re-routing traffic in order to disrupt a victim's connectivity is called a *blackholing* attack, because the attacker's goal is to attract and absorb the victim's traffic like a black hole. Such incidents often arise from misconfiguration, but numerous malicious cases are documented, too. Blackholing attacks can be quickly detected by the victim and are thus of limited duration. Mitigation is usually based on counter-advertisements of more

specific prefixes to regain route domination. In addition, victims often receive great support by contacting the attacker's upstream providers to filter out their forged announcements.

Short-Term Abuse

Anecdotal evidence has been brought up that hijacking attacks are also employed for short-term abusive actions. For instance, extensive IP blacklists are widely installed by operators of mail servers to prevent spamming. Thus, having access to unlisted IP prefixes can be a great benefit for spammers. To this end, networks might be briefly hijacked in order to send huge amounts of spam emails. Massive spam campaigns can be launched within minutes, hence countermeasures to retake hijacked networks are of limited effect. After a successful spam campaign, the abused IP prefix is typically unusable, i.e. listed on numerous blacklists, and the spammers move on to a new network. For the victim, the damage caused by blacklisting and loss of reputation is often irreversible.

Long-Term Abuse

Another type of attack aims at a productive use of hijacked networks over the course of several weeks or even months. Given an unnoticed hijacking attack on idle parts of a victim's network, the attacker is able to firmly host illegal services, like phishing hosts or darknet web sites. A stealthily hijacked safe-house network can also be used to operate command and control servers for botnets, or to launch targeted attacks on client systems. In any case, it is essential for an attacker to evade detection over a substantial period of time, i.e. the attack must not interfere with the victim's operations and should not lead to noticeable changes in the Internet topology.

Impersonation

Hijacking attacks can further be utilized to impersonate randomly or selectively chosen victims. Activities conducted by the attacker in connection with the hijacked networks appear to be carried out by the victim himself. Hence, the attacker effectively hides his own identity and hinders disclosure by acting on behalf of the victim. This approach makes hijacking attacks even more attractive as it enables riskless network abuse. It hinders criminal prosecution and could even be used to deliberately create tensions between organisations or countries. Even after the discovery of such an attack, it is difficult for the victim to mitigate since it is often the word of one person against another.

	Origin Relocation		**Route Diversion**		**Hidden Takeover**
	Prefix Hijacking	*Subprefix Hijacking*	*AS Hijacking*	*Man-in-the-Middle*	*Abandoned Resources*
Destructive attacks					
Blackholing	o	+	+	−	−
Short-Term Abuse	−	+	+	−	−
Sustainable attacks					
Long-Term Abuse	−	−	o	−	+
Impersonation	−	−	+	+	+
Interception	−	−	−	+	−

Table 5.1: Attack tactics for BGP hijacking attacks.

Interception

The most sophisticated type of attack is presented by man-in-the-middle attacks. These attacks are heavily tailored to a specific victim and pursue the objective of stealthily rerouting a victim's traffic through the attacker's network. The crucial factor is to sustain global connectivity by relaying all communication back to the victim. In such a scenario, the attacker is able to break confidentiality by intercepting arbitrary messages. Moreover, the attacker acquires the ability to alter messages at will. A highly involved technique, man-in-the-middle attacks are often attributed to nation-state adversaries. But high profit attacks on banking, for instance, are conceivable for organized crime as well.

5.3.2 Eligibility of Attack Vectors

Prefix hijacking attacks partially disrupt a victim's connectivity, but are of limited use in other respects, like hosting malicious services, since a significant part of the Internet might still prefer the victim's routes. In contrast, subprefix hijacking attacks are capable of breaking communications entirely, i.e. all hosts inside the victim's network become globally unreachable. Hence, this type of attack is also useful for launching illegal operations from a hijacked network, and it is an important element for more sophisticated attacks like AS hijacking or man-in-the-middle interception. AS hijacking attacks further introduce the benefit of disguise such that stealthy long-term operations and impersonation attacks become feasible, as is the case with hidden takeover attacks. Man-in-the-middle attacks serve the specific purpose of impersonation and interception. Table 5.1 summarizes different tactics depending on the attacker's goals.

On the one hand, it is apparent that destructive attacks can be realized with the simpler type of origin relocation attacks. On the other hand, however, route diversion attacks serve this purpose just as well, while reducing traceability at the same time. In contrast, viable options to carry out abusive attacks do not comprise origin relocation at all. Instead, more elaborate attacks based on route diversion or hidden takeover techniques have to be employed. Having no real-world benefit apart from partially damaging a victim's network, it is surprising to see that state-of-the-art focuses primarily on prefix hijacking attacks.

6 CHAPTER SIX
Related Work and State-of-the-Art

NOTE *This chapter contains prior publication.*

Published in **IEEE Journal on Selected Areas in Communications–Special Issue on Measuring and Troubleshooting the Internet (JSAC-SI-MT), 2016.**
Sections 6.2 and 6.3 are based on previous work [8]. The state-of-the-art on attack detection in Sections 6.2.1 to 6.2.3 is elaborated in much greater detail.

SUMMARY A variety of studies has been carried out on routing anomalies and hijacking attacks. Empirical observations allow to assess the threat and impact of attacks on the global routing system. More focused studies provide insights into the root cause of anomalies and the actual motivation behind attacks. Corresponding analyses are presented in Section 6.1, which, in essence, establish the basis for state-of-the-art systems to detect and monitor attacks. In this respect, the detection of hijacking attacks can be divided into control-plane and data-plane techniques, with hybrid approaches being considered most promising in general. Related work in this area of research is comprehensively discussed in Section 6.2. Based on the author's classification of attacks (refer to Section 5.2), a detailed assessment of state-of-the-art approaches in Section 6.3 provides instructive insights into their capabilities. Surprisingly, most types of attacks cannot be fully addressed with current techniques. This finding motivates the development of new detection techniques that can cope with the full spectrum of attacks.

6.1 Empirical Studies

Many approaches have been proposed to analyze reachability problems, performance issues or routing anomalies in the Internet. A common way to detect and assess such problems is based on the analysis of control-plane information [130, 131, 132, 133, 134, 135, 136, 137, 138, 139]. These approaches evaluate routing protocols, e.g. by analyzing BGP update messages or RIB exports. Other techniques utilize information from the data-plane [140, 141, 142, 143, 144, 145, 146, 147, 148, 149]. To this end, active IP measurements are carried out in order to obtain a forwarding-based view of the global routing system. In addition, hybrid systems [150, 151, 152, 153, 154, 155, 156, 157, 158] exist, which combine and correlate both types of information. A common strategy is to monitor the control-plane for anomalies, which trigger measurements in the data-plane to gather a richer view of an event. In addition, some techniques draw on further data sources like router configuration files [159, 135] or log traces of affected systems [160, 161]. In order to study and detect BGP hijacking attacks, similar concepts can be applied.

6.1.1 Routing Anomalies

The first study on Multiple Origin AS (MOAS) conflicts [162] was carried out in 2001. Until then, MOAS conflicts were believed to be of long-lasting nature due to legitimate setups like multi-homing or static routes that are invisible to BGP. However, the authors analyzed 3.5 years of archived routing data and learned that the average duration of MOAS conflicts was only 31 days. While transitions of non-BGP networks from one AS to another could lead to such short-lived conflicts, the authors concluded that a more likely reason is misconfiguration. A later study [163] discussed several other reasons for MOAS conflicts and revealed even shorter durations for these events. Recently, the authors of [164] argued that short-lived MOAS conflicts are mostly recurring events and cannot be attributed to misconfiguration in general. Instead, several legitimate topology patterns were identified as the root cause of such anomalies. A detailed analysis of route leaks is presented in [165]. To this end, ASes that originate prefixes of a significant number of other ASes were identified. The analysis shows that such events occur several times per year and only last for a few hours for the most part. Unwanted routing of private or unallocated address space is discussed in [166]. Surprisingly, such *bogon* routes leak every day into the global routing system, with 40% of these incidents lasting longer than a day. This implies that filtering of invalid routes is incomplete or absent at some ASes. At the same time, the authors show that filter rules

for unallocated prefixes remain active up to two months after a prefix is newly allocated. In [167], the prominent China Telecom route leak is analyzed in greater detail. The work in [168] presents a study on RPKI-enabled prefixes. At the time of writing, 2% of all prefixes in the global routing table were covered by RPKI, while 20% of these prefixes were RPKI-invalid. The authors thus conjecture that misconfigurations frequently occur within the RPKI itself.

6.1.2 Impact of Hijacking Attacks

A comprehensive study on the impact of hijacking attacks on the Internet was first presented in [169]. The authors' goal is to quantify the number of ASes that are affected by invalid prefix announcements. Naturally, subprefix hijacking attacks, by which *all* ASes are affected, were beyond scope. Based on a propagation model derived from best-practice routing policies, the study shows that higher-ranking ASes in the Internet hierarchy are best suited to attract a significant fraction of a victim's traffic. The authors use their findings to study the applicability of man-in-the-middle attacks based on prefix hijacking. To this end, selective announcements are assumed in order to presevere an unpolluted route for the attacker to forward traffic to the victim. With a sound test setup, the effectiveness of such an approach is measured. Depending on the actual deployment, about 20% to 80% of traffic can be intercepted with this technique. In addition, the authors built up a system to detect interception attacks based on corresponding next-hop anomalies derived from control-plane and data-plane information. Although several anomalies were detected, no decisive evidence for real interception attacks could be found. Complementary results from an operational point of view are provided with [170]. A practical demonstration for the applicability of interception attacks is presented in [171]. Based on simulation, the effectiveness of interception attacks is further studied in [172]. In contrast, the authors of [173] discuss the resiliency of the Internet topology against hijacking attacks based on simulations on an AS graph with policy-annotated edges. The study shows that ASes with high topological connectivity are most resilient against hijacking attacks, while being most effective as attackers themselves at the same time. Further studies on the effectiveness of various detection schemes are presented in [174]. The ultimate goal is to draft a detection-assisted mitigation scheme based on so-called lifesafer ASes that can purge or promote routes. With detailed simulations, the authors show that such a reactive approach can increase the Internet's resilience against hijacking attacks.

6.1.3 Malicious Hijacking

A study of malicious spamming activities correlated to BGP updates is presented in [32]. The authors conjecture that spammers leverage *BGP spectrum agility*, i. e. briefly announce prefixes to send spam. Interestingly, the findings indicate that attackers supposedly announce less specific prefixes for their malicious activities. While anecdotal evidence is provided, the actual prevalence of such hijacking-based spam campaigns still remains unclear. A recent study on malicious BGP hijacking attacks [175] complements these findings. The authors utilize control-plane and data-plane monitoring in combination with orthogonal data sets from spam traps and blacklists. Based on 18 months of data, a significant number of malicious hijacks could be identified. The analysis shows that spectrum agility is indeed a real threat. However, the study also implies that attackers prefer to abuse unannounced address space in order to send spam more stealthily.

6.2 Detection and Monitoring

Beside general studies, tailored techniques search the control-plane of BGP for specific anomalies that arise when an attacker injects erroneous routing information. In more specific terms, such techniques build a model of the Internet topology based on information extracted from routing tables and trigger alerts whenever a new advertisement conflicts with this model. While offering a high detection rate, these approaches usually suffer from high rates of false positives due to topological similarity between hijacking anomalies and benign BGP engineering practices. Other approaches look for significant connectivity changes in the data-plane. Further characterization of routers and networks along the forwarding path from a vantage point to a monitored network helps to distinguish between benign and malicious routing changes. Naturally, hybrid detection systems utilize both types of information or incorporate orthogonal data sources. In the following, a detailed overview of state-of-the-art detection concepts is presented. An assessment of strenghts and weaknesses of individual techniques is given at the end of this chapter.

6.2.1 Control-Plane Techniques

The first of its kind, a Prefix Hijack Alert System (PHAS) [176] was proposed as a monitoring scheme for prefix hijacking attacks. PHAS aggregates multiple BGP feeds and monitors prefix origin changes. An adaptive time-window based mechanism prevents recurring

alerts for similar events in an effort to reduce high rates of false positives. The system is not designed to provide a global view of hijacking incidents. It is rather aimed at individual operators to monitor their own prefixes. As a consequence, the distinction between legitimate and invalid origin changes is left to registered users. PHAS further provides means to reliably notify operators during an attack in the light of the fact that regular communication channels might become inaccessible. The detection scheme works best for ordinary prefix hijacking and it is also well-suited to monitor unused address space. Moreover, the authors sketch extensions to PHAS to detect subprefix hijacking in the future. It is, however, unclear how the higher number of false positives resulting from the far more wide-spread nature of corresponding subMOAS anomalies could be reduced. Hidden takeover attacks can be detect from an operator-centric point of view. Due to its focus on origin conflicts, PHAS is unable to detect route diversion attacks in which no origin changes occur.

An improved system to detect bogus BGP routes is presented in [177]. The system aims at providing a global view of attacks in real-time, while significantly reducing the rate of false positives as compared to PHAS. To this end, BGP messages are evaluated to create a mapping of prefixes and origin ASes together with directed AS links that are extracted from AS paths. After a learning phase, a directed AS-level topology is derived. Several heuristics are proposed based on general assumptions on routing policies, on the intended outcome of attacks, and on best practices for managing IP resources. These heuristics are designed to assess the validity of newly observed AS links and prefix origins. By applying the heuristics to individual AS links on a new routing path, the approach may yield the root cause of a bogus route. Its detection capability with respect to origin relocation is comparable to PHAS, and, in theory, the system is able to detect all types of hijacking attacks. In practice, however, a complete AS-level topology can hardly be obtained. As a consequence, careful tuning of heuristic parameters is necessary to yield suitable detection results for route diversion attacks. The same is true for assessing the validity of origin changes. The proposed heuristics tend to reduce false alarms at the price of an increased rate of false negatives and are thus of limited effectiveness. A similar approach to pinpoint an attacker in the AS-level topology is presented with LOCK [178]. The Buddyguard system [179] uses a learning-based approach to detect abnormal routing changes. Re-routed prefixes are compared with a reference set of prefixes that showed similar routing behaviour before. The creators of BGPmon.net [180] provide MOAS and subMOAS alarms in real-time, but do not publish details about their methodology.

6.2.2 Data-Plane Techniques

A light-weight distributed measurement scheme (LWDS) to detect hijacking attacks in real-time is proposed in [181]. This system is based on the observation that routing paths are relatively stable in general. Since the actual network location of a prefix rarely changes, corresponding BGP updates are assumed to be of limited scope. Consequently, traceroute path measurements are expected to yield stable path lengths for the majority of networks. With the help of multiple vantage points to initiate measurements, major violations of this conjecture may hint at a suspicious change in network location, while singular discrepancies indicate legitimate route updates. To account for valid changes in the Internet topology, carefully chosen reference points close to a monitored network are utilized. Under the assumption that major routing changes affect routes to both the reference point and the monitored network, a hijacking attack would lead to a significant path discrepency between the respective routes. In this regard, the authors assume that the path to such a reference node is a subset of the path to a monitored networkas this node is on the same path and close to the network. Path disagreement is computed on the AS-level to account for dynamic intra-AS route selection. Further considerations apply to the appropriate selection of vantage points. Parameters for hop count and path disagreement metrics are calibrated empirically. The proposed system is able to detect both hijacking of prefixes and subprefixes as well as route diversion attacks. It is unfeasible, however, to deploy it on a larger scale due to its dependency on suitable vantage points and reference nodes. The monitoring of unused address space is not part of the approach. A comparable technique that leverages ping tests, i.e. latency measurements, from multiple vantage points is proposed in [182]. Beyond that, the StrobeLight system [183] detects hijacking attacks in a similar way while being limited to a per-operator deployment.

Another operator-centric approach to detect hijacking attacks from a data-plane perspective is given with iSpy [184]. Its key idea is to create a prefix owner's view of reachability by carrying out traceroute measurements from an operator's network to major transit ASes. These measurements are performed periodically in order to detect subsequent differences. If an operator becomes the victim of a hijacking attack, a significant number of outgoing measurements will cease to yield results since replies are redirected to the attacker. In contrast, a small number of invalid measurements indicate a temporary link failure and not an attack. The system relies on a detection threshold, which is empirically derived. iSpy solely detects prefix hijacking attacks since its inherent detection metric is based on the identification of a polluted and an unpolluted part of the Internet. As a matter of fact, this de-

tection scheme cannot differentiate between subprefix hijacking incidents and temporary link failures or congestion near the victim's network. Attacks on unused address space or man-in-the-middle attacks are beyond the scope of the approach.

6.2.3 Hybrid Approaches

One of the first hijacking detection systems to include both passive and active measurements is a fingerprint-based approach [185]. This system utilizes a monitoring scheme for BGP messages to infer candidate hijacking attacks similar to previous techniques. In general, all route updates are classified as suspicious or valid events. Suspicious events are further divided into four types: hijacking of (sub-)prefixes with or without a manipulation of AS paths. Depending on the type of a potential attack, an active probing module is launched. For regular prefix hijacking, i. e. MOAS conflicts, distributed fingerprints are collected to identify two disjoint parts of the Internet. The idea is that hosts in the part that is affected by an origin change would exhibit different fingerprints than victim hosts in the unaffected part. Multiple fingerprinting techniques like OS detection, IP identifier probing, and TCP/ICMP timestamp queries are used to increase confidence. Route diversion attacks that do not exhibit MOAS conflicts are identified with the help of various heuristics. These include popularity measures of AS edges and the evaluation of geographic constraints based on third-party databases. For suspicious events, the fingerprinting technique is applied in a similar manner. Due to longest prefix match forwarding, subprefix attacks affect the Internet as a whole and do not lead to two disjoint parts of the Internet. As a consequence, these attacks cannot be evaluated with fingerprint-based techniques. Instead, the aforementioned heuristics are used together with the detection of possible violations of routing policies based on inferred customer-provider relationships. BGP updates that pass these filters are actively probed with a so-called reflect scan. This is an IP spoofing based technique, which relies on various assumptions like the ability to spoof IP addresses and a certain deployment of the attacker's and the victim's hosts. Generally speaking, the heuristics show the same limitations as in previous work. The approach performs best for regular prefix hijacking, if suitable hosts are available for scans. The detection capability for subprefix hijacking is limited due to the absence of fake AS edges and the limited scope of the reflect scans. Attacks on unused IP blocks are beyond the scope of the approach. The authors of [186] extend the concept of reflect scans to provide a more reliable detection of prefix hijacking. In [187], other types of fingerprints are proposed, e.g. DNS servers observed via reverse DNS lookups or active probing techniques to infer network firewall policies.

A more practical approach to detect hijacking attacks is realized with the Argus system [188]. This framework detects three different types of anomalies in BGP feeds. First, origin anomalies are infered if the origin AS of a prefixes changes, or if a new prefix appears. Second, a newly observed pair of neighboring ASes in an AS path leads to the reporting of an adjacency anomaly. Third, Argus accounts for policy anomalies by monitoring AS triplets. If a corresponding anomaly occurs, live IP addresses for the affected prefixes are identified with the help of publicly available measurement data and by analyzing DNS records. To confirm hijacking attacks, Argus leverages looking glasses to obtain BGP routes and traceroute measurements from a large number of vantage points. Given a reported prefix anomaly, corresponding data sets are repeatedly collected for active IP addresses in the respective prefix in order to identify a poisoned and an unpoisoned part of the Internet. If the aggregated vectors of control-plane and data-plane results show a strong correlation for a split view, a hijacking attack is assumed. Due to the broad availability of looking glasses, a larger fraction of anomalies can be analyzed compared to previous approaches. The detection technique works best for regular prefix hijacking attacks both in origin relocation and route diversion variants. Since subprefix hijacking attacks do not yield two disjoint parts of the Internet, Argus is of limited use as soon as malicious subprefixes reach global visibility. The monitoring of unused address space is not part of the presented methodology. A complementary approach to detect the immediate effects of hijacking attacks on the data-plane is to perform measurements during and after an attack. The SpamTracer project [189] utilizes this technique on networks that exhibit changes in the control-plane and emit huge amounts of spam emails at the same time.

6.3 Assessment and Comparison

In summary, it is surprising to see that related work concentrates its effort on the detection of prefix hijacking attacks, despite the fact that this kind of attack is of very limited use for an attacker in practice (refer to Table 5.1). As a matter of fact, subprefix hijacking and especially route diversion attacks offer a broader area of operations, but are considered circumstantially in state-of-the-art techniques. Moreover, hijacking attacks on abandoned Internet resources are almost entirely neglected in the literature.

Table 6.1 lists the detection capabilities for various techniques presented above. Two prominent state-of-the-art representatives were chosen for comparison for each of the general approaches as discussed in Sections 6.2.1 to 6.2.3. These techniques are assessed with

	Origin Relocation		**Route Diversion**		**Hidden Takeover**	**Applicability**
	Prefix Hijacking	*Subprefix Hijacking*	*AS Hijacking*	*Man-in-the-Middle*	*Abandoned Resources*	
control-plane detection						
PHAS [176]	+	o	–	–	+	o *(operator-level)*
Bogus Routes [177]	+	o	o	o	o	+ *(global scale)*
data-plane detection						
LWDS [181]	+	+	o	o	–	– *(incident-level)*
iSpy [184]	+	–	–	–	–	o *(operator-level)*
hybrid techniques						
Fingerprints [185]	+	o	o	–	–	+ *(global scale)*
Argus [188]	+	o	o	–	–	+ *(global scale)*

+ practical usefulness o theoretic applicability – not supported

Table 6.1: Comparison of state-of-the-art hijacking detection systems.

respect to their practical usefulness to detect different types of hijacking attacks. In addition, their applicability is appraised based on characteristic requirements imposed by individual techniques. Operator-level techniques, for instance, need to be installed at an ISP's network and are thus less effective than detection systems of global scale. Some techniques are applicable on a case-by-case basis only, i.e. on incident-level, possibly due to the necessity of suitable measurement nodes or specifically placed landmarks that respond to probes. Others depend on an attacker carrying out malicious actions such that orthogonal observations can be taken into account, e.g. an increased spamming activity that originates from a certain network. Hence, for an overall assessment of the detection techniques, their applicability needs to be taken into account. The author's own contributions will be similarly assessed and compared to these approaches in Section 14.2.

PART TWO

Discovering Routing Attacks

7 CHAPTER SEVEN
Legitimate Routing Anomalies

NOTE *This chapter contains prior publication.*

Published in **IEEE Journal on Selected Areas in Communications–Special Issue on Measuring and Troubleshooting the Internet (JSAC-SI-MT), 2016.**
Sections 7.1, 7.2 and 7.4 are based on previous work [8]. A discussion of ethical concerns is added to Section 7.4.

Published in **Proceedings of the International Workshop on Traffic Monitoring and Analysis (TMA), 2015.**
Sections 7.1, 7.2.1 and 7.2.2 incorporate previous work [7] providing technical background and comparative results. No substantial changes are made.

Published in **Proceedings of the IEEE ICC Communications and Information Systems Security Symposium (ICC CISS), 2014.**
Section 7.3 is based on previous work [5]. No substantial changes are made.

SUMMARY In this chapter, we develop the *Hijacking Event Analysis Program (HEAP)*, a new approach to assess the legitimacy of hijacking alarms. Suffering from high rates of false positives, state-of-the-art techniques fall short to detect elaborate kinds of hijacking attacks in a reliable manner. We improve on this situation by taking into account multiple data sources to disprove malicious intent for benign routing anomalies. Section 7.1 provides an introduction to our approach. In Section 7.2, we design the architecture for our analysis framework. To this end, we assess the usefulness of Internet Routing Registries, carry out large-scale SSL/TLS measurements, and incorporate a topology-based reasoning algorithm [164]. We further present use cases for our previous work on IP geolocation [1] and traffic flow analysis [2, 5]. Section 7.4 addresses limitations to our approach and discusses ethical concerns.

7.1 Introduction and Overview

We present the *Hijacking Event Analysis Program (HEAP)*, a novel approach to investigate routing anomalies. With HEAP, our goal is not to raise new types of alarms. Instead, HEAP receives input from readily available detection systems, i.e. from related work, in an effort to reduce the high rates of false alarms inherent to these techniques. We thereby leverage several unique data sources that can reliably disprove malicious intent.

First, we use information from IRR databases to infer business and management relationships between IRR objects. Such information can only be altered with valid access credentials. Our assumption here is that an attacker does not have these credentials. Second, we use data from Internet-wide scans of the SSL/TLS landscape to collect trusted public keys from corresponding hosts. After having established a ground truth, these hosts will serve as landmarks. If their public key remains the same during a routing anomaly, we are still dealing with the same hosts and can rule out malicious interference. The assumption we make here is that an attacker cannot compromise all of a victim's hosts to steal their keys. Third, we use a heuristic topology algorithm to reason whether a hijacking alarm indicates that a suspected attacker hijacks his own upstream provider's networks. This is an unlikely attack as the victim could simply filter out malicious route updates or even shutdown the attacker completely. In addition, we discuss the usefulness of traffic analysis in order to assess the root cause of suspicious routing anomalies. With access to traffic recordings obtained from a network with a large number of end users, we are able to study client behaviour during the occurrence of routing anomalies. An increase in malicious activities may indicate an attack, which enables the assessment of the attacker's motivation. Furthermore, we built a system to geographically locate arbitrary Internet hosts based on active measurements, which can be leveraged to gain further insights into the nature of routing anomalies.

Arguably, massive scans of the entire Internet can be considered intrusive. We address this issue by an open communication policy, and by maintaining a blacklist for complaining network operators. Moreover, the analysis of real user traffic concerns individual-related data and needs to be carried out with due diligence. We limit our analysis to meta data only and restrict access to the collecting machines.

7.2 The *HEAP* Framework

HEAP leverages three distinct data sources to assess hijacking events. Our main assumption here is that an attacker is capable to hijack networks in BGP, but cannot alter or-

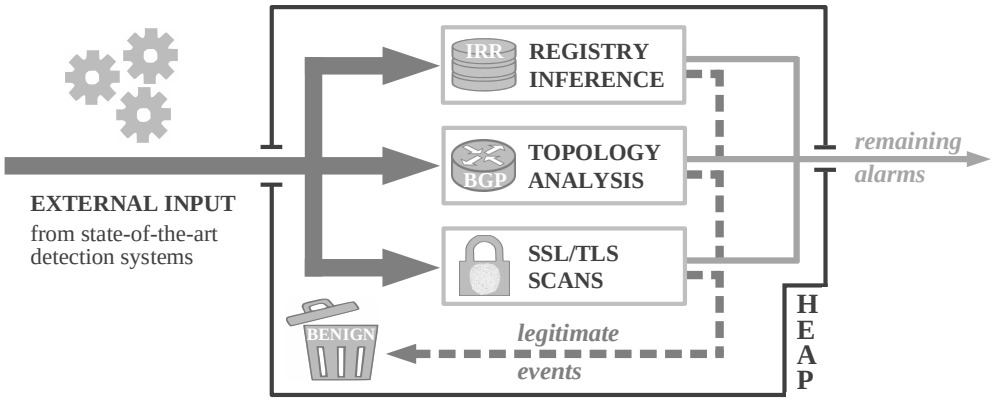


Figure 7.1: The Hijacking Event Analysis Program (HEAP).

thogonal data sources that relate to the operation of hijacked networks. Hence, we are able to rule out an attack if these data sources legitimize a suspicious routing anomaly. First, Internet Routing Registries (IRR) are utilized to infer legitimate business relationships between an attacker and his alleged victim. Second, a topological analysis is carried out in order to identify benign anomalies resulting from common operational practices. And lastly, SSL/TLS measurements yield cryptographic assurance that traffic to a supposedly hijacked network is still routed to the alleged victim.

Figure 7.1 illustrates the main workflow within HEAP. Given external alarms fed into the system, legitimate events are identified and eliminated as false positives based on the aforementioned data sources. Such events can be generic routing anomalies, like common sub-MOAS conflicts (refer to Subsection 5.2.1), for instance, or more elaborate hijacking alarms generated by state-of-the-art detection systems. Note that the SSL/TLS component needs a tight coupling to external systems that provide us with input alarms, since corresponding scans have to be carried out in response to the input received. The remaining events are highly suspicious indications for an attack, even if we take into account that we cannot make further assumptions about their nature. This is for two reasons: 1) the input source already provides potential hijacking candidates, and 2) none of our filter techniques yields evidence for a legitimate cause. In the following, we will show that this is indeed improbable for benign events. The remaining alarms lend themselves well to manual inspection, with a rich set of background information readily available from the individual analysis steps.

All filters applied by HEAP are executed concurrently. HEAP is easily extensible, i.e. additional filters can be incorporated without difficulty. At the moment, three independent techniques to eliminate legitimate alarms have been implemented and tested.

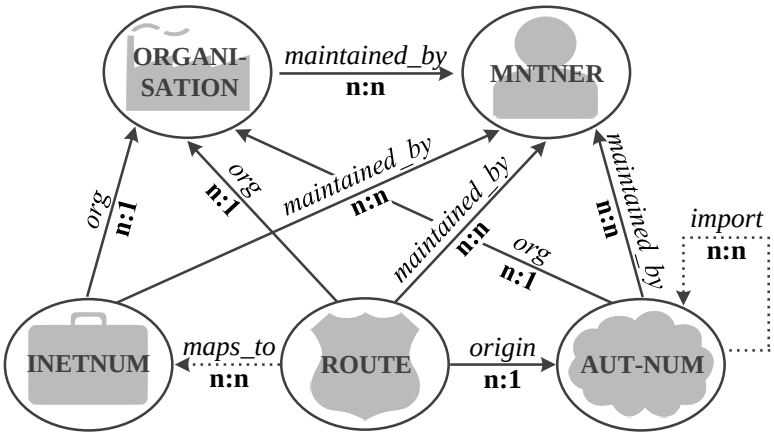


Figure 7.2: Entities and relations in the RIPE database.

7.2.1 Utilizing IRR Databases

Regional Internet Registrars (RIRs) maintain Internet Routing Registries, i.e. databases that contain information pertaining to the management of Internet resources (refer to Section 3.3.1). A recent study [190] matched prefixes and ASes observed in BGP and IRRs by looking for appropriate database objects. We provide a more generalized set of inference rules to identify benign routing events, which take into account complex relationships between affected prefixes and suspicious ASes.

IRR Data Model

The fundamental assumption behind our approach is that an attacker does not have the credentials to change an IRR database in order to cover his attack. To disprove an attack, we accordingly look for legitimizing database relations between the entities involved in a hijacking alarm, e.g. for a common organisation referenced by two ASes. To this end, we download and evaluate snapshots of the IRR databases, which are provided by RIRs on a daily basis. We use a graph database to store the extracted information using the schema presented in Figure 7.2 and track all changes over time. Note that IRR databases are updated by individual resource holders and can thus be outdated or even hold conflicting information. Our filter accounts for this by strictly searching for legitimizing relationships without drawing any conclusions in their absence.

To assess a given hijacking alarm, we map the affected AS numbers and prefixes to resource objects, i.e. graph nodes, in our graph database. We then traverse the graph along a

	RIPE	
instance	nodes	relations
`MNTNER`	**48,465**	—
←*maintained_by*– `[*]`	—	5,307,883
`ORGANISATION`	**81,260**	—
←*org*– `[*]`	—	199,644
`AUT-NUM`	**27,616**	—
←*origin*– `ROUTE`	—	245,831
←*import*– `AUT-NUM`	—	221,690
`INETNUM`	**3,871,827**	
`ROUTE`	**236,604**	—

Table 7.1: Data stored in our graph database. June, 2014.

path of legitimizing relations that document a right to use, which are given by `AUT-NUM` and `INETNUM` objects linked by *import*, *origin*, *maintained_by* or *org* relations. We look for such paths between a) two affected ASes, or b) a prefix and its origin AS. If we succeed with a), we can infer a valid business relationship between the victim and the suspected attacker. If we succeed with b), the suspected attacker holds ownership rights for the prefix and is thus authorized to originate the prefix from his AS.

Our filter supports IRR databases as provided by all five RIRs. Note that we transform the databases into the RIPE data model as presented in Figure 7.2, since it is most consistent and represents a superset of all available information. In fact, AfriNIC and APNIC already store their data in a similar format and can thus be directly processed by our filter. LACNIC and ARIN utilize their own data models, which can be converted in spite of some missing data points.

RIPE-based IRR databases model access rights with the help of `MNTNER` objects. Only those maintainers with valid credentials can modify or delete objects. For any object, this is expressed by adding a *maintained_by* reference pointing to the respective `MNTNER` object. `ORGANISATION` objects are mainly used to provide administrative contact details. For privacy reasons, most IRR database snapshots do not include details, but unique references to these objects are preserved. `INETNUM` objects document allocated or assigned IPv4 prefixes managed by the respective RIR. `AUT-NUM` objects represent AS numbers and may be referenced as the *origin* of `ROUTE` objects. Such `ROUTE` objects are created by resource hold-

instance	AfriNIC nodes	AfriNIC relations	APNIC nodes	APNIC relations	[1]ARIN nodes	[1]ARIN relations	LACNIC nodes	LACNIC relations	RIPE nodes	RIPE relations
MNTNER	**2,624**	—	**20,129**	—	**n/a**	—	**n/a**	—	**53,670**	—
←*maintained_by*– [*]	—	133,186	—	1,919,397	—	n/a	—	n/a	—	5,620,385
ORGANISATION	**1,877**	—	**n/a**	—	**2,976,707**	—	**n/a**	—	**90,102**	—
←*org*– [*]	—	32,476	—	n/a	—	3,536,502	—	n/a	—	249,319
AUT-NUM	**1,239**	—	**9,485**	—	**24,939**	—	**5,193**	—	**29,206**	—
←*origin*– ROUTE	—	464	—	216,865	—	[2]583,296	—	n/a	—	279,532
←*import*– AUT-NUM	—	6	—	10,734	—	n/a	—	n/a	—	228,509
INETNUM	**85,672**	—	**924,584**	—	**2,910,623**	—	**342,104**	—	**3,995,522**	—
ROUTE	**443**	—	**97,858**	—	[2]**600,940**	—	**n/a**	—	**267,216**	—

Table 7.2: Data stored in our graph database. August, 2015.

[1] ARIN's object identifiers can be directly mapped to RIPE's schema (e.g. ASHandle → AUT-NUM).

[2] Implicitly given in ARIN's INETNUM objects (via OriginAS attributes).

ers and are used to document or confirm intended prefix announcements by specific ASes. To create a ROUTE object, the resource holder needs to provide valid credentials for the respective INETNUM and AUT-NUM objects. A corresponding *maps_to* relation is derived by our parsing algorithm, as is the case with *import* relations deduced from free-text description fields, which are often used to model routing policies in RPSL (see Sections 3.3.1 and 4.3.1). When resources are deleted from a database, RPSL definitions may still reference (now) non-existing ASes. We account for this by tracking such orphaned *import* relations.

Since February, 2012, we download and evaluate daily snapshots of the RIPE database. As of August, 2015, we added support for all five RIR databases. All RIR databases can be transformed into the RIPE data model, albeit some data points might not be provided by certain RIRs. Our database currently holds more than 15 million nodes and 45 million relations extracted from the five databases. Tables 7.1 and 7.2 provide details for selected objects that are relevant to our approach. Entries marked with *n/a* are not available in the respective database snapshots. Note that we contrast two analyses that are 14 months apart.

For the RIPE database in 2015, for instance, we can see that less than 55,000 MNTNER objects share more than 5 million incoming *maintained_by* references. Although optional, roughly 90,000 ORGANISATION objects are referenced by 250,000 other objects. About 280,000 ROUTE objects bind prefix announcements to less than 30,000 AUT-NUM objects. Furthermore, these AUT-NUM objects document nearly 230,000 *import* routing policies. In the further course of this thesis, we will see that these figures allow our filter to be highly effective—except in the case of LACNIC, where none of the necessary cross-referencing objects are provided in the daily database snapshots.

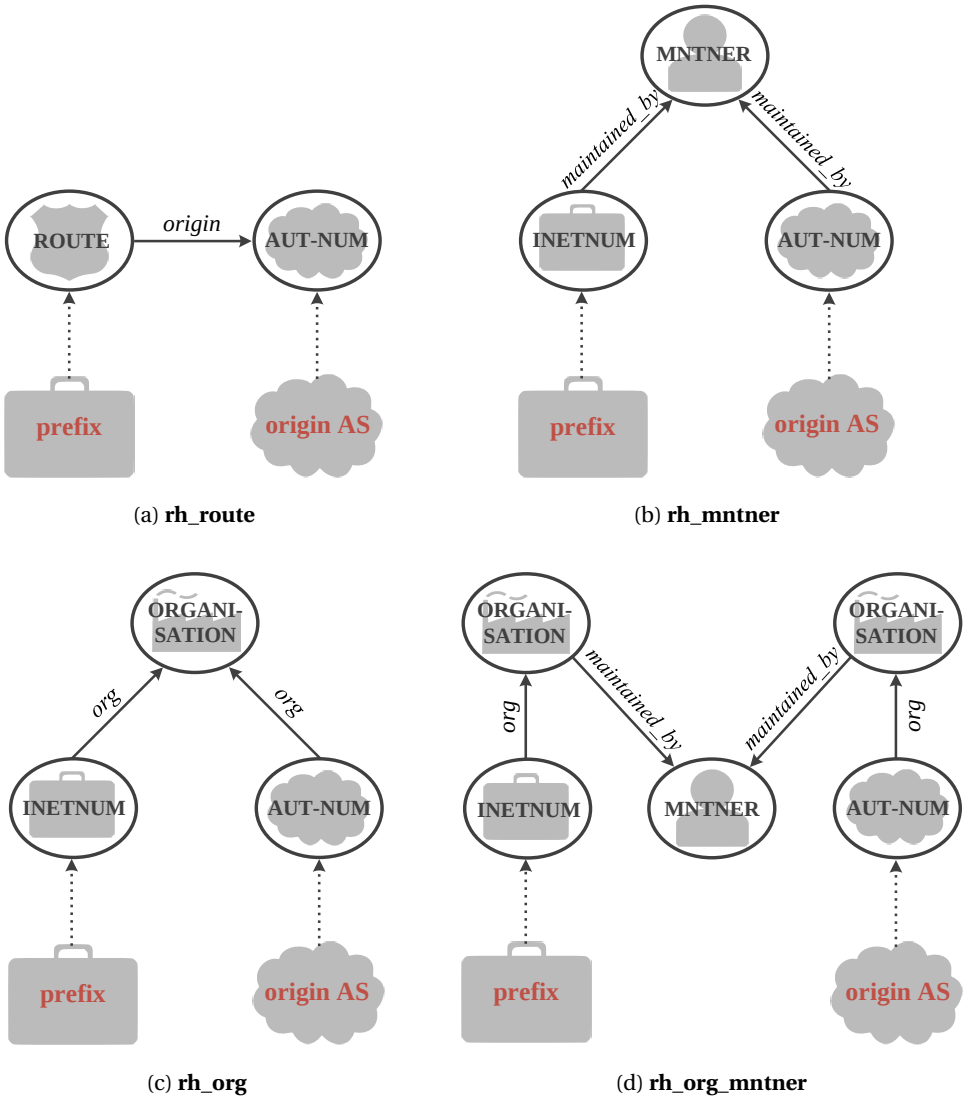
Filtering Legitimate Events

With our filter, we aim to assess arbitrary routing events. To be effective, however, any given routing anomaly that is fed into the IRR filter should at least relate to an IP prefix and its origin AS, i.e. an attacker's AS. Beyond that, hijacking alarms may also concern a victim's AS and possibly a less specific prefix as well. Given this input data, we use our graph database to identify legitimate paths between the prefix—if applicable the more specific one—and its origin AS, or both origin ASes respectively. Such paths are formed by one or more of the following relations: *import*, *origin*, *maintained_by*, and *org*. Figures 7.3 and 7.4 illustrate the complete set of our inference rules based on these relations. Entities without surrounding circles represent input data, whereas encircled items represent nodes in our graph database.

To identify legitimate events, we initially start by checking if an attacker is in fact the legitimate holder of a prefix resource in question. To this end, we query our database for the respective `AUT-NUM` object and first check if we can map the prefix to a `ROUTE` object. If it exists, we search for an *origin* relation to the suspected attacker's `AUT-NUM` object (Figure 7.3a). To create such a `ROUTE` object, valid maintainer credentials are needed for the `AUT-NUM` object, but, more importantly, also for the `INETNUM` object represented by the prefix. If the alleged attacker is able to provide both, we consider him the owner of the prefix, and the case at hand to be legitimate. We also check for `ROUTE` objects that bind less specific prefixes to the origin AS. This implies that the attacker actually has control over the corresponding larger IP range, of which only a part is related to the alarm. As network operators are free to announce their networks in any given size, such cases are legitimate, too.

Similar arguments apply for a prefix and its origin AS that have relations to a common `MNTNER` object (Figure 7.3b), since these relations represent shared access rights. Relations to a common `ORGANISATION` object (Figure 7.3c) and even a path from different affected organizations to a common `MNTNER` (Figure 7.3d) can further be considered strong evidence for a legitimate routing anomaly. It is worth mentioning that once again, we do not look for exact matches to the `INETNUM` object, but also search for objects representing less specific prefixes, since a resource holder is not required to announce his prefixes as a whole.

Our second set of inference rules is designed to find evidence for a legitimate business relationship, i.e. that two supposedly oppenent origin ASes in fact cooperate. We first look for an *import* relation from the alleged victim to the attacker (Figure 7.4a). Such a relation would imply that the suspected victim deliberately updated the RIR database to document his willingness to accept the attacker's route updates. As it is highly unlikely for a victim to



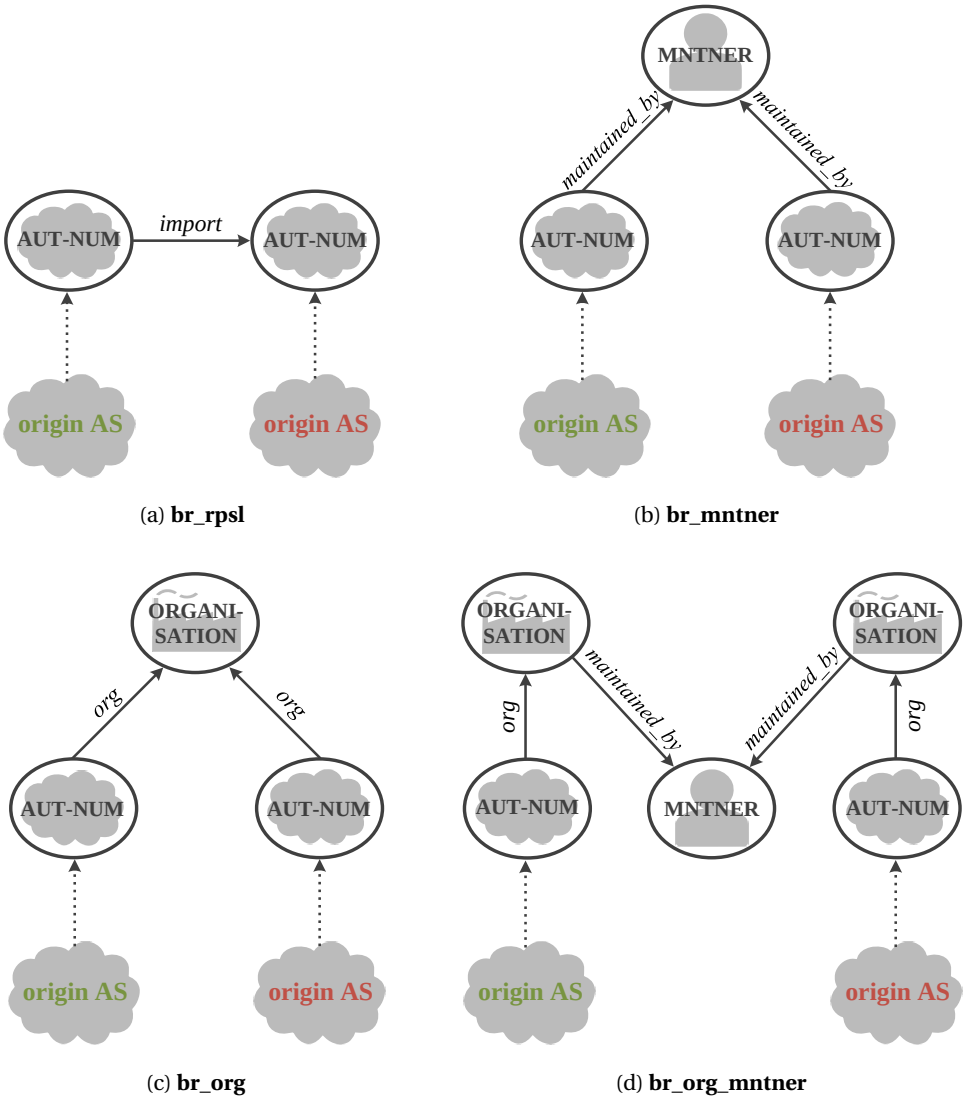
(a) **rh_route** (b) **rh_mntner**

(c) **rh_org** (d) **rh_org_mntner**

Figure 7.3: IRR legitimization rules – *legitimate resource holders*.

grant his attacker such privileges, we consider the existence of an *import* relation to be proof of a legitimate business relationship and thus conclude that the event is not an attack. The remaining rules in Figures 7.4b to 7.4d are similar to those in Figures 7.3b to 7.3d. Again, we try to identify a legitimizing path based on shared `MNTNER` or `ORGANISATION` objects. In this context, however, this path is to be found between two given origin ASes.

Note that our approach does not require an IRR database to be complete. A lack of information only implies that we are unable to legitimize certain events. Databases do not have to be conflict-free either, since we completely ignore such information. Again, this



(a) **br_rpsl** (b) **br_mntner**

(c) **br_org** (d) **br_org_mntner**

Figure 7.4: IRR legitimization rules – *legitimate business relationships.*

implies a potentially lower validation rate but on no account falsely legitimized events.

The figures presented in Tables 7.1 and 7.2 indicate that our set of rules has great potential to be effective. For the RIPE database in 2015, for instance, `MNTNER` objects are connected to 105 other objects on average, which yields a high probability to identify factually legitimate events with the according `MNTNER`-based filter rules. In addition, we observe a complex policy network that can be leveraged. In this respect, every RIPE `AUT-NUM` object is linked to roughly ten other Internet resources via *origin* and *import* relations each, on average. While some RIR databases do not provide the necessary data for each of our filters,

we still have access to virtually all autonmous systems, i.e. 70,000 `AUT-NUM` objects, and to more than 8 million `INETNUM` objects, which can both serve as entry points to our graph. As a consequence, we are able to cover a large number of possible events, albeit varying quality of RIR databases. Altogether, the data highly suggests that each of our individual filter rules offer unique potential to contribute to the reliable identification of legitimate events.

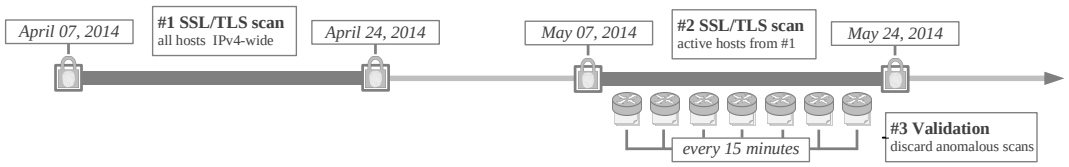
7.2.2 Cryptographic Assurance with SSL/TLS

We propose another strong filter that is based on regular Internet-wide scans of SSL/TLS protocols. For any given hijacking alarm concerning a certain IP prefix, we verify if affected SSL/TLS hosts present the same public key before and during the event. We make the assumption that an attacker cannot gain access to the private keys of a victim's hosts and thus cannot perform successful SSL/TLS handshakes. We conclude that such cases cannot be attacks.

A prerequisite for this filter is a *ground truth scan* to obtain a known-correct mapping from IP addresses to public keys that are used on corresponding machines. Given such a ground truth data set, we can carry out *validation scans* to hosts in a prefix that relates to a hijacking alarm and compare the retrieved public keys. Note that it is imperative for these scans to be executed in a timely manner, i.e. we need to compare public keys during the life time of an event. A tight coupling to the alarming system is dispensable if we can retroactively ascertain that an event lasted for the entire duration of a corresponding scan. Since our system is designed to assess subprefix hijacking attacks, which affect the Internet as a whole (refer to Section 5.1.2), the vantage point for our SSL/TLS measurements can be chosen freely. For this paper, we employed a scanning machine hosted at our university in Munich (AS56357).

Large-scale Scans

In recent years, large-scale measurements of the entire Internet have become feasible. The first of its kind, a study on larger parts of the global SSL/TLS deployment [191] was published in 2011 for the top one million web sites. In the same year, a long-term study of the X.509 PKI (refer to Section 3.2) utilizing both active and passive measurements [192] followed. Rapid developments on large-scale scanning tools [193, 194] led to variety of Internet-wide studies, not least with respect to SSL/TLS [195, 196].



(a) IPv4-wide ground truth. (b) Rescan of active hosts.

Figure 7.5: Timeline for obtaining our SSL/TLS ground truth – *first analysis period.*

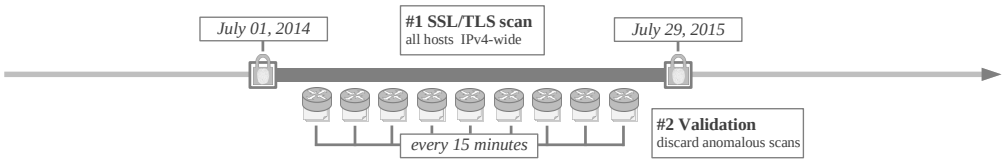


Figure 7.6: Timeline for obtaining our SSL/TLS ground truth – *second analysis period.*

Obtaining a Ground Truth

For our SSL/TLS filter, we need to obtain a ground truth data set that ideally covers all possible prefixes that might be fed into the filter. To this end, we first carry out a SSL/TLS scan of the entire routable IP space. To reduce intrusiveness and to avoid probing packets being dropped at destination networks, we carry out our scans much more slowly than it would be technically possible, e.g. by using tools like zMap [193]. Based on lessons learned from previous scans, we further notify a number of well-known Computer Emergency Response Teams (CERTs), as well as other research institutes and operators of blacklisting services before a scan. Naturally, we maintain our own blacklist of networks that must not be scanned again, based on live feedback from operators of previously scanned networks. The result of these scans is a mapping of IP addresses to trusted public keys. Further details can be found in [197].

It is worth mentioning that we have to ensure that no attacker possibly interfered with the measurement process by means of routing attacks. This requirement is addressed by discarding measurements that were affected by routing anomalies. Since we naturally do not have any knowledge about ongoing attacks at this stage, we extract all MOAS and sub-MOAS conflicts (see Subsection 5.2.1) from BGP routing tables. We assume that this set of anomalies is a larger superset of routing attacks. If an individual SSL/TLS measurement, and its target IP address respectively, is affected by such a conflict, we discard the measurement. As a matter of fact, we further relax this constraint to a grace period of two hours in order to account for BGP convergence effects. This procedure invalidates even more measure-

	port	duration	port open	handshake	in %
Implicit SSL/TLS		27d	**72,546,563**	**38,146,816**	**52.58%**
HTTPS	443	10d	42,676,912	27,252,853	63.85%
SMTPS	465	2d	7,234,817	3,437,382	47.51%
IMAPS	993	3d	6,297,805	4,121,108	65.43%
POP3S	995	3d	5,186,724	2,797,300	53.93%
FTPS	990	2d	2,657,680	344,400	12.95%
LDAPS	636	2d	2,273,771	112,978	4.96%
XMPPS/CLIENT	5223	2d	2,223,994	70,441	3.16%
XMPPS/SERVER	5270	1d	2,046,204	1,693	0.08%
IRCS	6697	2d	1,948,656	8,661	0.44%
Explicit SSL/TLS		9d	**51,768,705**	**18,316,920**	**35.38%**
FTP/STARTTLS	21	2d	14,493,966	2,939,048	20.27%
SMTP/STARTTLS	25	2d	12,488,000	3,848,843	30.82%
POP3/STARTTLS	110	1d	8,930,688	4,074,211	45.62%
IMAP/STARTTLS	143	2d	8,006,617	4,076,809	50.91%
SUBMISSION/STARTTLS	587	2d	7,849,434	3,378,009	43.03%
Total SSL/TLS		36d	**124,315,268**	**56,463,736**	**45.42%**

Table 7.3: Scanned SSL/TLS hosts for our ground truth data set. July, 2015.

ments. Note that archived routing table dumps are readily available. Hence, this sanitizing process can be easily implemented as a post-processing step after the ground truth scan is complete.

In the scope of this thesis, we applied the SSL/TLS filter technique in two subsequent years. Figures 7.5 and 7.6 show respective details on the process of obtaining both ground truth data sets. Note that we actually carried out two consecutive scans in 2014 in order to obtain a set of more stable landmark hosts, e.g. by excluding dynamically connected hosts that did not show up in both scans or presented differing keys. As a matter of fact, we do not draw false conclusions in our filter if keys differ. In 2015, we thus decided to skip this process due to limited effect and increased intrusiveness.

Our first scan targeted HTTPS only, i.e. port 443, and lasted for 18 days. It yielded 38.2 million hosts with an open port, of which 27.2 million were able to carry out a successful SSL/TLS handshake. Common reasons for these disproportions are beyond scope and further studied in [192]. We subsequently removed unstable hosts by means of a second scan, as well as hosts that presented non-unique keys. This precaution eliminates the risk of falsely legitimizing events imposed by default certificates. Such certificates often ship with popular web server software or with SSL/TLS-enabled devices and could thus be presented

by an attacker as well. We finally sanitized our data set to remove hosts that were affected by routing anomalies as discussed above. The resulting ground truth data set comprised a total of 5,356,634 usable landmark hosts.

In 2015, we greatly extended our ground truth scans to a variety of popular SSL/TLS protocols. Table 7.3 shows details on all corresponding scans. In many cases, we scanned for both TLS and STARTTLS (see Subsection 3.2.3 for technical details), and we ommited SSLv3 due to known vulnerabilities. It is instructive to see that the use of TLS greatly varies between application-layer protocols. An open dedicated port does not imply support for TLS per se, which is especially true for STARTTLS hosts. Due to the marginal contributions that our scans of XMPPS and IRCs provided, we have since stopped scanning these protocols.

In total, we tried to open connections to 124,315,268 individual ports. For successful SSL/TLS handshakes, we downloaded the certificate and extracted the public key. As discussed above, we only considered keys that were unique across the whole dataset. We relax this condition where the same key is presented by a single host for multiple protocols. This finally yields a total of 12,800,474 available keys, which were presented by a total of 8,402,023 different hosts. In the further course of this thesis, we will study the added benefit of this multi-protocol approach compared to our proceeding in 2014.

7.2.3 Topology Reasoning

The final filter to legitimize routing anomalies is a topology-based reasoning algorithm. The key idea is that an attacker is unlikely to hijack his own upstream provider. This assumption is based on the fact that the attacker's malicious BGP updates need to propagate via this upstream provider, who could simply counter an attack by filtering them out. As a consequence, we can rule out an attack if the suspected attacker resides in the downstream AS path of his victim.

To identify such benign anomalies, we utilize BGP collectors to extract all AS paths that lead to the affected prefixes and ASes respectively. If we do not find any AS path that contains both the attacker's and the victim's AS, we cannot draw any further conclusions. The same is true for AS paths in which the attacker is located upstream of his victim. In contrast, we can infer a legitimate cause of the anomaly if the attacker is actually located downstream of the victim, i.e. if we find a particular AS path in which the victim's AS precedes the attacker's AS. In this case, we can rule out malicious intent. Note that this filtering technique is implemented and operated by colleagues [164] in conjoint work with Eurécom, France.

Such benign situations might occur, for instance, if smaller organizations obtain Internet connectivity and an IP prefix from a larger carrier. Other reasons can be static routes invisible to BGP, imperfect multihoming setups, or even misconfiguration. Later on, we will see that a significant part of day-to-day routing anomalies is caused by such topological constellations.

7.3 Supplemental Data

With HEAP, we presented a comprehensive filtering framework to reliably identify benign routing anomalies. Although we aim to legitimize the majority of events fed into the system, a small number of inconclusive results is still to be expected. For these remaining alarms, a manual inspection can be worthwhile to assess their root cause.

HEAP readily provides a rich data set to manually assess events that could not be legitimized. Data records extracted from Internet Routing Registries may contain meaningful information about the involved holders of IP prefixes and ASes, e.g. names and descriptions of the resource holders' organizations, as well as contact details and operational policies. Active SSL/TLS measurements further yield a detailed view of the distribution of active hosts in a particular network. Hence, these measurements provide substantial insight into its actual utilization. The analysis of topological data derived from BGP might complement the view by providing details on upstream providers and customers of an affected AS. In addition to these native data sources, manual inspection within HEAP can be supplemented by our previous work on IP geolocation [1] and traffic analysis [2, 4, 5].

7.3.1 IP Geolocation

With our SSL/TLS filter, we identify active hosts in supposedly hijacked networks that can carry out SSL/TLS handshakes, i.e. provide us with cryptographic keys. For alarms that could not be validated with HEAP, we were unable either to confirm that these keys remained unchanged during an event or to carry out a successful handshake in the first place. In any case, our measurement results can be utilized to further assess these alarms by means of fingerprinting.

Table 7.3 indicates that only half of all hosts with an open SSL/TLS port actually carry out a successful SSL/TLS handshake, with an even lower fraction for regular service ports with respect to STARTTLS (refer to Subsection 3.2.3). To get a broader picture of suspicious

routing events, non-SSL/TLS hosts can be further fingerprinted with respect to open ports, TCP options, operating systems, etc. In addition, continuous monitoring can be employed for the duration of an event in order to analyze fingerprint changes during and after its occurence. These changes can indicate a suspicious correlation between the alerted routing event and an unusual change in network utilization. Such monitoring can be carried out at the network level [189] or on a host-by-host basis [185]. For the latter variant, our previous work [1] on IP geolocation can be taken into account in order to detect a geographical relocation of hosts after a routing anomaly disappears. In this work, we present a measurement-based approach to reliably estimate the geographic location of arbitrary Internet hosts. To this end, we established a distributed measurement framework in order to measure latencies towards a given target. Using a spring-based analysis technique, we are able to locate targets with a mean error of just under 100 km.

7.3.2 Netflow Analysis

We have access to archived netflow data of the *Münchner Wissenschaftsnetz (MWN)*—Munich's scientific network—which comprises more than 100,000 end hosts. It is used by researchers, students, and administrative personnel, who generate monthly upstream and downstream traffic volumes of more than 1,900 and 1,100 Terabyte, respectively. The netflow data is collected with the IPFIX [198] protocol.

To further assess suspicious routing anomalies, we can look at this netflow data in order to analyze changes in traffic patterns before, during, and after a suspected attack. Such changes can range from simple outages in the analyzed networks, where outgoing connection attempts from the MWN to these networks are suddenly unanswered, to changes in traffic volume or even to a significant amount of new connections from and to new sets of ports. Furthermore, we consider the MWN large enough to be affected by an attacker conducting large-scale malicious activities like launching massive spam campaigns. We also expect to observe at least some portions of malicious inbound traffic that originates from actually hijacked networks. Hence, manual inspection of suspicious events that pass the HEAP filter chain without conclusive result can greatly benefit from our netflow analysis. To further assist this process, an interactive visualization framework presented in previous work [2] can be utilized.

7.4 Applicability and Ethical Considerations

We acknowledge that our approach depends on external input, thus it is arguably not a full-fledged detection system. We will see, however, that we arrive at remarkable validation results even for a superset of potential alarms. Our approach works best for the assessment of subprefix hijacking alarms (see Section 5.2.1). Attacks that build upon the manipulation of AS paths, like AS hijacking (Section 5.2.2), for instance, can be assessed with HEAP as well. Due to a general lack of initially suspicious input events, however, we exclude this kind of attack from our evaluation. Ordinary prefix hijacking attacks, and MOAS conflicts respectively, impose limitations to our SSL/TLS filter, since we cannot assure that measurements reach a supposedly hijacked network. With the Internet decomposing into two disjoint parts as discussed in Section 5.2.1, our SSL/TLS scans might reach either part, which prevents reliable conclusions. Nevertheless, we can identify such particular input events comprising two origin ASes for a single prefix. For this type of input event, we thus deactivate the SSL/TLS filter. Despite this limitation, the remaining filters are still capable to provide decisive results for MOAS conflicts. Besides, detection systems that could feed our system with corresponding alarms for prefix hijacking attacks are already well-established (see Section 6.3), and thus do not necessarily profit from further assessment.

BGP-based man-in-the-middle attacks (see Section 5.2.2) are especially hard to identify [170]. In an interception scenario, in which an attacker is able to forward our active scans to the victim, the SSL/TLS filter would run the risk of wrongly legitimizing the incident. If marked as such by the input source, we could again deactivate the filter. However, since we cannot determine these cases by our own means yet, we consider sophisticated man-in-the-middle attacks beyond the scope of this approach. In the further course of this thesis, we will present a more reliable technique to detect these incidents such that corresponding alerts can be marked and thus fed into the HEAP filtering scheme while intentionally deactivating the SSL/TLS filter. For hidden takeover attacks (see Subsection 5.2.3), the only applicable filter technique is the IRR-based validation of resource ownership, since we do neither have two origin ASes to reason about nor comparative SSL/TLS measurements for long-term abandoned Internet resources. We consider this kind of attack beyond the scope and derive a complementary detection technique to deal with such cases.

Ethical Considerations For our SSL/TLS filter, we have established a careful proceeding to address ethical concerns. Some of these were already addressed in [199, 197]. In this respect, we inform CERTs, researchers, and blacklist operators before initiating a scan. In addition, we slowly start our scan campaigns, and carefully increase measurement speed.

In any case, we do not exploit the full potential of state-of-the-art scanning tools but deliberately accept longer measurement periods to reduce the stress we put on the measured networks. We further maintain our own blacklist to immediately exclude networks from subsequent scans in case of complaints, which is a conjoint effort with research groups at other academic institutes. It is worth mentioning that we did receive about one hundred complaint emails throughout the course of our ground truth scans, of which almost all were automated alerts raised by generic intrusion detection systems. Note that we strictly answer such emails within 24 hours, thereby describing our intention in much detail. In summary, we consider the overall intrusiveness of the SSL/TLS scans—not least due to our diligent approach—to be tolerable considering the scientific and operational reward.

The outlined analysis of netflow data needs special care. Although we do not analyze payload data, i.e. the actual content of communications, meta data like source and destination IP addresses including corresponding ports already relate to individuals. To protect their privacy, we restrict access to the netflow data to a small group of researchers who have legitimate use and sign corresponding non-disclosure agreements. In addition, we anonymize individual IP addresses in our publications, and do not publish raw data sets in any case.

8 CHAPTER EIGHT
Interception and Path Manipulation

NOTE *This chapter contains prior publication.*

Submitted to **IEEE/ACM Transactions on Networking (TON), 2016.**
Sections 8.1 to 8.3 are based on previous work [9]. Section 8.1 provides additional background on common graph analyses. A technical specification of the construction algorithm for route automata is added to Section 8.2. Individual subsections are slightly rearranged to improve the reading fluency.

SUMMARY In Chapter 5, the author presented a novel concept to study the global routing system in a rigorous model using formal languages. In the following, we introduce *Constructible Automata for Internet Routes (CAIR)*, a framework that translates network paths into incrementally constructible *route automata*. Based on the theoretic framework of finite route languages (see Section 5.1), CAIR fully preserves route diversity, is highly efficient, and well-suited to monitor path changes in real-time. In Section 8.1, we identify shortcomings in state-of-the-art network modeling and outline the need for a new data structure. We formally derive the concept of route automata, consider implementational aspects, and develop an implementable search pattern for interception attacks in Section 8.2. In Section 8.3, we discuss the applicability of our approach with respect to the analysis of Internet routing characteristics in general.

8.1 Introduction and Overview

The Internet interdomain routing system selects those paths from the topologically possible paths that are economically feasible and comply to individual policies. Path vectors in BGP represent the outcome of this hybrid decision process without revealing its underlying rules explicitly. As such, BGP effectively hides most of its operational semantics from observers and successfully withstands the quest for a simple explanatory model [200]. The collection of all locally valid paths is essentially what we can learn at any given observation point. These paths are represented by our *Constructible Automata for Internet Routes (CAIR)*. CAIR offers two key advantages over existing solutions:

a) By preserving policy-related information in its routing model, CAIR can reliably detect interception attacks or route leaks as they violate policies.

b) CAIR is an efficient yet complete representation of the observable inter-domain routing, which opens up the field for new analyses—even in real-time.

We approach CAIR by describing a gap we currently face in network modeling and analysis in connection with common graph models. We then illustrate with some background why a model based on formal languages and automata can actually close this gap.

Network Graphs

Inter-provider connections are traditionally modeled as a graph [201]. BGP peerings that show up in AS paths are considered valid links between nodes (ASes). Network graphs adequately represent connectivity in terms of router links or AS peerings. As a consequence, a huge body of related work exists in the field of graph analysis. Well-understood analysis techniques can be leveraged to derive fundamental graph properties [202, 203, 204, 205]. Such techniques were consequently adopted to model [206, 207, 208] and study [209, 210, 211, 212] various aspects of the Internet topology. Applications on top of network graphs can provide insights on a higher level of abstraction, e.g. finding the economical type of relationship between two ASs [213] or assessing resilience and recovery strategies for link failures [214]. For the analysis of policy-influenced routing, though, such graphs tend to oversimplify the real situation. Realistic routing paths are selected on a per-prefix basis and are often influenced by local routing policies. In particular, network graphs falsely imply transitivity over individual links and thus introduce additional, potentially nonexistent (sub)paths. Therefore, graph models cannot diagnose policy violations such as route leaking or complex anomalies such as interception attacks.

(a) Observed routes

(b) Network graph

Figure 8.1: Common representations of routes to exemplary prefixes P_{1-3} via AS_{1-4}.

Problem Statement

Consider a routing table extract that includes routes to three IP prefixes P_{1-3} originated by the same AS AS_3 as shown in Figure 8.1a. Obviously, a complete list of all paths preserves the observable routing properties but is inefficient to memorize and difficult to analyze. A common way to represent such network topologies reduces data to a graph $\mathcal{G} = (N, L)$ that is a set of nodes N and links L describing connectivity between ASes. For the analysis of policy-based routing, though, such graphs oversimplify real BGP operations. In our example, transit paths to AS_3 differ for individual prefixes. Such prefix-based policies [215] are not reflected in a graph model (compare to Figure 8.1b). Rather, the model implies transitivity, which presumes reachability that has not been observed. It is a modeling artifact.

In particular, graphs incorrectly imply transitivity such that the following assumption generally holds true:

$$AS_1 \rightarrow AS_2 \ , \ AS_2 \rightarrow AS_3 \quad \Rightarrow \quad AS_1 \rightarrow AS_3$$

Note that in our example, transitivity breaks for the prefix P_3, i.e. no route $AS_1 \rightarrow AS_2 \rightarrow AS_3 \rightarrow P_3$ has been observed. CAIR offers a natural solution to this problem. It allows for an accurate representation of observed routing paths and is a highly efficient approach at the same time.

Why Using Finite Automata

In the following, we solve the fundamental transitivity problem in graphs for Internet routes. We remain with the concept of (context-dependent) path vectors and utilize finite route language (see Section 5.1) to model BGP data. Starting from an alphabet of AS numbers and prefixes, we understand AS paths towards prefixes as words of this language. We formally construct so-called *route automata* and show how to put this concept into practical

use. Using incrementally minimized deterministic finite-state automata (DFA), we arrive at a most efficient representation of the path vector space, which outperforms network graphs. Route automata are policy-aware: They represent the full characteristics of the observable routing policies and, at the same time, reduce complexity to real-time compliant processing. With CAIR, we can further derive formal detection patterns for route anomalies that can be applied to real BGP data.

Our CAIR framework serves to construct and minimize route automata providing the following unique features:

Accuracy All BGP paths initially observed are stored in a route automaton without introducing further unobserved information. Consequently, CAIR accurately reflects the observable routing system.

Expressiveness The states of a minimal DFA represent equivalence classes such that two equivalent states exhibit the same (routing) behaviour. CAIR exploits this unique fact to reveal intrinsic routing properties, such as the routing importance of an AS or the location of a hidden attacker.

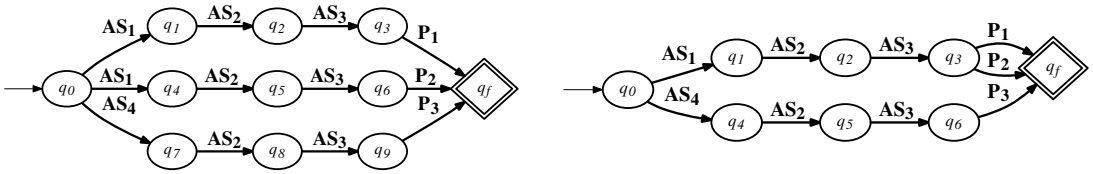
Feasability Well-understood algorithms exist to construct a DFA. Processing of paths equals traversing the automaton. As a consequence, CAIR supports random access of data. It can be deployed both in real-time to continuously monitor BGP as well as in retrospective on archived data sets.

Efficiency Any DFA can be minimized without sacrificing accuracy. The resulting minimal automaton is unique. This is a highly efficient way to represent a finite language in general [216], and consequently a set of network paths we model with CAIR.

Despite this powerful feature set, formal languages have not been applied, yet, in the context of Internet routes. It is worth mentioning that other, less popular data structures could correctly model network paths, e.g. *tries* [217]. We will see later, however, that CAIR naturally outperforms such data structures both in terms of expressiveness and efficiency.

CAIR in a Nutshell

CAIR utilizes a finite route language to construct a *route automaton* based on BGP data such that each transition (edge) is labeled with either an AS number or an IP prefix. CAIR thereby reduces the amount of states by on-the-fly minimization. Traversing edges until the final, accepting state of the automaton results into a complete AS path as included in the initial BGP data set. Having the automaton in place, we search for specific properties. Since



(a) Deterministic automaton (b) Minimal automaton

Figure 8.2: Novel representation of network routes with finite-state automata.

a DFA exists for any given finite language (and vice versa), route automata accurately model the underlying path input data and are thus well-suited to precisely describe its intrinsic routing characteristics.

Toy Model

For our examples in Figure 8.2, corresponding DFA and minimal DFA (MDFA) are shown in Figure 8.2a and Figure 8.2b respectively. In our CAIR framework, we utilize the latter one for route analysis. An important aspect of minimization is that we can observe route diversity based on nonminimizable states. In the minimal automaton, we obtain a sequence of independent states (q_1, q_2, q_3) and (q_4, q_5, q_6) that represents the nonuniform redistribution of prefixes $\{P_1, P_2\}$ and $\{P_3\}$. It is worth noting that CAIR does not intend to improve on the incompleteness of measured BGP data. Instead, we present a novel data model with maximum expressiveness that allows for an accurate representation of *observed* routing paths.

8.2 The *CAIR* Framework

After introducing the concept of *Constructible Automata for Internet Routes (CAIR)* and its practical aspects with respect to automata construction, we formulate an implementable search pattern for interception attacks. For detailed background on formal languages and automata theory we refer to [216].

8.2.1 Route Automata

We define a *route automaton* as a *minimal* deterministic finite-state automaton that accepts any given finite route language (see Section 5.1). In general terms, an automaton represents a state machine that accepts a formal language, i.e. processing one of its words

ends in an accepting state. Let $\mathscr{L} \subset \Sigma_{AS}^* \times \mathscr{P}(\Pi)$ be an FRL representing all routes in the global routing system. Then, we define a route automaton as the 5-tuple

$$M = (Q, \Sigma_{AS} \cup \mathscr{P}(\Pi), \delta, q_0, F)$$

with Q a finite set of states, $q_0 \in Q$ the start state, $F \subset Q$ a set of accepting states, and δ a partial mapping $\delta : Q \times (\Sigma_{AS} \cup \mathscr{P}(\Pi)) \to Q$ denoting transitions. We define the extended transition function δ^* for routes $u\vec{r} \in \mathscr{L}$ as the mapping $\delta^* : Q \times \mathscr{L} \to Q$ such that

$$\delta^*(q, \epsilon) = q$$
$$\delta^*(q, u\vec{r}) = \begin{cases} \delta^*(\delta(q,u), \vec{r}) & \text{if } \delta(q,u) \neq \perp, \\ \perp & \text{otherwise} \end{cases}$$

with $u \in \Sigma$, $\vec{r} \in \Sigma^* \times \mathscr{P}(\Pi)$ a partial route, $q \in Q$, ϵ an empty path, and $\perp \in Q$ a catching state that represents nonexistent routes. We further define $\mathscr{L}(M)$ as the language accepted by the automaton M as

$$\mathscr{L}(M) = \{r \in \mathscr{L} \mid \delta^*(q_0, r) \in F\}.$$

Then, M_P denotes an automaton accepting a set of observed routes $\mathscr{L}_P \subset \mathscr{L}$ from individual observation points $P \subset \Sigma_{AS}$ such that $\mathscr{L}(M_P) = \mathscr{L}_P$.

A path segment $w \in \Sigma_{AS}^*$ with $r = w\vec{r} \in \mathscr{L}$, $|w| < |r|$ is denoted $w \prec r$. We define the longest common path segment w_{lp} in M such that

$$\forall r \in \mathscr{L}(M),\ w \prec r\ :\ |w_{lp}| > |w|,\ \delta^*(q_0, w_{lp}) \neq \perp.$$

The power set $\mathscr{P}(\mathscr{L})$ holds all finite languages that represent (partial) routes. We define the *right language* of a state $q \in Q$ in M as

$$\vec{\mathscr{L}}(q) = \{\vec{r} \in \mathscr{P}(\mathscr{L}) \mid \delta^*(q, \vec{r}) \in F\}.$$

In other words, $\vec{\mathscr{L}} : Q \to \mathscr{P}(\mathscr{L})$ maps the state q to the set of partial routes producible in the automaton M starting at q. Note that the language accepted by M can accordingly be defined by $\mathscr{L}(M) = \vec{\mathscr{L}}(q_0)$.

All states $q \in Q$ that accept the same right language are equivalent and can be merged into a single state. Iteratively applied, this process leads to a minimal automaton, in the following called *route automaton*. This automaton is unique except isomorphisms and requires a minimum number of states among all automata that accept the same language [216]. Note that the number of accepting states in a route automaton is $|F| = 1$, since all routes $r \in \mathscr{L}$ end with an IP prefix, i.e. $\forall q \in Q,\ p \subset \Pi\ :\ \delta(q,p) \in F \cup \{\perp\}$ and $\forall q \in Q,\ w \in \Sigma_{AS}^*\ :\ \delta^*(q,w) \notin F$. In the following, we use q_f to refer to this single accepting state.

8.2.2 Implementational Aspects

For large volumes of input data such as a global set of routes in BGP, the construction of route automata is challenging. In the following, we describe our approach to minimization in CAIR in detail. The most basic algorithms to implement DFA minimization have a complexity of up to $\mathcal{O}(|\Sigma_{AS} \cup \mathscr{P}(\Pi)| \cdot |Q|^2)$, i.e. quadratic complexity in the number of states of a given DFA. Although more efficient algorithms exist [218], the size of our intended input data greatly exceeds that of common use cases in language processing, both in terms of the number of words (i.e. observable routes) and the size of the alphabet (i.e. nodes in the network). For comparison, the English alphabet consists of 26 letters, whereas the technical size of a routing alphabet is $|\Sigma_{AS}| + |\mathscr{P}(Pi)| = 2^{32} + \Sigma_{i=0} 322^i$ for the IPv4 address space of the Internet.

Minimization of Automata

To construct an MDFA efficiently, we adopt a special-purpose algorithm [219] that minimizes a DFA *during* construction without ever holding the full non-minimized automaton in memory. Its memory complexity is $\mathcal{O}(|Q_m|)$, where $|Q_m|$ is the number of states in the *minimized* automaton. The algorithm is capable to randomly add or remove routes, whereas common minimization algorithms need to re-minimize at each change of data. Note that this particular approach expects an acyclic transition function δ^* such that

$$\forall r \in \mathscr{L}(M) \,:\, \nexists\, q \in Q \,:\, \delta^*(q, r) \;=\; q\,.$$

Hence, corresponding automata do not support languages where symbols repeatedly occur within a given string. In the next section, we show that this is not a limitation in our context.

Incremental Construction Algorithm

The procedure to add a route to a given (possibly empty) route automaton is inspired by [219] and comprises three major steps. First, for each route the automaton is traversed along the longest common path segment, thereby ensuring that no invalid paths are introduced. Second, states that accept the remaining part of the route are newly created. The final step is to carry out an on-the-fly minimization while traversing the automaton in backward direction. In the following, we present an in-depth description of the individual steps. Note that during construction, we utilize a register of states Q_R, which is implemented as a hash table, i.e. an associative array that maps keys to values. Keys thereby represent unique

right languages $\vec{\mathscr{L}}(q)$ for individual states $q \in Q$. This provides an efficient way to search for equivalent states, i.e. for states that have identical right languages.

Algorithm: Incremental construction of route automata (based on [219]).

Step 1: *Common Path Traversal* To add a new route $r \in \mathscr{L}$ to M, we start at q_0 and traverse the automaton along a (possibly empty) sequence of existing states for the longest common path segment w_{lp} in M. For so-called *confluence states* that have more than one incoming transition, we need to create a new state with identical outgoing transitions and link it to the last state traversed. This cloning process prevents inadvertently adding false paths. Note that this is inherently the case with network graphs.

Step 2: *Remaining Route Insertion* After finding a specific state q that represents the longest common path segment w_{lp}, it is removed from the state register Q_R since its right language is about to change. If the whole route is accepted in the first step, i.e. if $|w_{lp}| = |r|$, it is already contained in the automaton. Otherwise, we add new states for the remaining part of the route and link them with corresponding transitions. The last created state is marked as an accepting state.

Step 3: *On-the-fly Minimization* Finally, we traverse the automaton in backward direction along the sequence of states that represent the newly inserted route r. For each state q, we search our register Q_R for an equivalent state $\tilde{q}$ that already exists in M. If found, we discard q and link its preceding state to $\tilde{q}$. Otherwise, q is unique across all states of M and needs to be added to the register Q_R.

Note that by traversing the automaton backwards in step 3, the comparison of right languages to identify existing equivalent states is reduced to a comparison of outgoing transitions, since recursive application would only yield the results of already compared states. Hence, it is sufficient for keys in Q_R to represent the outgoing transitions of individual states instead of their full right language, which greatly reduces computational complexity. The Algorithm provides a full specification of the construction algorithm in pseudo code.

Interestingly, by leaving out step 3, we obtain *trie* data structures (see Section 8.1). Beside redirection of transitions, this step only removes states from memory. As a consequence, route automata strictly outperform tries with respect to memory requirements and expressiveness due to the absence of redundant information.

To summarize, our construction algorithm allows to randomly add or remove routes while still ensuring minimality. CAIR is thus particularly well-suited for continuous monitoring of routing changes in BGP. We can also create and archive individual automata that

Algorithm: Incremental construction of route automata (based on [219]).

Input: set of loop-free routes $\mathscr{R} \subset \Sigma_{AS}^* \times \mathscr{P}(\Pi)$ over the alphabet $\Sigma_{AS} \cup \mathscr{P}(\Pi)$
Output: route automaton $M = (Q_R, \Sigma_{AS} \cup \mathscr{P}(\Pi), \delta, q_0, q_f)$ accepting all routes $r \in \mathscr{R}$

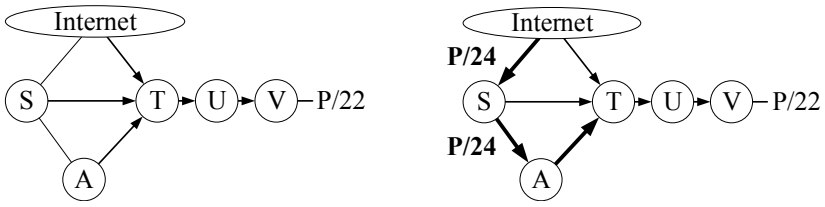
$q_0 \leftarrow$ new_state() `// start state`
$q_f \leftarrow$ new_state() `// accepting state`
$Q_R \leftarrow \{\}$ `// state register`
$\delta \leftarrow \{\}$ `// transition function`

for *r in* $\mathscr{R}$ **do**
    $V_Q = [q_0]$ `// state vector`
    $q_{lp} \leftarrow q_0$

    `// Step 1: Common Path Traversal`
    for *i in* 0 .. *len*$(w_{lp})-1$ **do**
        $q_i \leftarrow \delta(q_{lp}, w[i])$
        `// Check for confluence states`
        if $\exists \bar{q} \in Q_R,\ \bar{q} \neq q_{lp} : \delta(\bar{q}, u) = q_i,\ u \in \Sigma_{AS}$ **then**
            $q_i \leftarrow$ clone_state(q_i)
            $\delta \leftarrow \delta \cup \{((q_{lp}, w[i]), q_i)\}$
        $V_Q \leftarrow V_Q + q_i$
        $q_{lp} \leftarrow q_i$
    $Q_R = Q_R \setminus \{q_{lp}\}$

    `// Step 2: Remaining Route Insertion`
    for *j in len*(w_{lp}) .. *len*$(w)-1$ **do**
        $q_j \leftarrow$ new_state()
        $\delta \leftarrow \delta \cup \{((q_{lp}, w[j]), q_j)\}$
        $V_Q \leftarrow V_Q + q_j$
        $q_{lp} \leftarrow q_j$

    `// Step 3: On-the-fly Minimization`
    for *k in len*$(V_Q)-1$ .. 1 **do**
        $q_k = V_Q[k]$
        `// Check for equivalent states`
        if $\exists \tilde{q} \in Q_R,\ \tilde{q} \neq q_k : \tilde{\mathscr{L}}(\tilde{q}) = \tilde{\mathscr{L}}(q_k)$ **then**
            $\delta \leftarrow \delta \setminus \{((V_Q[k-1], w[k-1]), q_k)\}$
            $\delta \leftarrow \delta \cup \{((V_Q[k-1], w[k-1]), \tilde{q})\}$
        else
            $Q_R = Q_R \cup \{q_k\}$



(a) Regular traffic flow (b) Interception

Figure 8.3: Attacker A intercepts subprefix $P/24$.

represent the global routing system at specific points in time. We will make use of this feature in our evaluation.

Solving Loops

CAIR accounts for the construction of any set of network routes consisting of nodes and links [201] as long as the input is cycle-free. Even though most routing protocols such as BGP provide built-in support for loop prevention, loops may be included in routing data nonetheless. Network operators, for example, may decide to influence route selection by adding their own AS number multiple times to the AS path (*AS path prepending*). Still, this is not a problem for our approach: A simple way to model subsequent occurrences of a particular AS o in M is to extend Σ_{AS} by multiple instances $o_i \in \Sigma_{AS}$ with $i \in \{1,2,\ldots\}$.

8.2.3 A Search Pattern for Interception Attacks

We present a methodology to search route automata for anomalies that emerge from interception attacks. Before we explain the details, we briefly illustrate the intuition behind our approach (for background on interception attacks in general see Section 5.2.2).

Interception attacks are characterized by a subtle injection of illegitimate prefix routes to redirect traffic destined for a victim to the attacker To search for such events we need patterns and data models. A crucial factor for the attacker to succeed is to sustain global connectivity by relaying all communication back to the victim. Implementing such an attack in the Internet is feasible and has been demonstrated in practice [171, 167, 31]. A malicious AS needs to be connected to the Internet via at least two upstream ISPs. The first ISP is used to attract traffic by advertising the victim's address space or a part thereof, i.e. to carry out a hijacking attack. The second ISP serves to preserve a stable backhaul path from the attacker to the victim. To this end, the attacker includes all ASes between himself and his

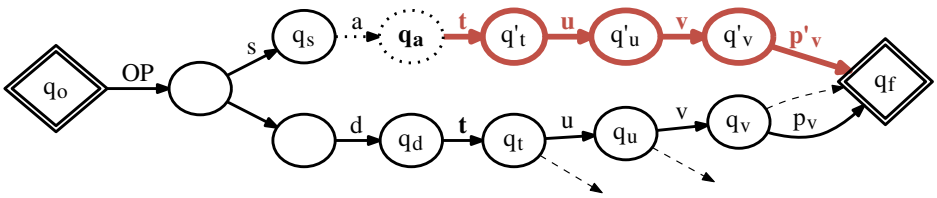


Figure 8.4: Route automaton for an interception attack.

victim in the malicious BGP announcement. As a consequence, all of these ASes discard the announcement due to loop prevention, and a stable link from the attacker to the victim's AS remains intact. Figure 8.3 illustrates the corresponding change in traffic forwarding.

Our key observation to derive a search pattern for interception attacks is that an arbitrary ISP T normally redistributes *all* routes of a *distant* AS V along the same ways [220, 215] (Figure 8.3a). From a BGP policy point of view, however, the attack induces a specific policy for the redirected prefix. In our example, S forwards traffic to V via the malicious AS A only for the prefix under attack, but reaches all other prefixes of the victim directly via T (Figure 8.3b). To diagnose the difference, data models that express route diversity are needed. Common graph models lack this capability. It is worth mentioning that the attacker can easily hide his own identity by not adding his AS number to the `AS_PATH` attribute of forged BGP messages, as it is common practice with route servers [128], for instance.

Intuition behind the Detection Scheme

Figure 8.4 shows an interception attack on a (sub)prefix $p'_v \subseteq p_v \subset \Pi$ of the victim $v \in \Sigma_{AS}$, in which the attacker $a \in \Sigma_{AS}$ fabricates an *artificial path segment* $w_v^a = tuv \in \Sigma_{AS}^*$ over $t, u \in \Sigma_{AS}$ as described in Section 5.2.2. Recall our assumption that a distant AS t that is not a direct upstream of v neutrally redistributes the routes of v. Hence, the right language (Subsection 8.2.1) of the automaton state q_t should represent all transit routes over t (indicated by dashed lines in Figure 8.4). In an interception scenario, however, the attacker seemingly changes the routing policy of t: It appears that t now forwards announcements of p_v and p'_v differently, i.e. selectively to $a, d \in \Sigma_{AS}$. As a result of this diversity, our route automaton holds separate states $\{q_v, q'_v\}$, $\{q_u, q'_u\}$, and $\{q_t, q'_t\}$. For the sake of completeness, Figure 8.5 shows a similar graph data structure for the attack depicted in Figure 8.4, which fails to correctly model the interception scenario. None of the vertices can be considered suspicious as such, as is the case for the subprefix announcement and the forged link between a and t.

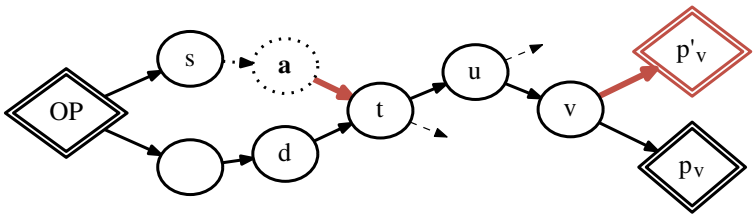


Figure 8.5: Network graph failing to recognize interception attacks.

In the remainder, we anticipate that the attacker hides his true identity by not adding his AS number a to the `AS_PATH` attribute of forged BGP messages (remove dotted elements in Figures 8.4 and 8.5). In this case, we rename q_s to q_a with $s \in \Sigma_{AS}$ being the attacker's second upstream provider. We further assume that the attacker seeks to globally attract traffic to v by means of a strict subprefix hijacking, i.e. $p'_v \subset p_v \subset \Pi,\ p'_v \neq p_v$ holds (see Section 5.2.1).

Translation to Route Automata

In order to detect interception attacks, we need to search for a partial route that a) is used in a single routing context only (*artificiality*), b) is in contradiction to existing announcements (*nonuniformity*), and c) leads to a subprefix of the benign routes (*interception alert*). Within our route automata, these conditions can be easily expressed and implemented as follows.

Artificiality An artificial path segment $w_v^a \in \Sigma_{AS}^*$ is given by a sequence of (at least) four states $\delta^*(q_a, tw_v^a) = q'_v$ with a single outgoing transition each, i.e.

$$\forall tw \in \Sigma_{AS}^*,\ w \neq w_v^a :\ \delta^*(q_a, tw) = \bot\,.$$

Nonuniformity We further search for an additional path segment w_v^d that is in contradiction with w_v^a such that

$$\exists q_d \in Q :\ \delta^*(q_d, w_v^d) = q_v\,.$$

Interception alert We verify if the offending state q'_v represents a strict subprefix hijacking of q_v, i.e. if the condition

$$\exists p_v, p'_v \subset \Pi,\ p'_v \subset p_v :\ \delta(q_v, p_v) = \delta(q'_v, p'_v) = q_f$$

holds true. We then raise an interception alert and report the prefixes p_v and the artificial AS path segment w_v^a for further investigation.

Discrimination of the Attacker

With our route automaton, we are able to pinpoint the attacker's location in the Internet topology. If the attacker adds his AS number a to his BGP updates, we can directly observe a transition labeled with a that points to the offending state q_a. Under the assumption that the attacker is taking precautionary measures to hide his AS (refer to Section 5.2.2), the incoming and outgoing transitions of q_a are labeled with s and t respectively. In this case, we can isolate the attacker to be a customer of both ISPs s and t, which leaves us with a small number of possible ASes to be manually scrutinized.

8.3 Applicability and Ethical Considerations

In this chapter, we presented a practical implementation of our formal routing model as developed in Section 5.1 to represent network paths. In general, such paths can be extracted from technical documentation, collected via active measurements, or observed by passively monitoring routing tables. IP-level paths, for instance, could be extracted from network plans or router configurations, but can also be the result of traceroute path measurements. These paths could be further abstracted by looking up AS numbers in the WHOIS system (see Subsection 3.3.1). A more common way to derive such AS-level topologies is to evaluate routing table exports from BGP routers. Our routing model, and the route automata respectively, account for these different levels of abstraction. As a matter of fact, any set of network paths consisting of nodes and links can be used to construct a route automaton.

As discussed earlier, our construction alogrithm can only be applied to cycle-free input data. With respect to our routing model, however, this circumstance is of minor importance. Routing protocols naturally provide built-in support for loop prevention. Hence, without loss of expressiveness, we can accordingly limit the input for the construction of route automata to loop-free network paths. For the analysis of BGP routing, and AS path prepending respectively, a specific AS number can be observed multiple times consecutively in a single path. A simple way to model such self-referencing loops is to add multiple AS instances as discussed in Subsection 8.2.2.

With an analysis of AS paths as included in BGP routing tables, we derived a methodology to detect interception attacks. The key idea behind this approach is that an arbitrary ISP has no incentive to divert particular prefix announcements of topologically distant ASes. It is still possible, though, that such ASes correspond to a larger institution that holds multiple

AS numbers and applies complex inter-AS routing policies. In this case, our detection technique might raise false positive alerts. With the help of the WHOIS system, however, we can manually inspect suspicious cases and rule out those with legitimate cause.

CAIR does not depend on the selection of a specific set of observation points. If an intercepted subprefix hijacking occurs, the update will be visible in the default-free zone, similar to the victim's less specific prefixes. However, our chances to detect the hijacking of prefixes depend on the observation of competing routes. For this reason, we suggest to utilize a well-connected BGP collector such as the one we used in our evaluation. Furthermore, CAIR does not require training of the data set. Routing data (either measured or artificially constructed) is transformed into a route automaton and reasoning is defined by search patterns, which are based on common operational practice.

Our detection technique is not limited to subprefix hijacking, which could even be protected against by RPKI (see Section 4.4). If an attacker intercepts an already announced prefix as such, CAIR would detect this incident. The interception leads to a different routing policy, i.e. the attacked prefix is announced differently compared to all other prefixes of the origin AS. At the moment, CAIR cannot detect incidents in which *all* prefixes of an origin AS are simultaneously intercepted. The current search pattern is based on route diversity per AS. If all prefixes of an AS are comprehensively intercepted as announced by the victim, we would de facto observe a uniform redistribution. However, CAIR stores all routing paths and allows for easy integration of additional search patterns.

Ethical Considerations From an ethical point of view, no restrictions apply to our approach due to the analysis of publicly available routing data. However, we adhere to responsible disclosure of detected attacks in order to avoid creating public tensions between the affected parties.

9 CHAPTER NINE
Abandoned Internet Resources

NOTE *This chapter contains prior publication.*

Published in **Proceedings of the International Workshop on Traffic Monitoring and Analysis (TMA), 2015.**
Sections 9.1 to 9.3 are based on previous work [6]. A discussion of ethical concerns is added to Section 9.3.

Published in **ACM SIGCOMM Computer Communication Review (CCR), 2013.**
Section 9.2 incorporates previous work [4] discussing the draft for an early warning system. It is extended to take into account recent administrative activities of Internet resources.

SUMMARY Almost all discussions in literature assume that hijacking incidents are enabled by the lack of security mechanisms in the interdomain routing protocol BGP. In this chapter, we focus on threats that emerge from abandoned Internet resources, where a lack of ownership validation from an administrative point of view enables an attacker to take over resources, thereby effectively hiding his own identity. In Section 9.1, we show that such hidden takeover attacks are feasible with very little effort. To further study the risk potential of this kind of attack, we introduce the *Prefix Hijacking Early Warning (PHEW)* system. We present PHEW together with techniques to assess the utilization of resources in Section 9.2 and show that abandoned resources are a real phenomenon. We further address the applicability of PHEW as well as related ethical concerns in Section 9.3.

9.1 Introduction and Overview

A particular threat to the Internet architecture emerges from abandoned Internet resources like IP address blocks and AS numbers. When DNS names expire, attackers gain the opportunity to take resource ownership by re-registering domain names that are referenced by corresponding RIR database objects. We draw on several data sources to identify such abandoned resources. To this end, we utilize public resource databases, extensive WHOIS queries for administrative DNS information, and a large set of archived BGP data. Our intention is to identify vulnerable Internet resources in order to inform resource holders to deploy countermeasures in time. Naturally, this knowledge could be abused by attackers. We address this issue with a strategy for responsible disclosure of our findings.

Risk Potential

Conventional attacks on BGP are based on a lack of origin validation, which allows an attacker to originate arbitrary prefixes or subprefixes from his own AS, possibly supported by the manipulation of AS paths. With hidden takeover attacks (refer to Subsection 5.2.3), we introduced a new type of attack that accounts for hijacking of both Internet resources and corresponding ownership information stored in Regional Internet Registry (RIR) databases. This kind of attack is more attractive than conventional hijacking, since the attacker can act in full anonymity. In addition, such attacks are significantly harder to disclose. Consequently, current detection techniques are not qualified to deal with these attacks (refer to Section 6.3 and Table 6.1).

So far, hijacking attacks were characterized by an attacker altering the global routing table without consent of the victim. In addition, tailored attacks can be derived to actually impersonate a victim at RIR level, a fact that has received little attention so far. In a history of more than three decades, a vast number of Internet resources has been handed out to numerous users under varying assignment policies. Some ASes or prefixes have never been actively used in the interdomain routing, others changed or lost their original purpose when companies merged or vanished. It is not surprising that some Internet resources became abandoned, i.e. that resource holders ceased to use and maintain their resources.

A formless *letter of authorization* sent from a customer is often accepted by upstream ISPs as a legitimation to redistribute the customer's prefix announcements. To check authenticity, RIR-operated databases can be queried in order to validate the contact details of a resource holder, thereby assuming that corresponding resource objects cannot be modi-

fied without valid access credentials. Thus, an attacker can successfully deceive an ISP if he is able to demonstrate control over his victim's resource objects. In many cases, this is equivalent to the ability of sending emails from the documented address in the RIR database. This can be arranged by convincing a RIR of recent changes in responsibility for the resources in question (e.g. with forged papers of a company acquisition), by exploiting flaws in the RIR database software, or by taking over the victim's DNS domain.

Preconditions for an Attack

The following preconditions have to be met in order to enable a hidden takeover of Internet resources: (a) Internet resources are evidentially abandoned and (b) the original resource holder can be impersonated. If an organisation goes out of business inadvertently, these conditions are naturally met. As a first consequence, the organisation ceases to use and maintain its resources. If this situation lasts over a longer period of time, the organisation's domain name(s) are going to expire. Since day-to-day business lies idle, re-registration of the victim's domain names and thus impersonation becomes practicable for an attacker. At that moment, upstream connectivity can be arranged on behalf of the victim, since face-to-face communication is not required in general. Routers can be sent via postal service, or even be rented on a virtualized basis. Details on BGP and network configuration are usually exchanged via email or IRC, and payment can be arranged anonymously by bank deposits or other suitable payment instruments. Without revealing any evidence about his real identity, the attacker is able to stealthily hijack and deploy the abandoned resources.

9.2 The *PHEW* Framework

We present the *Prefix Hijacking Early Warning (PHEW)* system to evaluate the risk potential of hidden takeover attacks (see Subsection 5.2.3 for details). PHEW is designed to identify vulnerable resources and to provide early warnings for corresponding attacks. An abandoned Internet resource is attractive for an attacker for two reasons. First, the resource is assigned to an organisation for operational use and thus represents a valid resource in the Internet routing system. Second, hijacking such resources enables an attacker to stealthily operate on behalf of his victim.

An attacker can possibly claim ownership of abandoned resources by merely taking control of his victim's contact address, which is documented by corresponding RIR database objects (refer to Section 3.3.1). Consequently, we identify abandoned Internet resources by

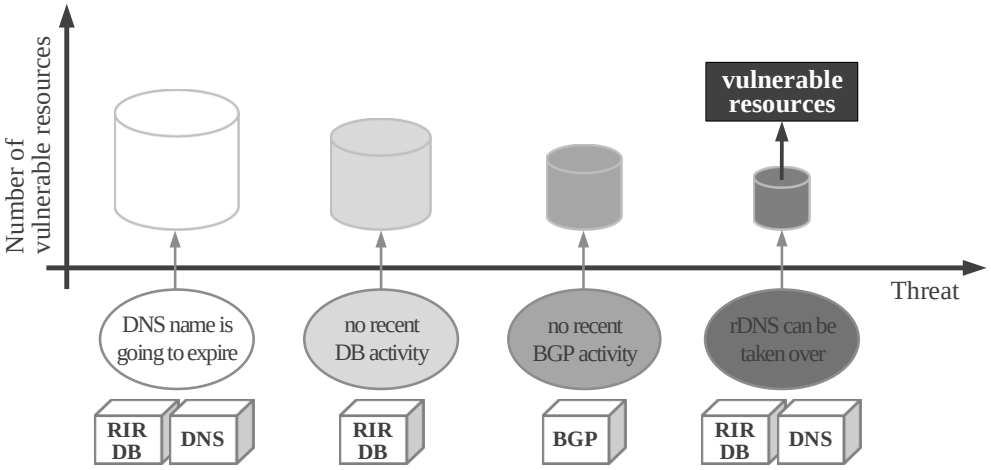


Figure 9.1: The Prefix Hijacking Early Warning System (PHEW).

searching RIR databases for objects that reference email addresses with *expired DNS names*. Control over these addresses can be simply gained by an attacker by re-registering the according domain name. Note that expired domain names referenced in RIR databases can also be the result of typing errors or a recent change in contact details not yet reflected in the database, while the resources themselves might still be in operational use. Hence, it is important to take into account recent database activities for individual resource holders and to correlate this information with activity measures from an operational point of view. To this end, we utilize archived BGP update messages in order to assess the historical deployment of a given resource.

We further take into account the ability to pass so-called Forward-Confirmed Reverse DNS (FCrDNS) checks with hijacked networks (refer to Subsection 3.3.2). Such checks are, among others, widely deployed with respect to spam detection and mitigation. Control over reverse DNS can thus be beneficial for an attacker to support abusive actions. Authority over reverse DNS is delegated by RIRs to name servers of prefix holders. Redelegation requires access to the corresponding RIR database, which we do not assume for an attacker. If the delegation, however, points to a server name that corresponds to an expired DNS domain or if its IP address is located within a vulnerable prefix, control over reverse DNS, and the ability to pass FCrDNS checks respectively, is gained as a side-effect of an attack. Note that DNSSEC [42] effectively protects against this threat.

Figure 9.1 shows our general approach to identify vulnerable Internet resources. The PHEW system is based on the evaluation of multiple data sources. First, DNS domains are

extracted from email addresses provided with RIR database snapshots where available. For these domain names, the expiry dates are obtained. Resources with an expiring domain in the near future are subject to closer investigation. The threat level escalates for resources with no recent activity both in terms of RIR database updates and BGP announcements. To this end, we integrate publicly available BGP data into our system. We further raise the threat level for vulnerable reverse DNS delegations, i.e. where DNSSEC is not deployed and the name servers can be captured. Note that information on reverse DNS delegation is provided with the RIR databases as well.

Threat levels will change over time and have to be re-assessed on a periodical basis. Holders of resources that meet all criteria can be readily informed about an imminent threat, since corresponding email addresses are already obtained in the first step. In addition to contacting vulnerable resource holders, resources with high threat levels can be subsequently monitored with respect to DNS re-registration, suspicious BGP activities, and appearance on blacklists until a re-evaluation yields lowered threat levels.

The analysis steps in the PHEW system can be easily extended by additional steps that provide a further assessment of vulnerable Internet resources. In order to establish an activity metric with respect to database updates and BGP operation, reasonable thresholds have to be chosen. To this end, we identify and study the landscape of abandoned Internet resources. The following analysis is based on archived RIPE database snapshots over 2.5 years (23 February, 2012 till 9 July, 2014). Our results are representative for the European service region only, but similar analyses can be done with little effort for other service regions, too. Note that we utilized a graph database for our analysis similar to the approach described in Subsection 7.2.1.

9.2.1 Resource Candidates from RIR Databases

RIPE provides publicly available database snapshots on a daily basis with most of its individual-related data removed due to privacy concerns. Some object attributes, however, remain unanonymized, a fact we can exploit to extract DNS names. We further assess the authentication mechanism for the RIPE database, and identify groups of resource objects that are maintained by a single authority. Our assumption here is that if we learn that an organization lies idle, *all* resources under its control will be vulnerable.

Object type	Frequency	DNS references	
INETNUM	3,876,883	1,350,537	(34.84%)
DOMAIN	658,689	97,557	(14.81%)
ROUTE	237,370	50,300	(21.19%)
INET6NUM	231,355	8,717	(3.77%)
ORGANISATION	82,512	0	(0.00%)
MNTNER	48,802	0	(0.00%)
AUT-NUM	27,683	6,838	(24.70%)
ROLE	20,684	14,430	(69.76%)
AS-SET	13,655	2,500	(18.31%)
ROUTE6	9,660	723	(7.48%)
IRT	321	162	(50.47%)
Total	**5,239,201**	**1,531,764**	**(29.24%)**

Table 9.1: RIPE database objects and references to DNS names. July, 2014.

Available Data Objects

In Subsection 7.2.1, we presented a detailed analysis on the number of objects and relations in the RIPE database. These objects can be updated from the web or via email. Many objects hold an email address in the `notify` field to which corresponding update notifications are sent. Despite anonymization, we found that these `notify` fields are preserved in the publicly available database snapshots, which is also the case for `abuse-mailbox` attributes. To extract DNS names, we parse these email addresses where applicable.

Table 9.1 shows the distribution of stored objects by type together with the number of DNS names we were able to extract. Although we found more than 1.5 million references to DNS names, the total number of *distinct* names is only 21,061. This implies that, on average, more than 72 objects reference the same DNS name. The overall fraction of objects that reference a domain name is 29.24%, which is surprisingly high since the database snapshots are considered to be anonymized. In reality, however, this fraction is possibly close to 100%. As a consequence, the following analysis underestimates the situation of abandoned resources, and can only provide a lower bound.

Relevant Internet resources are given by `INETNUM` and `AUT-NUM` objects, which represent blocks of IP addresses and unique AS numbers respectively. Exemplary database objects are provided in Section 3.3.1 (Figure 3.1), further details on the RIPE database data model and update procedures are available at [221]. It is worth mentioning that an attacker neither

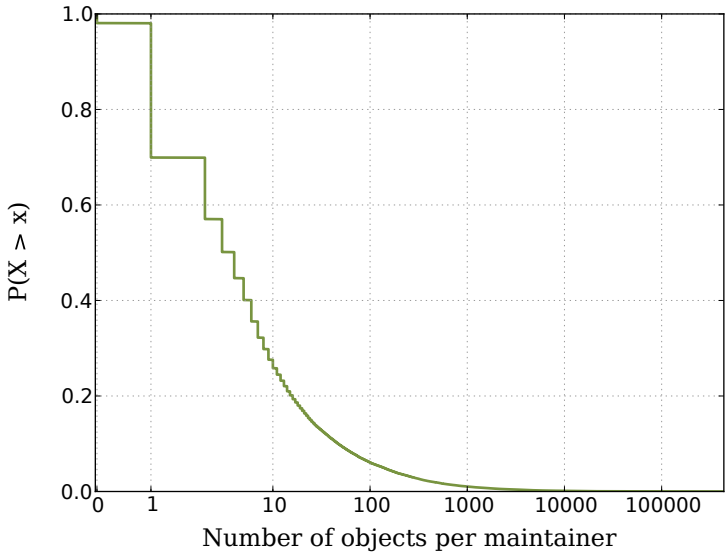


Figure 9.2: RIPE database objects grouped by common maintainer objects (CCDF).

needs authenticated access to the database nor does he need to change the database objects themselves in order to take ownership of a resource. Instead, it is sufficient to identify and capture a valid contact address. In the following, we assume that the aforementioned *notification* and *abuse-mailbox* addresses generally belong to the same DNS domain as of the technical contact of a resource object. Detailed analysis is subject to future work. In the following, we disregard groups of associated objects that reference more than a single DNS domain as a precaution.

Grouping Objects by Maintainer

The RIPE database is mostly maintained by resource holders themselves. As discussed in Subsection 7.2.1, its security model is based on references to so-called `MNTNER` (maintainer) objects, which grant update and delete privileges to the person holding a `MNTNER` object's password. This security model allows us to infer objects under control of the same authority by grouping objects with references to a common `MNTNER` object. We use these so-called *maintainer groups* to estimate the impact of an attack for individual authorities. On average, we observed nearly 110 such references to `MNTNER` objects in July 2014, with a maximum of up to 436,558 references[1]. The distribution of the number of objects per maintainer group is presented in Figure 9.2.

[1]The meta information refers to *Interbusiness Network Administration Staff* of Telecom Italia.

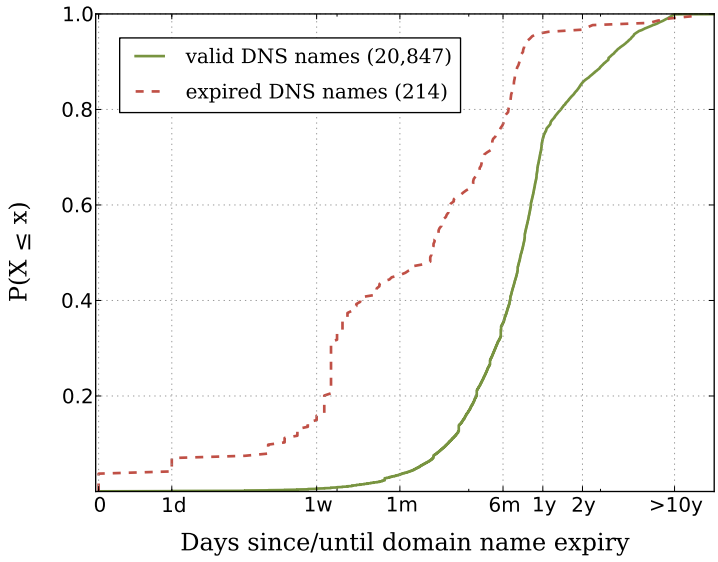


Figure 9.3: Expiry of DNS names referenced in the RIPE database (CDF).

For each of these maintainer groups, we obtain the set of unique DNS names referenced by all objects of a group. To unambiguously identify maintainer groups with expired domain names, we merge disjoint groups that reference the same DNS domain and discard groups with references to more than one DNS name. From an initial amount of 48,802 maintainer groups, we discard (a) 937 groups of zero size, i.e. unreferenced `MNTNER` objects, (b) 31,586 groups without any domain name references, and (c) 4,990 groups with multiple references. The remaining 11,289 groups can be merged to 8,441 groups by identical DNS names. We further discard groups that do not include any hijackable resources, i.e. neither `INETNUM` nor `AUT-NUM` objects, which finally leaves us with 7,907 object groups.

Note again that the number of these groups is a lower bound: An attacker could identify even more groups with access to unanonymized RIPE data, for instance by utilizing online web queries or the WHOIS system. As discussed above, each of these groups is maintained by a single authority represented by the according `MNTNER` object. If a group's DNS name expires, we consider all of the affected resources to be a valuable target for an attack.

Lifetime of Domain Names

We use the WHOIS system (refer to Subsection 3.3.1) to query expiry dates for all extracted DNS names as discussed above. To this end, we parse the registration data provided by NICs. The distribution of the resulting expiry dates is shown in Figure 9.3. On July 9, 2014,

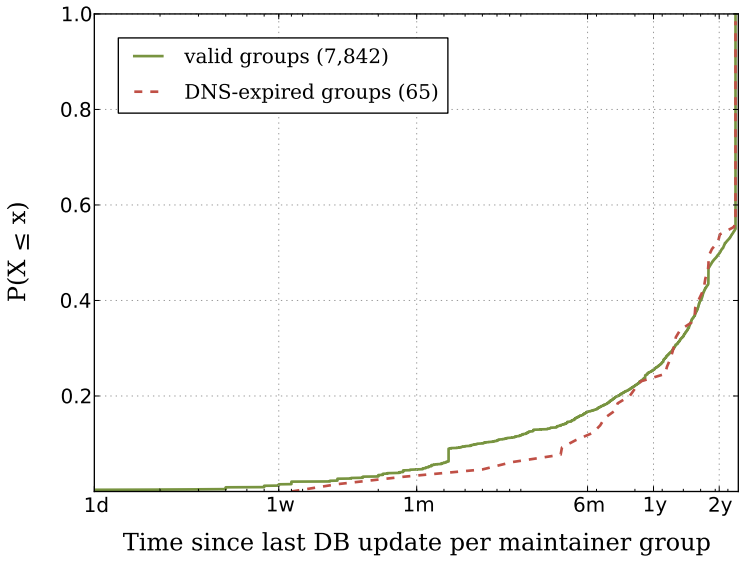


Figure 9.4: RIPE database updates observed by maintainer group (CDF).

214 domain names had expired. Another 121 domain names were to expire within the week, given that the owners would miss to renew their contracts.

The most frequent Top Level Domains (TLDs) we observed are `.com` (27.9%), `.ru` (21.5%), and `.net` (13.0%), while the most frequent *expired* TLDs comprised `.ru` (20.1%), `.it` (16.4%), and `.com` (9.81%). The longest valid domains are registered until 2108 and mostly represent governmental institutions. The longest expired domain is unregistered for nearly 14 years. With respect to the maintainer groups derived above, a total of 65 groups that reference expired DNS names remains. These groups thus suggest themselves to closer investigation.

9.2.2 Refinement by Activity Measures

To confirm that a set of resources is also abandoned from an operational point of view, we leverage complementary data. We start with domain names that expired, which is a strong yet inconclusive indication for fading resources. We gain further evidence by considering only resources that are neither changed in the RIPE database nor advertised in BGP recently. Including both administrative (DNS, RIPE) and operational (BGP) measures finally gives us a comprehensive picture on the actual utilization of resources.

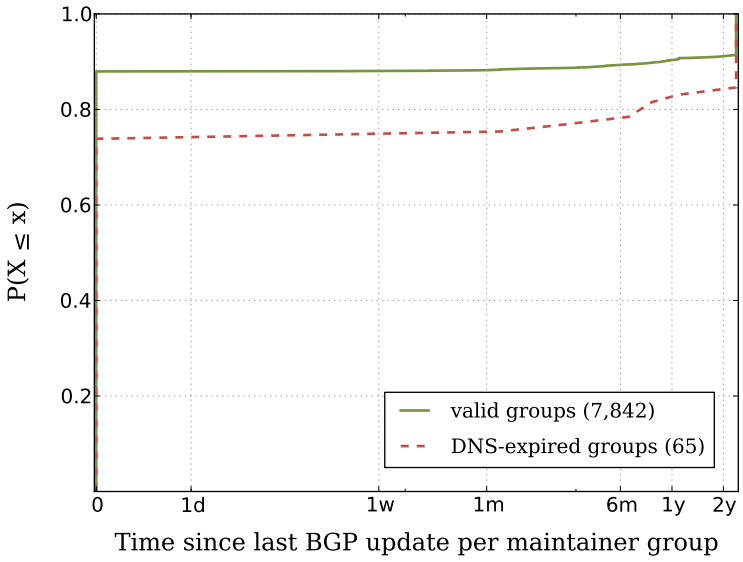


Figure 9.5: BGP activity observed by maintainer group (CDF).

RIPE Database Updates

For each of the 7,907 maintainer groups identified above—divided into 7,842 valid groups and 65 with expired DNS names—we extract the minimal time since the last change for any of its corresponding database objects. Note that we filter out automated bulk updates that affected *all* objects of a certain type[2]. Figure 9.4 shows the minimal time since the last database update for any resource in groups with valid and with expired domain names. Groups that received their last update longer than 9 months ago, i.e. 80% of our groups, show a similar distribution of update times irrespective of the expiry state of their domain names. The remaining 20% of groups, however, differ strikingly in this respect. We learn that expired groups indeed receive fewer updates than valid groups. Note that we do not assume inactivity in absence of such database updates.

Activity in BGP

To further confirm inactivity, we correlate database updates with activities in the global routing system. To this end, we analyze all BGP update messages from the RouteViews Oregon [222] archive for the respective time frame. This data set comprises 83,255 files with 18.4 billion announcements and 1.04 billion withdraw messages for resources assigned by

[2]For instance, RIPE added a new `status` attribute to all `AUT-NUM` objects on May 27, 2014.

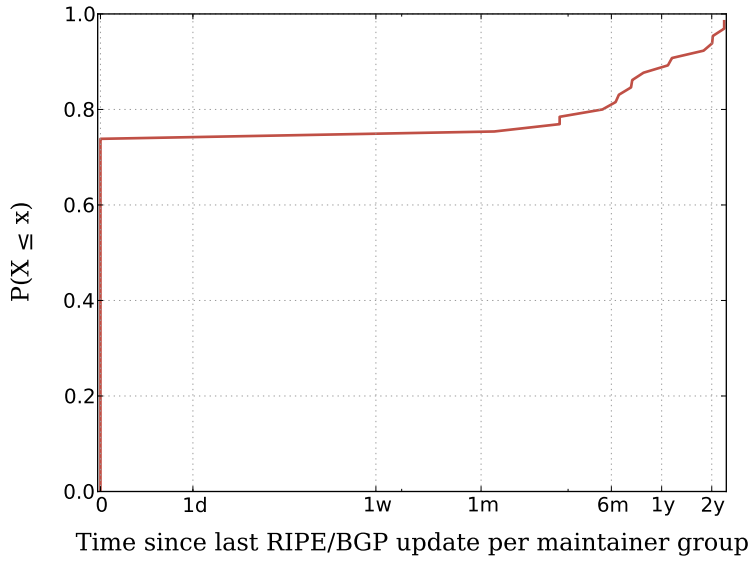


Figure 9.6: Combined activity metric for DNS-expired maintainer groups (CDF).

RIPE. Given this data, we are able to extract two indicators: (1) the time since an *IP prefix* was last visible from the RouteViews monitor and (2) the time since the last deployment of a RIPE-registered *AS number* by looking at AS paths in the announcements. Figure 9.5 shows the distribution of last activities in BGP for our maintainer groups with respect to any of the aforementioned Internet resources. Nearly 90% of resources in valid groups are visible in BGP at the moment. Surprisingly, most of the remaining 10% of groups did not show any activity at all during the last 2.5 years. While less than 75% of the DNS-expired resources are present in today's routing table, this percentage increases to about 85% when looking more than six months into the past. Hence, a significant fraction of these resources showed activity before, and does not show any activity today. This observation clearly differs from the steady situation observed for the valid resource groups.

These findings confirm our assumption that the absence of updates in the RIPE database does not necessarily imply that resources lie idle from an operational perspective. Up to 85% of the DNS-expired resources were active in BGP within the last 2.5 years, but only 55% of them received an update in the RIPE database during this time frame (refer to Figure 9.4).

Combined Activity Measures

We combine both activity measures presented above, and plot the respective minimum in Figure 9.6. This yields our final activity measure, which splits the 65 expired maintainer

groups into two disjoint sets: 52 cases were active within the last 3 months, the remaining 13 cases did not show any activity for more than 6 months. We consider these cases to be effectively abandoned, and consequently choose an inactivity threshold of 6 months for the minimal time since the last database or BGP activitiy. This threshold establishes a basis for continuously monitoring vulnerable Internet resources with our PHEW system.

9.3 Applicability and Ethical Considerations

With the above analysis, and with our PHEW system respectively, we are currently bound to the RIPE service region. This is no inherent limitation but ongoing work, i.e. our technique can be easily extended to the other continents as well. In addition, RIPE provides database snapshots that are anonymized for the largest part. For about 70% of all objects in the database, we are unable to extract corresponding domain names. Our analysis can thus only yield a lower bound for the number of abandoned resources. Further, we disregard resource groups that relate to more than a single domain name since the expiry of one of them would be inconclusive—and the expiry of all of them implausible. This affects about 10% of all resource groups and leads again to an underestimation of vulnerable resources. We can improve on this situation in the future by filtering out popular domain names beforehand that are unlikely to expire, for instance, like `gmail.com`.

We further acknowledge that our technique is incapable of detecting attacks in the past. Since we disregard resources with recent BGP activity, these resources could potentially be hijacked already. Hence, attacks that might have started before the deployment of our analysis technique are beyond the scope of our approach. With the PHEW system being deployed on a continuous basis, however, this limitation is naturally resolved over time.

Ethical Considerations We have identified vulnerable resources, and feel obliged to protect them. Since any attacker could readily take over these resources, we do not publish technical details before contacting endangered resource holders. Although communication via e-mail is futile due to expired domains, we can fall back on telephone numbers provided in the RIPE database to reach out for these operators. With our PHEW monitoring system deployed in the future, we intend to learn about imminent threats and contact resource holders in good time. In any case, we adhere to responsible disclosure practices. For inactive operators that cannot be contacted anymore, we further discussed the re-registration of domain names at our institute as a preventive measure.

PART THREE

Monitoring the Threat

10 CHAPTER TEN
Evaluation of *HEAP*

NOTE *This chapter contains prior publication.*

Published in **IEEE Journal on Selected Areas in Communications–Special Issue on Measuring and Troubleshooting the Internet (JSAC-SI-MT), 2016.**
Sections 10.1 to 10.3 are based on previous work [8]. For comparison, earlier results are added to the evaluation.

Published in **Proceedings of the International Workshop on Traffic Monitoring and Analysis (TMA), 2015.**
Sections 10.1 and 10.2 incorporate results from previous work [7]. No substantial changes are made.

Published in **Proceedings of the IEEE ICC Communications and Information Systems Security Symposium (ICC CISS), 2014.**
Section 10.4 is based on previous work [5]. No substantial changes are made.

SUMMARY In Chapter 7, we presented HEAP, a framework aimed at identifying legitimate causes for routing anomalies. HEAP takes input from existing hijacking detection frameworks and assesses the legitimacy of corresponding alarms. We evaluate our method against a superset of potential hijacking alarms and carry out case studies to put our results into perspective. In Section 10.1, we thoroughly analyze the characteristics of HEAP. By studying common day-to-day events, we are able to establish a base line for its validation capabilities. We see that HEAP is able to legitimize up to 56.93% of such events, and we obtain an even higher rate of 78.56% for a more focused set of relevant alarms in Section 10.2. To demonstrate the practical effectiveness of HEAP, we further apply our assessment scheme to publicly reported hijacking alarms in Section 10.3. We show that even such a set of highly suspicious events still contains nearly 10% of false positives. A detailed case study carried out in Section 10.4 illustrates the usefulness of HEAP with respect to practical validation of hijacking alarms.

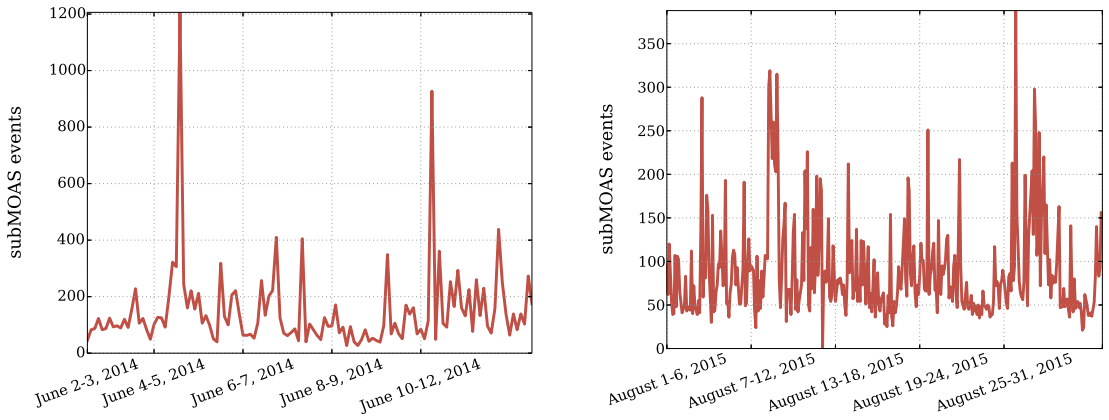
10.1 Overall Results

Most of the proposed detection techniques in previous work (see Table 6.1) do not offer publicly available interfaces yet. We compensate for the resulting lack of real alarms by studying common subMOAS conflicts (refer to Section 5.2.1) observed in BGP. Such cases occur numerous times per day and do not indicate attacks per se. Although more careful heuristics should be employed in practice to actually feed suspicious events into HEAP, we are able to establish a base line for its validation capabilities nonetheless. To verify our results, we deployed HEAP in two subsequent years for 2 weeks, and 4 weeks respectively. We further use HEAP to cross-check a set of publicly reported real hijacking alarms and demonstrate its practical usefulness in identifying false positives.

Experiment Setup

Our evaluation setup comprises several steps that are repeatedly executed. First, we obtain a full BGP table export holding all prefixes currently present in the global routing system and construct a binary prefix tree such that a tree node holds the date and origin AS of an announcement. To discover emerging subMOAS events, we obtain subsequent BGP messages and update the binary tree accordingly. We consequently extract all *strict* subMOAS events (see Section 5.2.1) from the tree that newly appeared in the BGP updates. For these, we apply our filters individually, i.e. we query our graph database for business and resource relations, construct an event-specific AS-level topology, and initiate SSL/TLS measurements for affected hosts that also appear in our ground truth data set. We then retrieve the scan results from successful SSL/TLS handshakes and compare the cryptographic host keys with those from our initial ground truth scan. As discussed in Subsection 7.2.2, we need to ensure that our scans actually reached a targeted prefix, since our BGP view, respectively the subMOAS events, might be outdated at the time of observation. Thus, we re-evaluate the aforementioned BGP update messages and discard previous scan results for which a subMOAS event changed or vanished during a scan. We accordingly sanitize the data in our ground truth, too, to ensure that no initially scanned SSL/TLS hosts were affected by subMOAS events.

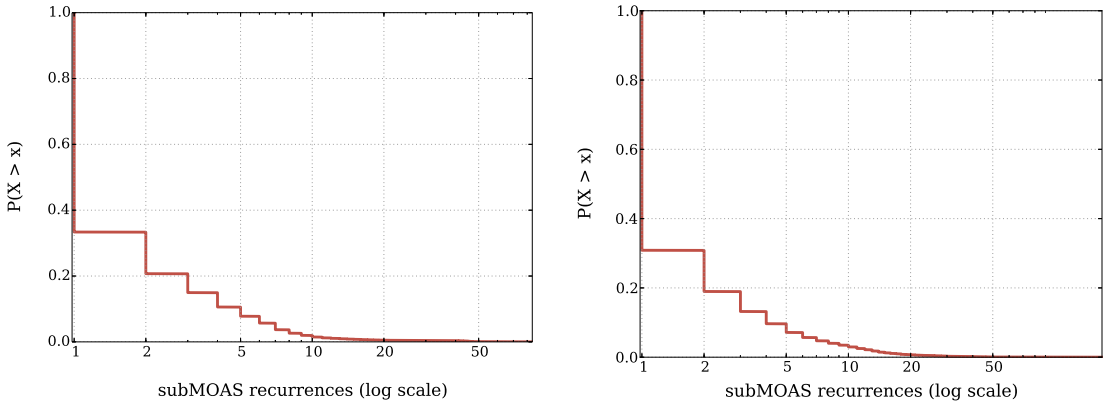
For the following evaluation, we utilized publicly available BGP data from RouteViews Oregon [222], which provides BGP tables every two hours. As a consequence, we cannot recognize shorter-lived events. This is no inherent limitation: In productive environments, HEAP can be interfaced with a live stream of BGP data, e.g. directly obtained from BGP routers or from services like BGPmon [223].



(a) Experiments during June 2-12, 2014.

(b) Experiments during August 1-31, 2015.

Figure 10.1: subMOAS events observed in our experiments.



(a) Experiments during June 2-12, 2014.

(b) Experiments during August 1-31, 2015.

Figure 10.2: Distribution of subMOAS reoccurrences (CCDF).

Overall Results

Figure 10.1 shows the frequency of subMOAS events observed for two analysis periods in June, 2014 and August, 2015. On average, we encountered 88 (*in 2014:* 148) events every two hours. The minimum number is 1, the maximum number is 388 (*in 2014:* 1,206). Figure 10.2 gives details on subMOAS events that occurred more than once, i.e. concerned the same prefixes and ASes. On average, subMOAS events recurred 2.2 times in both analysis periods, with a maximum of 169 (*in 2014:* 84) reoccurrences. In the following evaluation, multiple occurences of identical subMOAS events are considered only once.

	total	in %
All subMOAS events	**8,071**	**100.0%**
IRR analysis	870	10.78%
SSL/TLS scans	1,851	22.93%
topology reasoning	2,560	31.72%
Legitimate events (cum.)	**3,755**	**46.53%**

(a) Experiments during June 2-12, 2014.

	total	in %
All subMOAS events	**14,050**	**100.0%**
IRR analysis	5,699	40.56%
SSL/TLS scans	2,639	18.78%
topology reasoning	2,328	16.57%
Legitimate events (cum.)	**7,998**	**56.93%**

(b) Experiments during August 1-31, 2015.

Table 10.1: Overview of HEAP results.

During our experiments, we observed a total of 14,050 (*in 2014:* 8,071) unique subMOAS events. Our data sources cover 11,222, i.e. 79.87% (*in 2014:* 11,22, i.e. 59.41%) of these events. Hence, our coverage can still be increased, which suggests that extending HEAP by additional filters can further improve our legitimization results. By feeding the subMOAS events into HEAP, we are able to legitimize 56.93% (*in 2014:* 46.53%). Table 10.1 presents an overview of individual filter results. Note that an event might be legitimized by multiple filters: In total, we obtain 10,666 (*in 2014:* 5,281) legitimate events, which amount to 7,998 (*in 2014:* 3,755) distinct cases.

At the same time, 5,653 (*in 2014:* 2,256) of these cases were legitimized by only a single filter, i.e. each filter contributes unique results. The IRR analysis yields 3,660 (*in 2014:* 447) unique legitimized cases, followed by SSL/TLS scans with 1,244 (*in 2014:* 584) cases and our topology reasoning with 749 (*in 2014:* 1,225) cases. Overall, we are able to legitimize about half of all subMOAS events. We will see that our technique performs even better under more realistic conditions, i.e. for alarms relating to networks that are of high value for an attacker.

In-depth Analysis of the IRR Filter

For the experiment in 2014, Table 10.2 shows how effective our RIPE-based filter is at eliminating legitimate subMOAS events. We identified 870 legitimate causes for 1,048 covered cases, i.e. for cases where the affected IP prefixes and autonomous systems were registered by RIPE. While the overall validation rate of 10.78% appears to be relatively low, it compares well to covered events, i.e. 12.99% of the subMOAS events, which yields a relative legitimization rate of 83.02%. This pleasingly high percentage motivates our decision to increase coverage by including further IRR databases.

With the help of all five IRR databases—AfriNIC, APNIC, ARIN, LACNIC, and RIPE—we can legitimize 40.56% of all subMOAS events observed during our analysis period in 2015,

All subMOAS events	RIPE	
8,071	total	in %
Covered subMOAS events	**1,048**	**12.99%**
Inference rule *(business relationships)*		
`br_rpsl`	362	4.49%
`br_mntner`	519	6.43%
`br_org`	51	0.63%
`br_org_mntner`	145	1.80%
Inference rule *(resource holders)*		
`rh_route`	692	8.57%
`rh_mntner`	599	7.42%
`rh_org`	159	1.97%
`rh_org_mntner`	160	1.98%
Legitimate events (cum.)	**870**	**10.78%**
Legitimization rate (rel.)		83.02%

Table 10.2: Overview of HEAP results (IRR filter). June 2-12, 2014.

i.e. nearly four times as much compared to 2014. We identified 5,971 legitimate causes for 11,530 covered events, i.e. for cases where the affected IP prefixes and ASes were registered in one of the IRR databases. Note that some of these resources are registered in multiple databases. Overall, we legitimize a total of 5,699 distinct cases out of 10,500 covered events.

Table 10.3 shows details on the effectiveness of individual IRR filters at eliminating benign subMOAS events. Entries marked with *n/a* are not applicable for the respective registrar due to missing data, which its database model either not provides for, or which is removed with respect to privacy concerns (see Subsection 7.2.1 for details). The highest coverage of subMOAS events is provided by ARIN (37.61%), while AfriNIC and LACNIC cover less than 5%. At the same time, the ARIN filter legitimizes a comparatively low fraction (38.19%) of its covered events due to missing maintainer and RPSL information in the respective IRR data model. In absolute terms, RIPE, ARIN, and APNIC yield the highest number of legitimized events. Note that LACNIC removes all privacy-related information from its IRR database snapshots. As a consequence, none of its covered subMOAS events can be legitimized.

In Subsection 7.2.1, Tables 7.1 and 7.2 already suggested that filters based on *org*, *import*, and *maintained_by* relations, i.e. filters utilizing `ORGANIZATION`, `ROUTE` and `MNTNER` objects, show the potential to perform best due to rich relations between these resource objects. Our results confirm this assumption. The most effective filters are based on the `ORGANIZATION` objects in the ARIN database (11.86%), followed by `ROUTE` objects in the RIPE database (11.17%). Where applicable, maintainer relations are highly effective as well

All subMOAS events	**AfriNIC**		**APNIC**		**ARIN**		**LACNIC**		**RIPE**	
14,050	total	in %	total	in %	total	in %	total	in %	total	in %
Covered subMOAS events	**340**	**2.42%**	**2,020**	**14.38%**	**5,284**	**37.61%**	**574**	**4.09%**	**3,312**	**23.57%**
Inference rule *(business relationships)*										
`br_rpsl`	0	0.00%	94	0.67%	n/a		n/a		819	5.83%
`br_mntner`	19	0.14%	1,005	7.15%	n/a		n/a		783	5.57%
`br_org`	41	0.29%	n/a		970	6.90%	n/a		405	2.88%
`br_org_mntner`	9	0.06%	n/a		n/a		n/a		136	0.97%
Inference rule *(resource holders)*										
`rh_route`	n/a		249	1.77%	378	2.69%	n/a		1,569	11.17%
`rh_mntner`	21	0.15%	1,167	8.31%	n/a		n/a		719	5.12%
`rh_org`	89	0.63%	n/a		1,666	11.86%	n/a		758	5.40%
`rh_org_mntner`	10	0.07%	n/a		n/a		n/a		139	0.99%
Legitimate events (cum.)	**104**	**0.74%**	**1,397**	**9.94%**	**2,018**	**14.36%**	**0**	**0.00%**	**2,452**	**17.45%**
Legitimization rate (rel.)		30.59%		69.16%		38.19%		0.00%		74.03%

Table 10.3: Overview of HEAP results (IRR filter). August 1-31, 2015.

(up to 8.31%). Interestingly, filters that are based on the combination of `ORGANIZATION` and `MNTNER` objects (`br_org_mntner` and `rh_org_mntner`) contribute least to the overall validation results, i.e. 0.99% at most. In total, filter rules that aim at identifying business relationships can eliminate 25.17% of all events, while rules that establish confirmation of resource holdership yield 36.39% legitimate events. If we combine them, we find that 40.56% of all subMOAS events (or 54.28% of all covered events) can be legitimized.

In-depth Analysis of the SSL/TLS Filter

It is worthwile to study the performance of our SSL/TLS filter in more detail. Table 10.4 shows further information about scans to individual ground truth hosts. In 2014, we scanned a total of 37,043 SSL/TLS hosts distributed over 2,116 (26.22%) of all subMOAS events. Overall, 89.00% of these hosts presented the same SSL/TLS key, leading to 1,851 (22.93%) legitimate events. Thus, we obtain a relative legitimization rate of 87.48% for covered subMOAS prefixes, i.e. for such prefixes with at least one SSL/TLS-enabled host. This encouragingly high percentage suggests to further increase coverage by extending the set of known SSL/TLS hosts in our ground truth.

In 2015, we scanned various SSL/TLS protocols in addition to HTTPS. We were able to legitimize 2,639 (18.78%) of all subMOAS events. Note that we aggregate scans to individual hosts in Table 10.4b, i.e. to hosts that run more than one SSL/TLS protocol. If any

	total	in %
Individual SSL/TLS host scans	**37,043**	**100.0%**
same SSL/TLS key	32,968	89.00%
different SSL/TLS key	773	2.09%
no response (port closed)	3,302	8.91%
discarded scan results	986	2.66%

(a) Experiments during June 2-12, 2014.

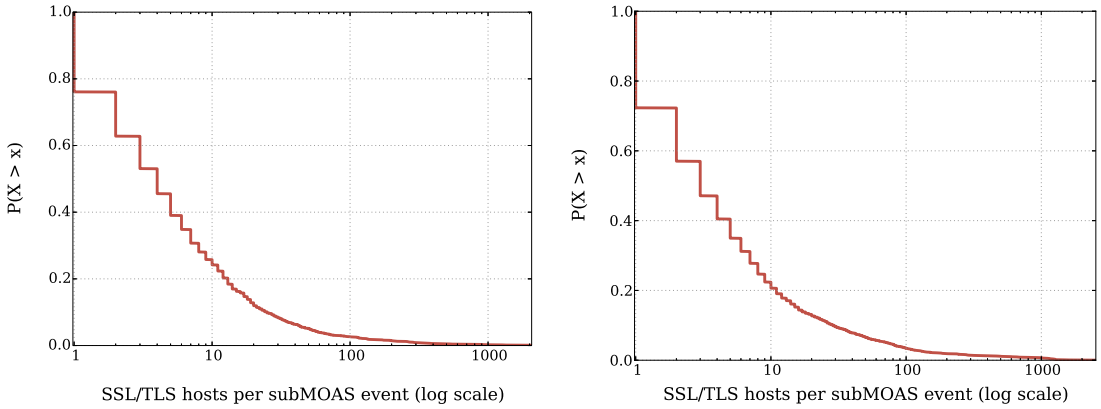
	total	in %
Individual SSL/TLS host scans	**95,486**	**100.0%**
same SSL/TLS key	45,572	47.73%
different SSL/TLS key	13,202	13.83%
no response (port closed)	19,119	20.02%
discarded scan results	17,593	18.42%

(b) Experiments during August 1-31, 2015.

Table 10.4: Overview of HEAP results (SSL/TLS filter).

of a host's keys remains unchanged during a subMOAS event, we consider the host to be not compromised and legitimize the corresponding event. Hosts with changed keys as well as unresponsive hosts accordingly imply that *all* of a host's keys have changed, and *all* of its previously open ports were closed respectively. Note that we discarded 18.42% of the scan results, for which the subMOAS events changed or vanished during the scans. Another 20.02% of our ground truth hosts did not respond to the validation scans. Overall, 47.73% of the retrieved SSL/TLS keys did not change, i.e. the percentage of stable SSL/TSL keys nearly halved compared to 2014. The reason for this decline is a higher number of unresponsive hosts resulting from a less intrusive measurement of the ground truth (see Section 7.2.2 and the discussion below). In spite of the utilization of multiple SSL/TLS protocols, our coverage does not significantly increase either compared to 2014. With the extended approach, we cover 23.03% (*in 2014:* 26.22%) of the subMOAS events, which implies that HTTPS hosts already represent a large subset of all SSL/TLS-enabled hosts. Table 7.3 in Section 7.2.2 supports this finding.

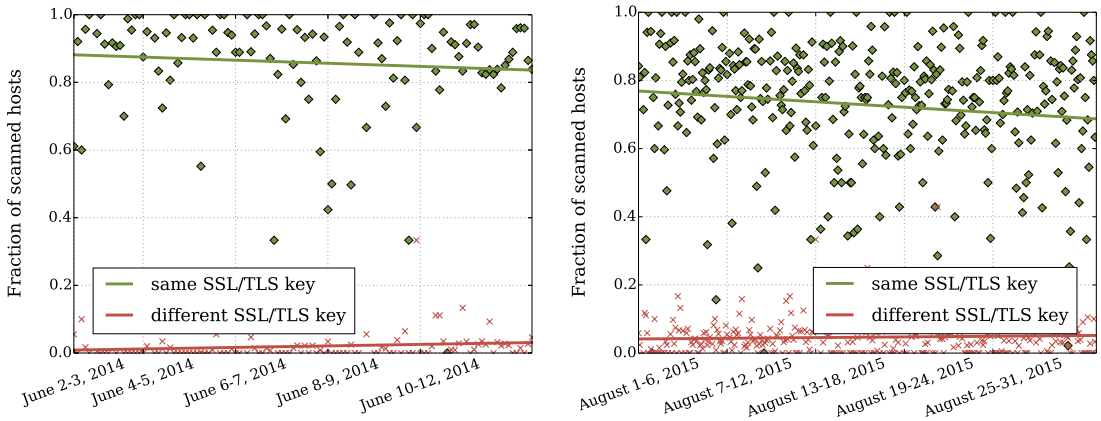
Despite the comparatively high number of unusable scan results, we obtain a relative legitimization rate of 81.55% for covered subMOAS prefixes (*in 2014:* 87.48%). To elaborate, Figure 10.3 shows the distribution of available SSL/TLS hosts per subMOAS prefix for both analysis periods. On average, we scanned 24.38 (*in 2014:* 17.51) SSL/TLS hosts per covered event. In spite of a higher number of individual scans in 2015, the distribution of available hosts per event shows a similar pattern for both analysis periods (see Figure 10.3), which again supports the assumption that HTTPS hosts already account for a majority of SSL/TLS-enabled hosts. For 50% of the events, our ground truth comprises more than three available hosts. 20% of the events provide more than ten hosts, and about 5% of them even more than 100. The maximum number of available hosts is 2,531 (*in 2014:* 2,070). These figures actually allow our SSL/TLS filter to be highly robust against outages of individual hosts or services, since it is enough for our technique to confirm that *at least one* cryptographic key on *any* of the affected hosts inside a prefix remains unchanged during a subMOAS event.



(a) Experiments during June 2-12, 2014.

(b) Experiments during August 1-31, 2015.

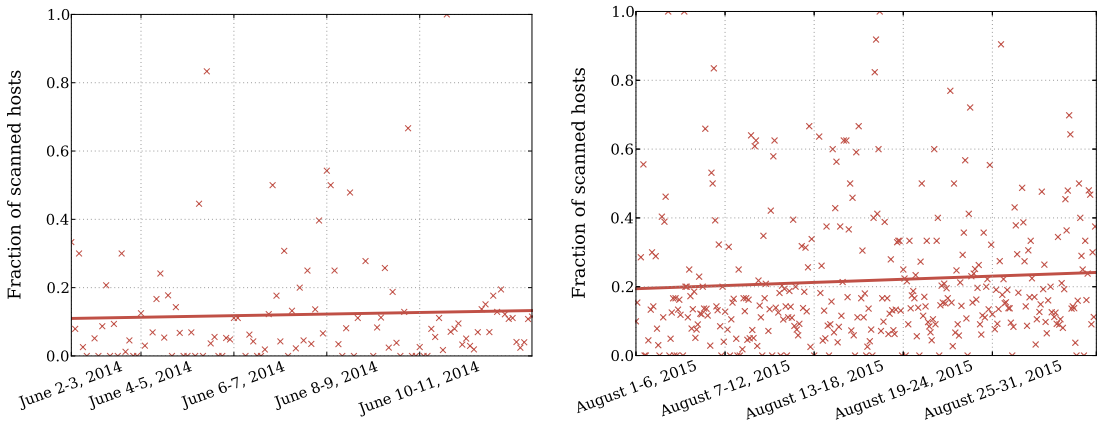
Figure 10.3: Available SSL/TLS hosts per subMOAS event (CCDF).



(a) Experiments during June 2-12, 2014.

(b) Experiments during August 1-31, 2015.

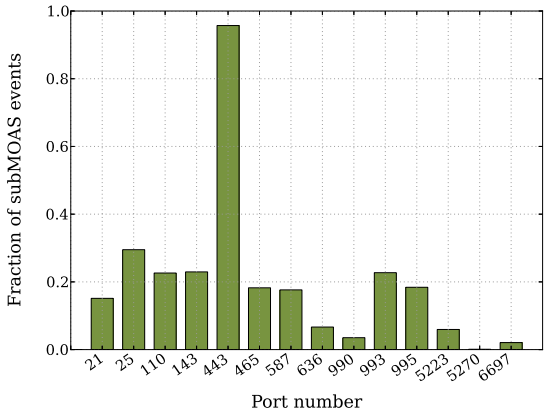
Figure 10.4: Fraction of validated SSL/TLS keys.



(a) Experiments during June 2-12, 2014.

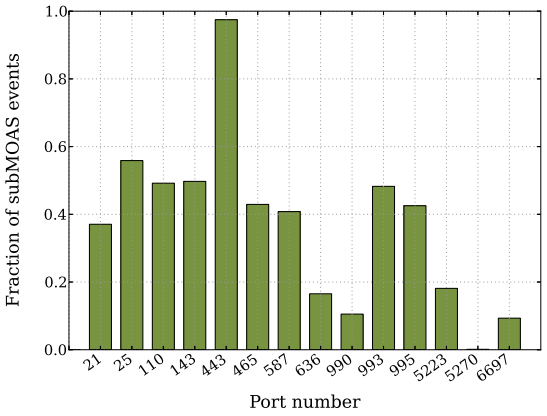
(b) Experiments during August 1-31, 2015.

Figure 10.5: Fraction of unresponsive SSL/TLS hosts.



Experiments during August 1-31, 2015.

Figure 10.6: SSL/TLS ports.

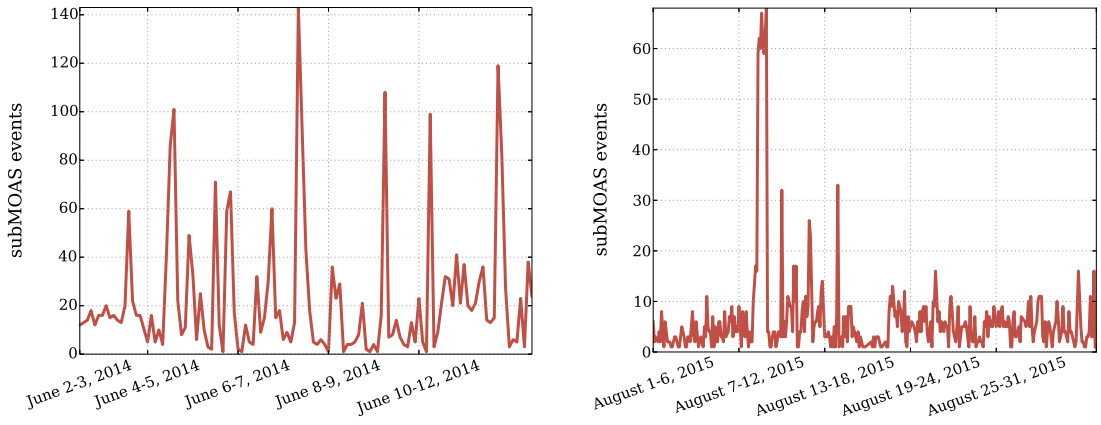


Experiments during August 1-31, 2015.

Figure 10.7: Alexa-based SSL/TLS ports.
Events affecting top 1 million web sites only.

In 2014, we re-scanned all hosts in our ground truth data set before the beginning of the experiment in order to identify long-term stable SSL/TLS hosts. The effects can be seen in Figure 10.4a, where the fraction of changing keys shifts rather slowly over the analysis period. While a certain decline in stable keys is visible, it remains in the range of less than 10%. Likewise, the number of unresponsive hosts hardly increases (Figure 10.5a). We skipped this re-scanning process in 2015 in order to reduce intrusiveness. Accordingly, Figure 10.4b shows a higher fluctuation of keys, while the churn rate is similar to our experiment in 2014. Consequently, the fraction of unresponsive hosts has doubled as can be seen in Figure 10.5b. However, this circumstance does not weaken our results, since we disregard such hosts in both experiments—either in advance (2014), or during the experiment (2015). Note that in any case, we need to occasionally renew our ground truth data set. Our findings suggest that the interval for obtaining new ground truth hosts can be set to one month or even longer.

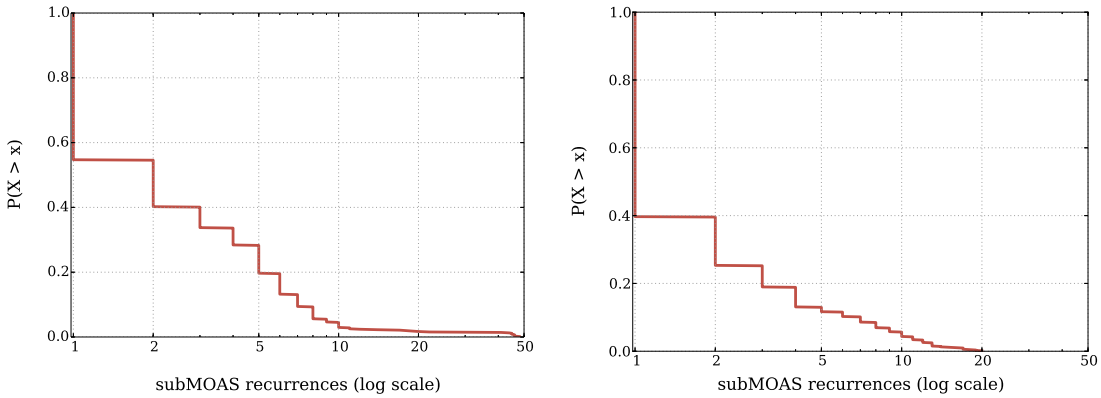
Another interesting fact with respect to the legitimization capabilities of our SSL/TLS filter is the set of ports, i.e. network protocols, that contribute to the validation. Figure 10.6 illustrates that for more than 95% of covered subMOAS events, respectively events with at least one SSL/TLS host available, HTTPS servers can be utilized for the validation. Other protocols like LDAPS, FTPS, XMPPS, and IRCS are apparently ill-suited for our purposes. While arguably adding robustness against outages of HTTPS services for 1,458 (54.96%) HTTPS-validated cases, these protocols contribute as few as 75 (2.83%) unique legitimate events. These facts will be taken into account for future re-scans of our ground truth.



(a) Experiments during June 2-12, 2014.

(b) Experiments during August 1-31, 2015.

Figure 10.8: Alexa-based subMOAS events observed in our experiments.
Events affecting top 1 million web sites only.



(a) Experiments during June 2-12, 2014.

(b) Experiments during August 1-31, 2015.

Figure 10.9: Alexa-based distribution of subMOAS reoccurrences (CCDF).
Events affecting top 1 million web sites only.

10.2 Case Study: "The Top One Million"

So far, we evaluated HEAP with respect to its legitimization capabilities of rather general day-to-day events. To get a more realistic view of its capabilities to identifiy false positive hijacking alarms in practice, we conduct a case study as follows. We assume that an attacker has little interest in hijacking small and insignificant networks since the corresponding address space can be easily monitored and, more importantly, has no particular reputation in terms of globally whitelisted IP ranges. Instead, we assume that a real attacker would typically hijack smaller parts of large and popular networks in order to launch and sustain

	total	in %
All subMOAS events	**711**	**100.0%**
IRR analysis	106	14.91%
SSL/TLS scans	514	72.29%
topology reasoning	285	40.08%
Legitimate events (cum.)	**575**	**80.87%**

(a) Experiments during June 2-12, 2014.

	total	in %
All subMOAS events	**849**	**100.0%**
IRR analysis	294	34.63%
SSL/TLS scans	576	67.85%
topology reasoning	146	17.20%
Legitimate events (cum.)	**689**	**81.15%**

(b) Experiments during August 1-31, 2015.

Table 10.5: Overview of Alexa-based HEAP results.
Events affecting top 1 million web sites only.

malicious activities. As a consequence, we evaluate HEAP with respect to the more well-known networks. To this end, we utilize a list of the top one million web sites provided by *Alexa Inc.* [224]. For each of the domain names in this list, we perform a reverse DNS lookup. Since multiple domains or subdomains can be hosted on a single server, we obtain a total of 522,655 (*in 2014:* 618,816) different IP addresses. We consequently re-assess all subMOAS events in both analysis periods and restrict the input fed into HEAP to those events that affect the aforementioned addresses.

Overall Results

We find that only a small subset of 849 (*in 2014:* 711) distinct subMOAS cases out of the full set of 14,050 (*in 2014:* 8,071) events affects these popular networks, i.e. less than 10%. Figure 10.8 shows the corresponding distribution of events. This observation confirms that high-ranking networks account for less routing anomalies in BGP. At the same time, we see that the average number of recurring events increases by nearly 13.84% (see also Figure 10.9), which indicates intentional use of the subMOAS announcements and supports arguments against misconfiguration or attacks. Hence, one would expect to identify a larger fraction of legitimate subMOAS events.

Table 10.5 presents the overall results. We see that HEAP yields a significantly higher legitimization rate of 81.15% (*in 2014:* 80.87%) for subMOAS events related to the top one million web sites as compared to 56.93% (*in 2014:* 46.53%) for all observed events. A major reason for this improvement is an increase in coverage of our methodology. The combined filter set now covers 98.82% (*in 2014:* 91.56%) of the respective events compared to 79.87% (*in 2014:* 59.41%) in the section before. Most notably, the performance of our SSL/TLS filter is more than three times as high due to a higher density of SSL/TLS hosts, which is not surprising for highly frequented networks.

All subMOAS events	RIPE	
711	total	in %
Covered subMOAS events	**116**	**16.32%**
Inference rule *(business relationships)*		
`br_rpsl`	37	5.20%
`br_mntner`	81	11.39%
`br_org`	6	0.84%
`br_org_mntner`	29	4.08%
Inference rule *(resource holders)*		
`rh_route`	94	13.22%
`rh_mntner`	85	11.96%
`rh_org`	20	2.81%
`rh_org_mntner`	33	4.64%
Legitimate events (cum.)	**106**	**14.91%**
Legitimization rate (rel.)		91.38%

Table 10.6: Overview of Alexa-based HEAP results (IRR filter). June 2-12, 2014.
Events affecting top 1 million web sites only.

In-depth Analysis of the IRR Filter

Table 10.6 presents the RIPE filter results for our experiment in 2014. For subMOAS events affecting the top 1 million web sites, we observe a moderate increase in the percentage of both covered and legitimized events. The fraction of covered events that can be legitimized notably increases from 83.02% to 91.38% (compare to Table 10.2). While all individual filters show better performance, the greatest increase is observed for `MNTNER`-based filters (`br_mntner`, `br_org_mntner`, `rh_mntner` and `rh_org_mntner`). A plausible explanation for this effect could be that high-ranking networks often operate multiple autonomous systems under a single administrative control, hence we are able to identify a higher number of shared *maintained_by* relations.

In 2015, the coverage of our IRR filters changes significantly for Alexa-based events (see Table 10.7). The fraction of subMOAS events that affect the ARIN and APNIC service region increases from 37.61% to 53.71%, and from 14.38% to 25.56% respectively (compare to Table 10.3). The remaining IRR filters, and in particular LACNIC and AfriNIC, lose part of their coverage. Altogether, the overall coverage of subMOAS events increases from 74.73% to 93.64%, while the overall legitimization rate slightly decreases from 40.56% to 34.63%.

With respect to the legitimized events, no reasonable conclusions can be drawn for AfriNIC due to the low absolute number of affected events. Surprisingly, all other IRR filters yield a lower legitimization rate for covered events. In absolute terms, APNIC nearly

All subMOAS events	AfriNIC		APNIC		ARIN		LACNIC		RIPE	
849	total	in %	total	in %	total	in %	total	in %	total	in %
Covered subMOAS events	**6**	**0.71%**	**217**	**25.56%**	**456**	**53.71%**	**22**	**2.59%**	**139**	**16.37%**
Inference rule *(business relationships)*										
`br_rpsl`	0	0.00%	15	1.77%	n/a		n/a		42	4.95%
`br_mntner`	0	0.00%	57	6.71%	n/a		n/a		16	1.89%
`br_org`	3	0.35%	n/a		16	1.89%	n/a		10	1.18%
`br_org_mntner`	1	0.12%	n/a		n/a		n/a		10	1.18%
Inference rule *(resource holders)*										
`rh_route`	n/a		18	2.12%	15	1.77%	n/a		64	7.54%
`rh_mntner`	0	0.00%	114	13.43%	n/a		n/a		40	4.71%
`rh_org`	4	0.47%	n/a		53	6.24%	n/a		36	4.24%
`rh_org_mntner`	1	0.12%	n/a		n/a		n/a		10	1.18%
Legitimate events (cum.)	**4**	**0.47%**	**138**	**16.25%**	**62**	**7.30%**	**0**	**0.00%**	**98**	**11.54%**
Legitimization rate (rel.)		66.67%		63.60%		13.60%		0.00%		70.50%

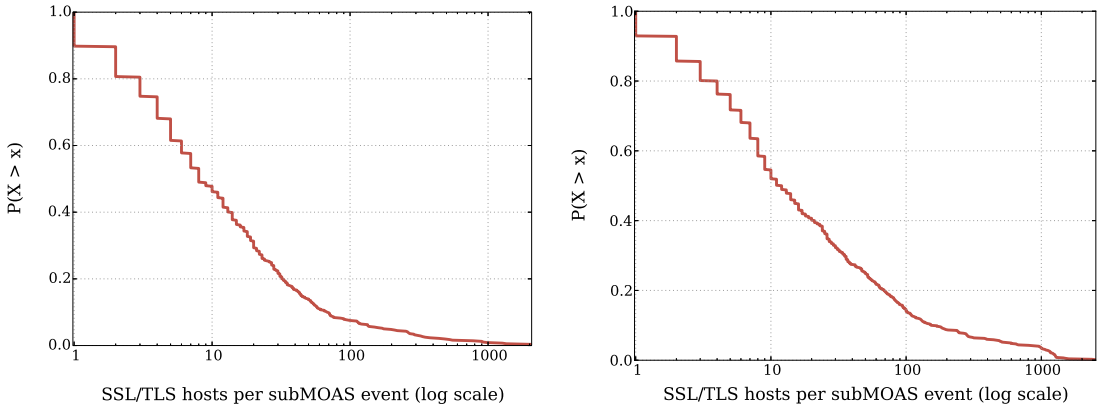
Table 10.7: Overview of Alexa-based HEAP results (IRR filter). August 1-31, 2015.
Events affecting top 1 million web sites only.

doubles the number of legitimized events (16.25% compared to 9.94%), while the results for ARIN and RIPE degrade from 14.36% to 7.30%, and from 17.45% to 11.54% respectively.

To put these results into perspective, we use our graph database to identify the responsible registrars for all of the top 1 million web sites, i.e. the respective databases that hold information about corresponding *Alexa* IP prefixes. Most of the prefixes are registered in the ARIN database (99.99%), followed by RIPE (98.86%) and APNIC (90.77%). LACNIC (2.38%) and AfriNIC (0.54%) only account for a small number of these web sites. It is apparent that the largest part of corresponding `INETNUM` objects is registered in multiple IRR databases. Such networks often relate to several regional subsidiaries of worldwide operating companies under independent administrative control, which possibly explains the slight decrease in our IRR legitimization rate in spite of an increase in coverage. Hence, in addition to our filter rules presented in Section 7.2.1, more comprehensive rules should be developed in the future to model such complex corporate structures.

In-depth Analysis of the SSL/TLS Filter

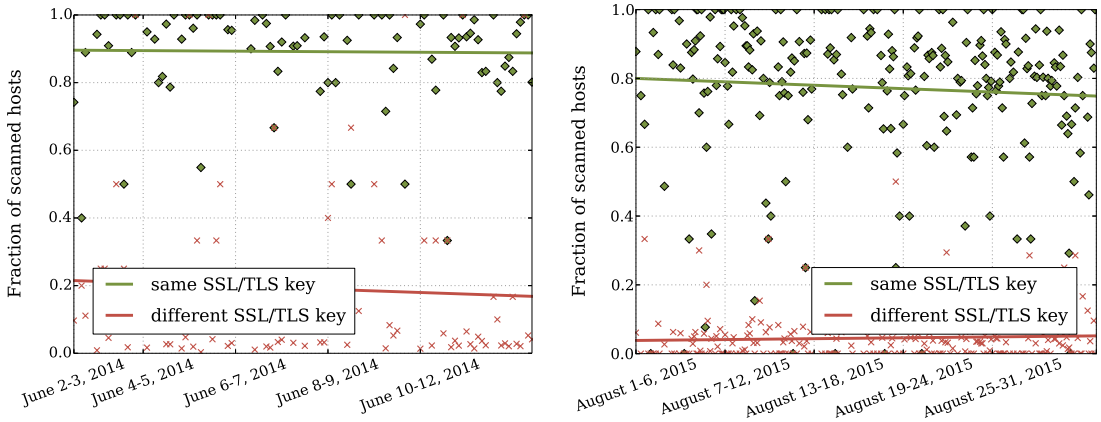
The number of subMOAS events that affect the top one million web sites account for 6.04% of all observed events. The set of available SSL/TLS hosts scanned to assess these events, however, comprises 70,464 (*in 2014:* 26,200) hosts, i.e. 73.80% (*in 2014:* 70.73%). As a consequence, the coverage of our SSL/TLS filter greatly increases from 27.88% (*in 2014:*



(a) Experiments during June 2-12, 2014.

(b) Experiments during August 1-31, 2015.

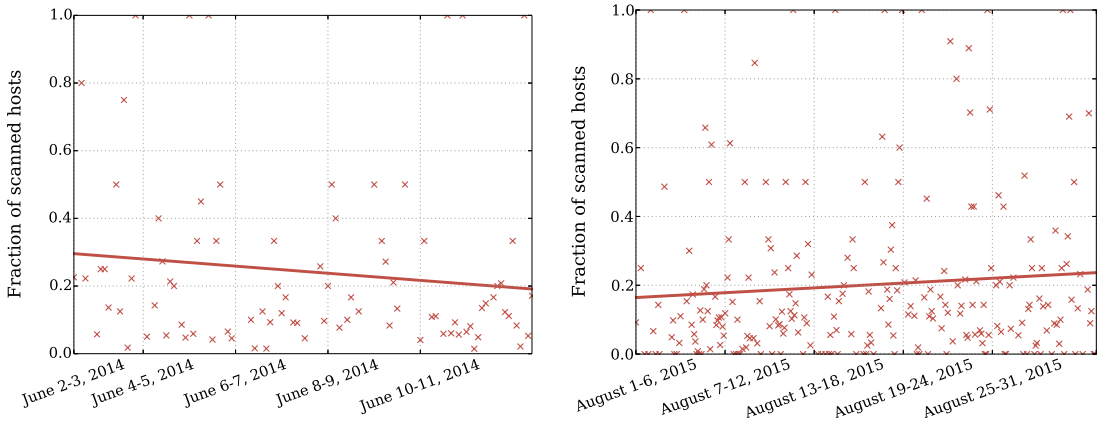
Figure 10.10: Alexa-based SSL/TLS hosts per subMOAS event (CCDF).
Events affecting top 1 million web sites only.



(a) Experiments during June 2-12, 2014.

(b) Experiments during August 1-31, 2015.

Figure 10.11: Alexa-based fraction of validated SSL/TLS keys.
Events affecting top 1 million web sites only.



(a) Experiments during June 2-12, 2014.

(b) Experiments during August 1-31, 2015.

Figure 10.12: Alexa-based fraction of unresponsive SSL/TLS hosts.
Events affecting top 1 million web sites only.

	total	in %
Individual SSL/TLS host scans	**26,200**	**100.0%**
same SSL/TLS key	22,758	86.86%
different SSL/TLS key	546	2.08%
no response (port closed)	2,362	9.02%
discarded scan results	534	2.04%

(a) Experiments during June 2-12, 2014.

	total	in %
Individual SSL/TLS host scans	**70,464**	**100.0%**
same SSL/TLS key	31,888	45.25%
different SSL/TLS key	6,508	9.24%
no response (port closed)	17,529	24.88%
discarded scan results	14,539	20.63%

(b) Experiments during August 1-31, 2015.

Table 10.8: Overview of Alexa-based HEAP results (SSL/TLS filter).
Events affecting top 1 million web sites only.

26.22%) to 88.34% (*in 2014:* 78.62%). At the same time, the legitimization rate is more than three times as high (compare Table 10.1 and Table 10.5).

Table 10.8 shows details about individual SSL/TLS scans. For both analysis periods, the obtained results are comparable to those in Table 10.4. The percentage of hosts with changing and unchanged keys as well as the fractions of unresponsive hosts and discarded scan results show the same distribution. Figures 10.11 and 10.12 indicate a slightly lower churn rate for SSL/TLS hosts over the duration of our experiments. The number of available SSL/TLS hosts per subMOAS event differs strikingly (see Figure 10.10). Nearly 50% of events that affect popular networks offer more than 10 SSL/TLS-enabled hosts, while 10% provide even more than 100 hosts. Given this high density of available hosts, a major increase in our legitimization rate as discussed in the aforegoing analysis is self-evident.

With respect to individual SSL/TLS services, we observe HTTPS hosts for 97.50% of all events (see Figure 10.7). Other SSL/TLS protocols show the same ratio to each other as before (compare to Figure 10.6), while their individual percentages roughly doubled. Despite the fact that these protocols only contribute 12 (1.60%) uniquely legitimized events, they add robustness to our filter against short-term unavailability of HTTPS services for 178 (31.96%) legitimate events.

10.3 Case Study: "Real Alarms"

With HEAP, we intend to provide a framework that enables reliable assessment of arbitrary subprefix hijacking alarms. To demonstrate its effectiveness, we study a set of real alarms reported by BGPmon.net [180] during August, 2015. This set consists of 85 highly suspicious subprefix hijacking alarms $\hat{\mathscr{A}}$ each given by

$$\hat{\mathscr{A}} = \{ vp_v \mid v \in \Sigma_{AS},\ p_v \subset \Pi_v \} \cup \{ ap'_v \mid a \in \Sigma_{AS},\ p'_v \subset p_v \}\,.$$

Reported hijacking alarms: **85**	**IRR analysis**		**SSL/TLS scans**		**Topology reasoning**		**Cumulative**	
	total	in %	total	in %	total	in %	total	in %
Covered hijacking alarms	60	70.59%	1	1.18%	3	3.53%	61	71.76%
False positives identified	**6**	**7.06%**	**0**	**0.00%**	**1**	**1.18%**	**7**	**8.24%**

Table 10.9: HEAP cross-check of BGPmon.net hijacking alarms [180].

During our evaluation, we observed 61 corresponding subMOAS events for which we applied our filtering scheme. The lower number of observed events compared to the full set of reported alarms results from technical aspects of our experiment design: 7 events lasted for less than two hours, while 9 events were not classified as strict subMOAS (see Section 10.1 for details). Another 8 events re-occured and were considered only once. For the remaining cases, we retroactively applied our IRR and topology-based filters, while we naturally lack SSL/TLS measurement data from targeted scans.

Table 10.9 shows the overall legitimization results. In total, our methodology covered 61 (71.76%) distinct alarms, of which 7 (8.24%) were explicitly identified as false positives. Note that BGPmon.net already provides a highly focused set of alarms, since as few as 85 (0.61%) out of the total number of 14,050 subMOAS events were reported during August, 2015. It is thus suprising to see that these reports still contain nearly 10% false alarms. At the same time, these findings evidence the strength of a cross-validation with HEAP. We plan to provide a public interface to HEAP, which accepts input alarms in the format as specified above supplemented by timestamps of the events. Current and future detection systems may then benefit from our validation scheme as well.

10.4 Case Study: "The Bulgarian Case"

In addition to a live assessment of routing incidents, our HEAP framework can also be used to carry out post-mortem analyses. In Section 7.3, we discussed the usefulness of manual IRR inspection and the added value of forensic traffic flow inspection. The following case study demonstrates the strength of such an approach.

In an effort to study malicious intent behind BGP hijacking attacks, we losely connected HEAP with a spam-based hijacking detection framework developed by fellow colleagues at Institut Eurécom, France. Their framework analyzes spam data collected from spam traps, which receive about 4 million spam emails every day. Immediate traceroute path measurements are launched towards suspicious, spam-emitting networks [189]. Corresponding IP

prefixes that show routing changes in the data-plane during and after such an event are subsequently passed to individual HEAP filters. First, we rule out benign routing changes with our BGP control-plane monitoring scheme, which evaluates legitimate causes for (sub-)MOAS events [164]. This part of our filter chain is provided by Eurécom (refer to Subsection 7.2.3). Second, we utilize HEAP to assess the remaining alarms by studying routing consistency using our IRR graph databases, and we further analyze network traffic footprints collected at our academic network. A forensic analysis of past incidents, we skip the real-time assessment of SSL/TLS hosts.

In the following, we present a real case where suspicious routing changes coincide with a large volume of spam and web scam traffic from corresponding networks. By performing a cross-data sources analysis, we are able to reconstruct the routing history and network behavior of several prefixes announced in turn by three different autonomous systems for a period of about three months. Through this case study, we show that a correlation of malicious activities with suspicious routing events is insufficient to evidence harmful BGP hijacking attacks. At the same time, we learn that HEAP meets our expectation to reliably identify legitimate causes behind such events.

Course of Events

Based on several alarms raised by the spam detection system on February 3, 2013, we became aware of an incident taking place in Bulgaria. Several subMOAS and MOAS conflicts were subsequently observed for networks with emerging spam activities. To assess a potential hijacking attack and the role of all parties involved, we leverage our data sources and apprehend the course of events in great detail. Below, we present our findings in chronological order. Note that all results are anonymized with good cause.

Phase 1: *Normal Situation* Since 2008, the prefix `A.B.0.0/16` has been announced in BGP by a small Bulgarian ISP, in the following called *Alice*. This ISP is known to provide hosting services for a variety of customers. We did not observe announcements of more specific prefixes during the whole time of Phase 1. Figure 10.13 illustrates this initial situation.

Phase 2: *Hijack and Spam* On December 4, 2012, a second AS, called *Mallory*, started advertising a set of nine more specific `/24` prefixes, while *Alice* carried on with the original `/16` announcement (see Figure 10.13). The resulting subMOAS events could not be legitimized by our topology-based reasoning algorithm. By querying our IRR graph databases for the affected networks, we learned that *Mallory* supposedly is a VPS service provider also located in Bulgaria. A thorough web-search, returned no result for this specific company.

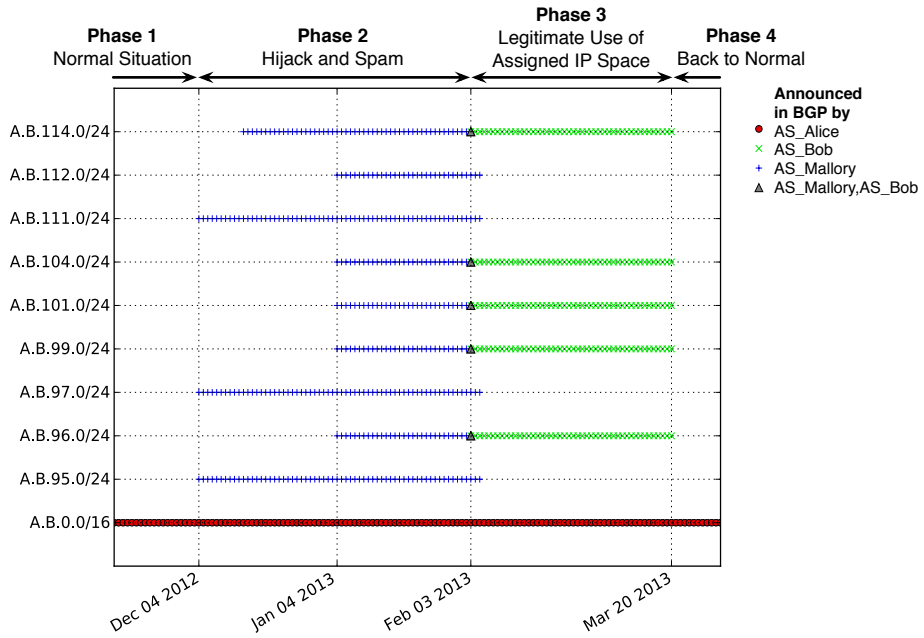


Figure 10.13: Analysis of BGP announcements for the *Bulgarian Case*.

(a) Spam. Figure 10.14 shows the amount of spam observed for IP addresses that are related to *Mallory's* prefix announcements. Our spam honeypots received up to 80 spam emails per day, which directly correlate with the announcement of the more specific networks. Note that based on spam signatures, our spam traps are capable to identify botnets responsible for individual spam campaigns. Since corresponding spam bots usually represent compromised client machines, no such botnet should be observed for hijacked IP space. Instead, in such a hijacking scenario, we expect spammers to operate their own infrastructure in order to emit spam. As a matter of fact, our spam traps did not report any involvement of botnets at all.

(b) Blacklisting. We further analyze a spam blacklist provided by the UCEPROTECT-Network [225]. The results are shown in Figure 10.15. Again, we observe a strong correlation between the BGP announcements, spam emails, and blacklisted IP addresses. Most prefixes had up to half of their IP addresses blacklisted for serveral days. Note that some blacklist entries persisted after the end of Phase 2, which we attribute to the one-week expiration period of corresponding records.

(c) Scam Hosting Infrastructures. By analyzing the spam emails received from *Mallory's* networks in more detail, we are able to identify advertised URLs. Out of 118 extracted domain names, 89 resolved to an IP address within six of the obtrusive prefixes. We conclude that the spamming activities also served as a platform to promote a scam infrastruc-

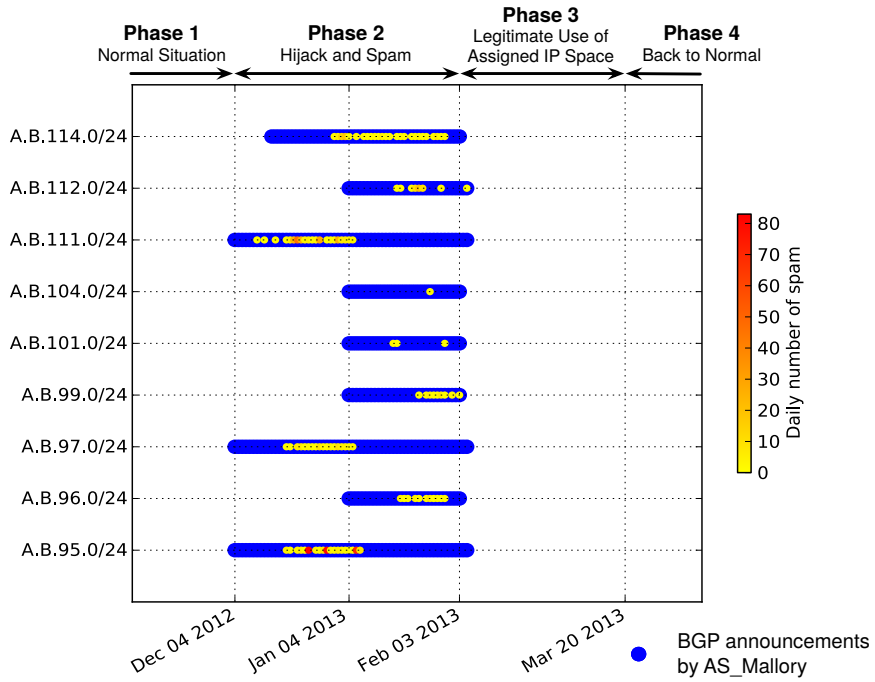


Figure 10.14: Analysis of spam emails for the *Bulgarian Case*.

ture hosted within these prefixes. About 90% of all scam hosts in the more specific prefixes coincided with IP addresses of spamming hosts, which indicates that the spammers took full advantage of the abused prefixes. It is interesting to see that most scam hosts followed a particular addressing scheme while being spread over all abused networks, for instance `A.B.{95,96,97,114}.14`. Similar characteristics apply for the resolution of domain names to IP addresses within the nine prefixes. All 89 resolvable domains were created at nearly the same time as the prefixes were first announced in BGP by *Mallory*. Altogether, our results suggest a single administrator operating both domains and network infrastructure.

(d) Traffic Flows. We further evaluate archived netflow data for the period of December 2012 to March 2013 and collect 13,001 inbound flows from the suspicious prefixes. The majority of these flows account for SMTP requests (71.0%), DNS replies (25.2%), HTTP replies (1.6%), and SMTP replies (1.4%). The remaining 1.8% of flows indicate traffic to an IRC server within our networks and to ephemeral UDP ports. For 97.4% of all incoming flows, we observed corresponding outgoing flows. An analysis of the IRC traffic revealed that these flows originated from 1,381 hosts spread over 254 different `/24` subnets within the `/16` prefix announced by *Alice*. Such orchestrated IRC traffic beyond the prefixes used by *Mallory* and across all networks of *Alice's* customers seems to be implausible. We assume that these flows attribute to IP spoofing activities, i.e. represent backscatter traffic unrelated to the

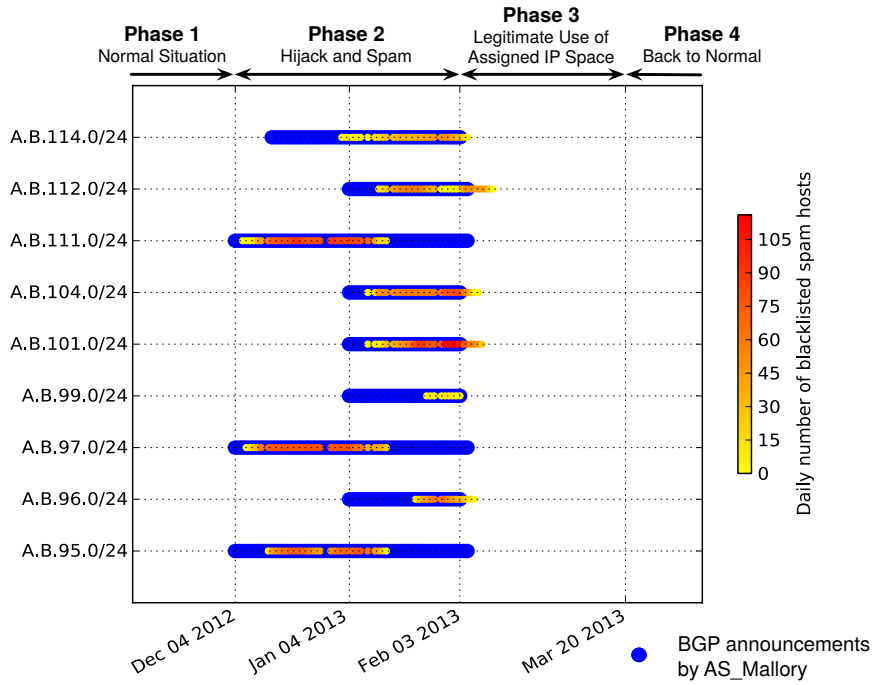


Figure 10.15: Analysis of blacklist records for the *Bulgarian Case*.

case at hand, and exclude them from our analysis. All incoming and outgoing connection requests are depicted in Figure 10.16. We observe yet again a strong correlation with our previous results. Out of a total of 925 IP addresses involved in the spammer's activities, 850 IP addresses were used to send spam mail. About 10% of these addresses were re-used for DNS or HTTP activities. We further found 30 distinct DNS servers mostly hosted in the prefixes `A.B.96.0/24` and `A.B.114.0/24`, which were queried over 3,000 times by clients in our networks. The flow data also shows 200 bidirectional HTTP connections to more than 100 web servers in the affected prefixes.

So far, our analysis confirms that a potential hijacking attack of more specific prefixes coincided with a large amount of spam sent from several hundred clients hosted in the corresponding networks. Furthermore, we learned that the attacker operated more than a hundred live services, i.e. DNS and HTTP, and presumably carried out phishing or similar fraudulent activities.

Phase 3: *Legitimate Use of Assigned IP Space* On February 3, 2013, a third AS, called *Bob*, started advertising five of the nine prefixes already announced by *Mallory*, resulting in MOAS conflicts for several hours. Soon afterwards, *Mallory* withdrew all of its prefix announcements. *Alice*, once more, kept on advertising the original `/16` prefix (see Figure 10.13). Individual spam hosts that used to reply to our traceroute probes on consecutive

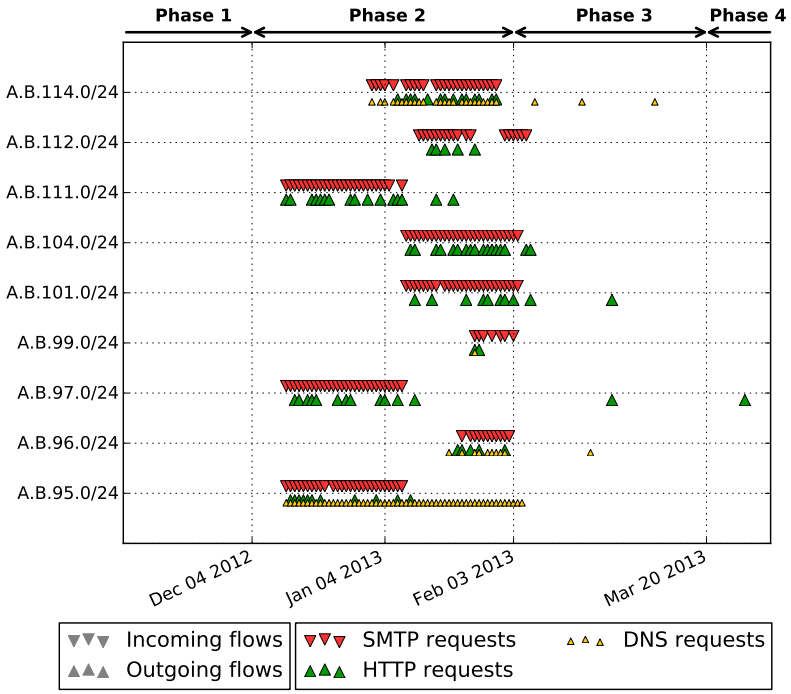


Figure 10.16: Analysis of traffic flow data for the *Bulgarian Case*.

days during Phase 2 suddenly became unreachable, which evidences a real change in network topology. We learned that according to their website, *Bob* is a business-to-business IT service provider located in Bulgaria as well. Note that in this case, our topology-based reasoning algorithm legitimized the corresponding subMOAS event between *Alice* and *Bob*, since all five /24 prefixes were advertised via *Alice* acting as *Bob's* upstream provider. Figure 10.17 shows the corresponding BGP topology. With the beginning of Phase 3, all malicious activities suddenly stopped. This indicates that *Bob* was regularly assigned the five prefixes by *Alice* in the context of a provider-to-customer business relationship.

Phase 4: *Back to Normal* On March 20, 2013, *Bob* withdrew all announcements of the more specific prefixes, resulting in the same initial situation as in Phase 1, where the whole prefix `A.B.0.0/16` was announced by *Alice* only.

Given these findings, state-of-the-art analysis techniques, e.g. based on assumptions on BGP spectrum agility [32] or the correlation of spam and routing anomalies [185], would tend to infer a malicious hijacking attack. All evidence presented so far, especially the strong correlation between both the control-plane and data-plane observations, indeed support this conclusion.

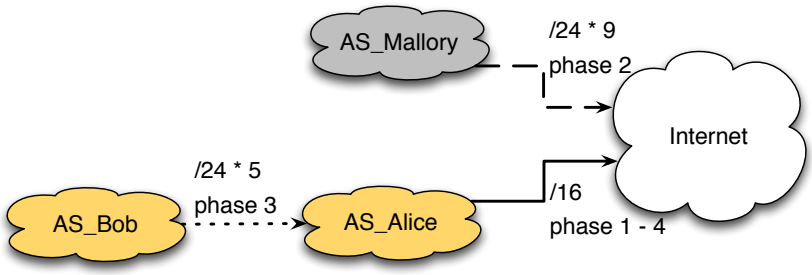


Figure 10.17: AS topology derived from BGP for the *Bulgarian Case*.

Counter-evidence

Despite the compelling evidence for a malicious hijacking incident, we can still draw on another data source. By passing the event to our IRR filter, we are able to find significant evidence against a hijacking attack. With our graph database, we analyze more than one year of archived IRR database snapshots in order to identify the legitimate holders of the involved prefixes over time. To this end, we search for `ROUTE` objects that document a relationship between an IP prefix, i.e. `INETNUM` objects, and an origin AS, i.e. an `AUT_NUM` object. As a matter of fact, *Alice* carefully maintained such `ROUTE` objects in the RIPE database throughout all four phases described above.

Figure 10.18 gives an overview of all relevant `ROUTE` objects that we found in our graph database. We observe the first three `ROUTE` objects related to *Alice's* prefixes on December 4, 2012. Their *origin* attributes reference *Mallory*, and the creation time corresponds to the first BGP announcements. This clearly indicates that according to the RIPE database, *Mallory* was authorized to use these prefixes. We further learn that corresponding objects were created for all of *Mallory's* and *Bob's* subsequent announcements, and that their dates of appearance perfectly match the activities in BGP. All of these objects were maintained by *Alice*. Since we generally assume that an attacker is incapable to alter the IRR databases at will (and that he has no access to his victim's maintainer account), we must conclude that *Alice* delegated all nine prefixes to *Mallory* by choice on December 4, 2012, and reassigned some of them around February 3, 2013 to *Bob*.

We further analyzed the *descr* attributes of all database objects, i.e. free-text description fields, and even found some weak evidence for a relationship between *Mallory* and *Bob*. In Phase 2, all fields were set to `BG-{XX}-{N}`. `BG` presumably indicates Bulgaria, whereas `N` corresponded to the third byte of each of the prefixes. More importantly, `XX` represented the initial letters of *Bob's* company name. After reassignment in Phase 3, the description changed to *Bob's* full company name.

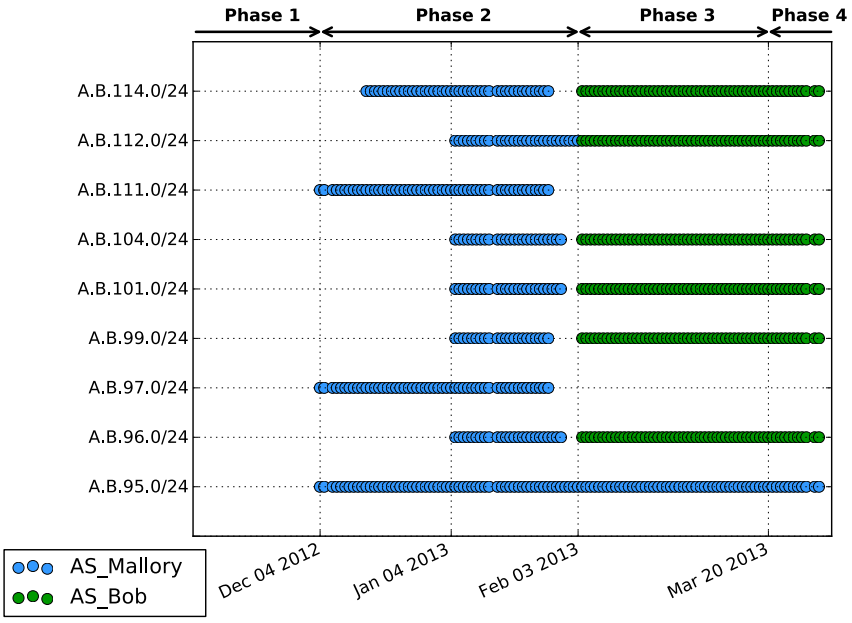


Figure 10.18: Analysis of `ROUTE` objects in the RIPE database for the *Bulgarian Case*.

Finally, we contacted *Mallory's* upstream provider and learned that *Mallory* requested to announce her rented prefixes in the context of a regular business case. After receiving complaints, the upstream provider cancelled *Mallory's* contract and terminated corresponding BGP sessions.

Discussion

Our case study resulted from an effort to confirm findings of related work on the root causes of BGP hijacking events. Previous studies [32, 185] reported a correlation between BGP hijacking attacks and spamming activities. At a first glance, the same appeared to be true for the *Bulgarian Case*. For our analysis, we combined a variety of orthogonal data sources including spam traps, IP blacklists, traceroute measurements, and traffic flow data. A strong temporal correlation between suspicious BGP announcements, i.e. subMOAS events, and malicious activities like spamming and phishing apparently pointed us to a hijacking attack with malicious intent.

Even though we have accumulated a series of converging evidence incriminating one of the actors to be involved in a hijacking attack, our HEAP framework eventually uncovered that the affected networks were instead rented for abuse. A forensic analysis carried out with our IRR graph databases revealed thorough documentation of all routing changes.

In this respect, the alleged victim granted the spammer the right to use for corresponding networks. While clearly being abused for illegal activities, we conclude that no attack on these networks took place at the routing level.

10.5 Lessons Learned

In this section, we evaluated our HEAP framework to assess the legitimacy of suspicious routing events. With a thorough analysis of day-to-day anomalies, we established an encouraging base line for practical validation of hijacking alarms: We legitimized 56.93% of these events. In our case study, we further narrowed down the search space for practical hijacking attacks by focusing on networks that host the top one million web sites. Our ability to legitimize 81.15% of corresponding events indicates that our methodology performs even better for such popular networks. These networks may be at higher risk of being attacked due to their good reputation in whitelists. With an analysis of publicly reported hijacking incidents, we demonstrated great practical benefits of our system by identifying nearly 10% of the alarms as false positives. HEAP is thus ready to interface with current and future detection systems of fellow researchers to receive and assess their alerts.

Based on our evaluation, we arrived at the following conclusions. First, data obtained from IRR databases, albeit possibly incomplete, is highly useful to assess hijacking alarms in practice. Second, the topology reasoning technique proves to be of equally high effectiveness. Last, but not least, active scans greatly support a reliable assessment of hijacking alarms. The applicability of this approach is remarkably high, which, more importantly, relates to a huge set of SSL/TLS-enabled hosts that remained stable throughout our experiments. We consequently encourage network operators to "opt-in" to HEAP by simply setting up HTTPS servers with unique SSL/TLS keys in their networks—these would be automatically found by our ground truth scans and incorporated into HEAP—ready to be used for validation scans in case of an alarm. In our evaluation, we further observed striking differences in the deployment of SSL/TLS, which led to improvements to regular ground truth scans in the future.

By studying a real abuse incident, our HEAP framework proved again to be highly beneficial to assess the legitimacy of routing anomalies. We thoroughly evaluated a hijacking alarm in which routing anomalies coincided with malicious activities originating at the affected prefixes. With the same body of evidence, previous work might have concluded the existence of a malicious hijacking case. Despite the compelling evidence for an ongoing hi-

jacking attack, we were able to infer a business relationship between the affected parties, suggesting that spammers legitimately rented IP space instead. We consequently suggest that previous cases should again be put to test. This practical example further illustrates that with our HEAP framework, we can avoid drawing conclusions too quickly based on a limited set of evidence skewed towards one verdict or the other. This fact is of particular interest to avoid misattributing attacks launched from hijacked IP space, especially when responding with legal actions.

11 CHAPTER ELEVEN
Evaluation of *CAIR*

NOTE *This chapter contains prior publication.*

Submitted to **IEEE/ACM Transactions on Networking (TON), 2016.**
Sections 11.1 to 11.3 are based on previous work [9]. Section 11.1 provides additional statistics and a more detailed comparison to ground truth data. Additional diagrams are added to Sections 11.2 and 11.3. Individual subsections are slightly rearranged to improve the reading fluency.

SUMMARY In Chapter 8, the author presented CAIR, a system to rigorously study Internet routing based on deterministic finite-state automata. We evaluate the CAIR framework by analyzing publicly available BGP data over the last seven years. To this end, we study its performance properties in Section 11.1 and compare them to those of common network graphs. We further apply our technique to detect ongoing path manipulations in BGP and reveal 22 critical interception incidents so far unknown to the public. In Section 11.2, we explain details of the approach along two case studies, including a ground truth attack presented at DEFCON [171]. To further demonstrate the practical relevance of CAIR, we derive measures to study the importance of ASes with respect to global routing and evaluate their characteristics over time. In Section 11.3, we compare the results to a sudden and radical routing change during a recent route leak of Telekom Malaysia (2015) [30]. Based on our findings, we outline an approach to reliably detect such incidents in the future.

11.1 Overall Results

We introduced the CAIR framework as a powerful tool to study Internet routing (see Chapter 8). Now we deploy CAIR in practice and analyze seven years of BGP data, with particular focus on its performance characteristics and its capability to detect interception attacks. To this end, we construct a series of route automata based on BGP routing table exports from the RouteViews Oregon [222] collector. Our analysis period covers August, 2008 to November, 2015, divided into intervals of two weeks (i.e. 174 RIB exports with a total of ≈2.4B entries).

11.1.1 Performance Properties

To study the performance of CAIR in more detail, we construct a route automaton using the RouteViews RIB export on November 1, 2015. We show that the required resources for the route automaton are competitive with graphs and can even decrease with more routes being added as an effect of minimization. We further study the correlation between automata size and Internet growth.

States and Transitions

The RouteViews collector peers with 41 ASes that represent our set of observation points $P_{ore} \subset \Sigma_{AS}^*$. These peers advertise 600,216 IP prefixes $p \subset \Pi$ via 2,875,026 distinct AS paths $w \in \Sigma_{AS}^*$ that comprise 52,396 individual ASes $o \in \Sigma_{AS}$. The full set of routes known to the collector is thus given by $\mathscr{L}_{ore} \subset \{wp \mid w \in \Sigma_{AS}^*, p \subset \Pi\}$ and consists of $|\mathscr{L}_{ore}| = 22,303,775$ routes. The corresponding route automaton M_{ore} accepts the finite route language $\mathscr{L}_{ore}$. It utilizes $|Q_{ore}| = 302,598$ states and $|\delta_{ore}| = 10,355,671$ transitions to hold the entire RIB export. Further details are provided with Table 11.1. It is worth mentioning that CAIR clearly outperforms a trie [217] holding the same information, which would require 73.71 times as many nodes and 2.15 times the amount of transitions. Our vanilla implementation in Python needs 18.4 minutes to parse the input data and 99.4 minutes to create the automaton; subsequent updates can be applied in real-time. An optimized C++ version for operational deployment is part of our future work.

	August 10, 2008	November 1, 2015
RIB entries $\lvert\mathscr{L}_{ore}\rvert$	**10,686,819**	**22,303,775**
IP prefixes $\lvert\{p \subset \Pi\}\rvert$	276,706	600,216
AS numbers $\lvert\Sigma_{AS}\rvert$	29,203	52,396
Unique AS paths[1] $\lvert\{w \mid wp \in \mathscr{L}\}\rvert$	1,421,062	2,875,026
Average AS path length[1] $\frac{\Sigma\lvert w\rvert}{\lvert\{w \mid wp \in \mathscr{L}\}\rvert}$	4.11	4.24
Route automaton M_{ore}		
CAIR states $\lvert Q_{ore}\rvert$	168,184	302,598
CAIR transitions $\lvert\delta_{ore}\rvert$	4,939,314	10,355,671
Graph nodes $\lvert N_{ore}\rvert$	305,909	652,612
Graph links $\lvert L_{ore}\rvert$	339,515	725,425

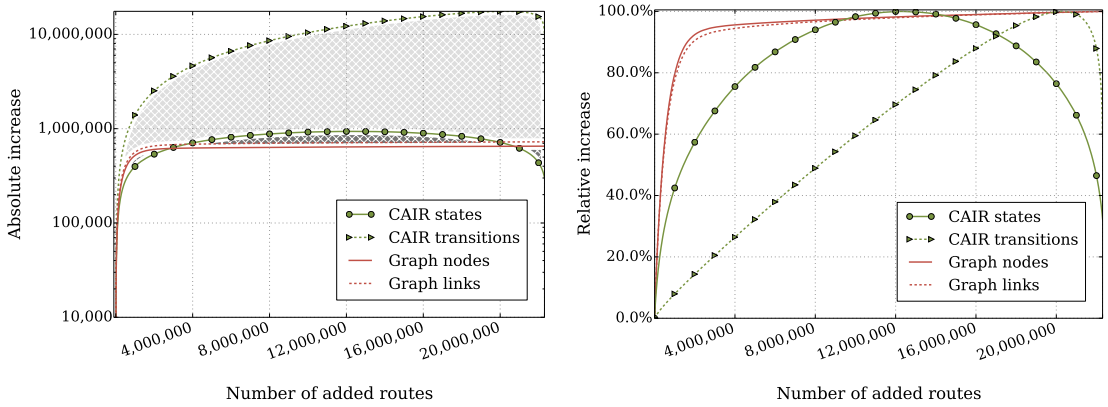
Table 11.1: RIB information content.
[1]Unprepended AS paths only.

Comparison to Network Graphs

We define a network graph $\mathscr{G} = (N, L)$ as a set of nodes $N = \Sigma_{AS} \cup \mathscr{P}(\Pi)$ and a set of links $L \subset \Sigma_{AS} \times (\Sigma_{AS} \cup \mathscr{P}(\Pi))$, which yields a total number of $|N| = 652,612$ nodes and $|L| = 725,425$ links for the collector studied above. In comparison, CAIR requires only 46.37% of states but 14.28 times more transitions to represent the full routing table. Taking into account that CAIR also holds all observed routes on top of individual AS links as represented by the graph, its efficiency is remarkable.

A common measure used in conjunction with graphs is given by the number of incoming and outgoing links per node (further refered to as *indegree* and *outdegree*). For the network graph, this metric yields 2.29 respectively 14.62 links on average. Our route automaton, in contrast, is built upon two particular states $q_0 \in Q$ and $q_f \in Q$ to enter and leave the routing model. Since all outgoing transitions of q_0 eventually lead to incoming transitions in q_f, both the average indegree and outdegree evaluate to the same value of 34.22 transitions. As a consequence, CAIR provides a significantly richer connected data structure as compared to network graphs due to its larger number of transitions.

Figure 11.1a quantifies the evolving number of states and transitions in CAIR and the network graph while consecutively importing new routes. CAIR requires fewer states but needs more transitions to implement its expressiveness. The minimization approach in CAIR, however, introduces self-adaptive optimization. To clarify this, Figure 11.1b shows the number of states (transitions) relatively to their maximum number during construction. While the graph data structure created ≈90% of the required objects already after importing



(a) Absolute growth of data structures.

(b) Relative growth of data structures.

Figure 11.1: Performance characteristics of CAIR in comparison to network graphs.

10% of all routes in the input data set, CAIR grows slower. This nicely illustrates that the graph, in contrast to CAIR, does not learn additional information with additional routes observed. More importantly, the amount of states decreases in CAIR significantly after adding 55% of the routes. This implies that adding further routes to the automaton can actually reduce its size, which is due to the minimization process getting more effective on larger volumes of input data. This finding is of particular interest with respect to possible applications of CAIR in real-time routing analysis.

Resource Requirements Depending on Internet Growth

We now study retroactively the routing system at particular points in time. We use the RouteViews data as described above to construct different route automata in an interval of two weeks over the period of August, 2008 till November, 2015. Table 11.1 presents the absolute number of required resources. We see that the number of RIB entries has roughly doubled (+108.70%) in seven years. The same holds true for the number of advertised IP prefixes (+116.92%). The growth of both CAIR and the network graph are in line with these findings. The number of states for the route automaton increased by 79.92% and the number of transitions by 109.66%. As for the network graph, we see an increase of 113.34% and 113.67% for the number of nodes, and links respectively.

Figure 11.2 further shows the relative evolution of the RIB information content compared to resource requirements in CAIR and the corresponding graph data structure. It is clearly visible that the graph depends mainly on the number of IP prefixes, whereas CAIR depends on the number of RIB entries as those provide additional routing insights.

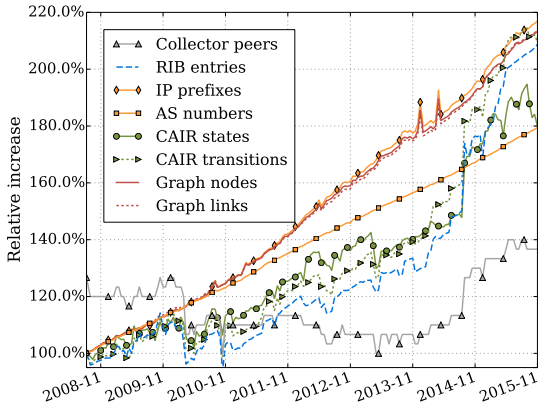


Figure 11.2: RIB Information content.

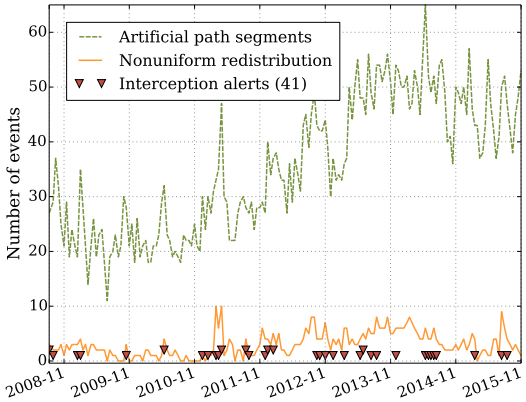


Figure 11.3: CAIR interception alerts.

11.1.2 Interception Incidents

In Subsection 8.2.3, we presented a detection scheme for interception attacks, in which an adversary falsely attracts traffic while maintaining a backhaul path to its victim in order to relay eavesdropped packets. By applying this technique to the aforementioned seven years of archived BGP data, we identified 6,171 artificial path segments $w_v^a \in \Sigma_{AS}^*$ (for notations see Section 5.2.2). For 527 of them, we found a corresponding segment $w_v^d \in \Sigma_{AS}^*$ that evidenced nonuniform redistribution of BGP updates. Our detection scheme finally raised 41 alerts for (strict) subprefix hijacking attacks. Table 11.2 presents the results. In Figure 11.3, we plot our findings over time. Note that one of the identified events lasted for 8 weeks, i.e. accounts for four subsequent alerts. Additional five events were observed twice. This leaves us with a total number of 32 distinct cases.

Sanitizing: Exclude False Positives

Recall that our search pattern for interception attacks is based on the observation that nonuniform redistribution of BGP announcements should only take place at a victim v itself or at his direct upstream ISPs. We consequently search for artificial path segments of length $|w_v^a| \geq 3$. This assumption breaks, though, if the victim operates multiple ASes [226] that are consecutively visible in an AS path. We carefully analyzed the 32 interception alerts raised by CAIR using the WHOIS system (see Subsection 3.3.1 and Subsection 7.2.1 for details). Note that some of these events date far back to the past. We consequently utilized archived WHOIS data for the respective dates. Altogether, we were able to identify 9 sibling cases, which we exclude from further investigation. The remaining 23 alerts are marked as suspicious.

August 10, 2008 – November 1, 2015	total	in %
Artificial path segments w_v^a	**6,171**	**100.0%**
Nonuniform redistribution $w_v^d \neq w_v^a$	527	8.54%
Interception alerts[1] $p'_v \subset p_v$	41	0.66%
Unique alerts (victims)	**32**	**0.52%**
Manual inspection		
Sibling ASes (victim)	-7	0.11%
Sibling ASes (upstream)	-2	0.03%
Alerts after manual inspection	**23**	**0.37%**

Table 11.2: Interception alerts raised by CAIR.

[1]Concerning subprefix announcements only.

Substantial Alerts

We present further details on the remaining 23 substantial alerts in Table 11.3. Entries that are highlighted in grey represent the upstream AS $s \in \Sigma_{AS}$ that is used by an attacker to redistribute forged BGP updates. The corresponding AS neighbor $t \in \Sigma_{AS}$ represents the attacker's second upstream, which is used to uphold the backhaul path. The underlined parts of the AS paths show the artificial segment w_v^a.

In 13 cases, w_v^a is of length 3 (plus the attacker's two upstreams and possibly further intermediate ASes), while 6 cases exhibit a segment length of 4 AS hops. The 4 remaining incidents show longer segments. To better grasp the details of the incidents, we checked AS names in the WHOIS system.

Comparison with Ground Truth

Beyond the well-known demonstration of an interception attack at DEFCON [171], we have two further sources of ground truth for publicly reported incidents. First, *Dyn Research* published two cases of interception attacks during 2013, namely the Belarusian and Icelandic traffic diversions [227]. Second, a heuristic detection scheme to detect possible interception attacks based on so-called next-hop anomalies reported a total of 41 incidents on four days in late 2006 [169]. For the resulting 44 ground truth events, we obtained the corresponding routing tables, constructed route automata, and applied our own detection scheme. CAIR reported one alert for the time frame studied in [169], which we attribute to a false positive case of sibling ASes, though. For the events presented in [227], we could not find any evidence. In constrast, CAIR did raise an alert for the DEFCON attack.

Date	**Forged AS path** / nonuniform route segment					**Victim** / country / company
[1,2]2008/08/10	AS26627	AS4436	AS22822	AS23005	AS20195	United States, *Sparkplug Las Vegas*
2009/02/01	AS3303	AS1299	AS701	AS3491	AS37004	AS30988 Nigeria, *IS InternetSolutions*
2009/10/15	AS3561	AS7018	AS4837	AS4808	AS17431	AS17964 China, *Beijing Network Technologies*
2010/05/15	AS2914	AS3549	AS3356	AS23148	AS20080	AS1916 Brasil, *Rede Nacional de Ensino*
2010/12/15	AS34984	AS12301	AS3549	AS9121	AS12794	Turkey, *Akbank*
2011/01/15	AS9002	AS21230	..(+2)..	AS3356	AS9121	AS44565 Turkey, *Vital Teknoloji*
2011/03/01	AS4134	AS40633	..(+4)..	AS6453	AS9299	AS18223 India, *Capital IQ Information Systems*
2011/04/01	AS3549	AS5391	AS25144	AS42432	AS8670	Bosnia, *University of Sarajevo*
2011/04/01	AS3549	AS10026	AS9957	AS10036	AS18334	South Korea, *Gyounggidongbu Cable*
2011/08/15	AS1273	AS1299	AS3491	AS20485	AS8402	Russia, *Vimpelcom*
2011/12/01	AS6762	AS31133	AS12695	AS34123	AS28738	Russia, *InterLAN Communications*
2011/12/15	AS702	AS701	AS3549	AS21371	AS49669	United Kingdom, *Cognito*
2012/09/15	AS2828	AS1299	AS9498	AS58459	AS4613	Nepal, *Mercantile Office Systems*
2012/11/01	AS30496	AS11427	AS7843	AS6461	AS33481	AS40610 United States, *Digital Passage*
2012/12/15	AS30496	AS11427	AS7843	AS6461	AS33481	AS21854 United States, *Digital Passage*
2013/02/15	AS1273	AS2914	AS8928	AS5391	AS57888	Croatia, *Telesat*
2013/06/01	AS1299	AS6663	..(+3)..	AS6939	AS197043	AS197890 Germany, *Megaservers*
2013/07/15	AS3356	AS1299	AS6663	AS41571	AS48828	Romania, *Carosystem*
[2]2013/08/15	AS3549	AS3491	AS12880	AS43343	AS21341	AS25306 Iran, *Institute IsIran*
2013/12/01	AS1299	AS9498	AS12880	AS41881	AS51411	Iran, *Toos-Ashena*
2015/02/15	AS1299	AS52320	AS16735	AS28284	AS262353	Brasil, *Marcelo Bonini*
2015/07/15	AS3356	AS209	..(+2)..	AS721	AS27066	AS747 United States, *US Army ISC*
2015/08/15	AS46450	AS1299	AS3356	AS6079	AS55079	United States, *Third Gear Networks*

s: ASN t: ASN w_v^a: ASN ... ASN

Table 11.3: Remaining interception alerts after manual inspection.
[1]Public demonstration of an interception attack at DEFCON [171].
[2]Studied more closely in Section 11.2.

11.2 Case Study: "Interception Alerts"

The DEFCON attack [171] was a well-known experiment to demonstrate interception attacks in the Internet to the research community. All details on the attack are publicly available [170] and thus provide perfect ground truth for validating CAIR. In a subsequent step, we evaluate a real-world incident reported by CAIR that targeted an Iranian AS.

Ground Truth: The DEFCON Attack (2008)

Figure 11.4 shows the route automaton that corresponds to the alert raised by CAIR for the DEFCON attack. The victim `AS20195` (*Sparkplug*) initially advertises four distinct

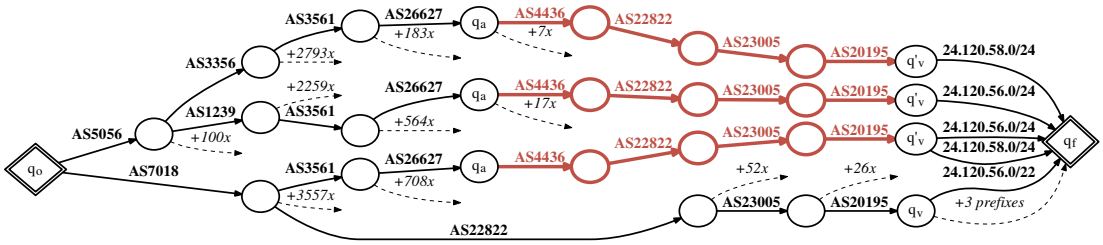


Figure 11.4: Route automaton for the *DEFCON Attack* (AS20195). August 10, 2008.

prefixes including 24.120.56.0/22. During the attack, two more specific prefixes (namely 24.120.56.0/24 and 24.120.58.0/24) appear to originate from this AS. With our automaton, we can identify the highlighted artificial path segments. Note that none of the corresponding states shows any other outgoing transitions in contrast to the valid segment. We can further locate the attacker at q_a (see Section 8.2.3). It is either AS26627, or a common customer of this AS and AS4436. Our observations fully comply with the public, verified information about that DEFCON incident [170]. The attacker was located at AS4436.

The Tehran Incident (2013)

Another interception alert of particular interest as reported by CAIR affects AS25306 (*INSTITUTE-ISIRAN IsIran*). We refer to this event as the *Tehran Incident*. Compared to the DEFCON attack, the analysis of this incident is more challenging for two reasons. First, this incident has not been discussed in the public so far. Second, the attack took place in 2013, which makes verification based on additional data sets difficult.

Figure 11.5 shows the corresponding details. On August 15, 2013, AS25306 advertised eight prefixes including 81.28.23.0/19. At the same time, we observe an artificial path $w_v^a \in \Sigma_{AS}^*$ of length 5 to the subprefix 81.28.37.0/24. Within our route automaton we are able to localize the attacker at the state q_a. Our technique yields that he must be a common customer of both AS3549 (*Level3*) and AS3491 (*Beyond the Network America (BTNA)*).

To substantiate our observation, we searched operator mailing lists and online discussion platforms for further evidence. The incident itself was not reported. However, we may speculate about potential customers of the ISPs involved, who maybe benefit from an interception. Level 3 is a *tier 1* ISP with a quite diverse set of customers. BTNA seems to have a particular reputation as a spammer-friendly service provider. Note that spam activities also misuse BGP [32] to prevent backtracking. Although interception itself has not been documented so far in the context of spamming, an operator of a spamming network could be

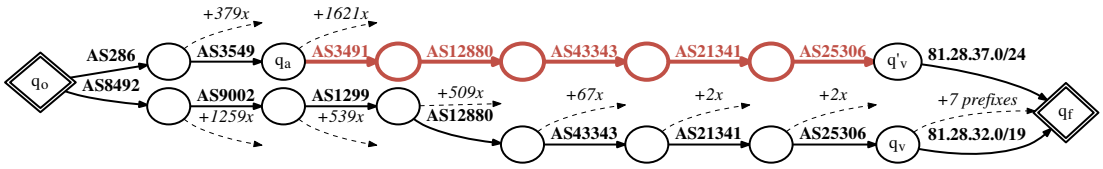


Figure 11.5: Route automaton for the *Tehran Incident* (AS25306). August 15, 2013.

interested to implement this attack: Any mitigation mechanism at the receiver side that requires the correct end point (e.g., callback verification) can be fouled using the backhaul path. Furthermore, we found an abuse report [228], where BTNA apparently served as upstream for a hijacking attack during an other event.

To conclude, we do not have a proof that the Tehran Incident was indeed an interception attack. However, based on manual investigation we found strong evidence that the alarm triggered by CAIR was correct.

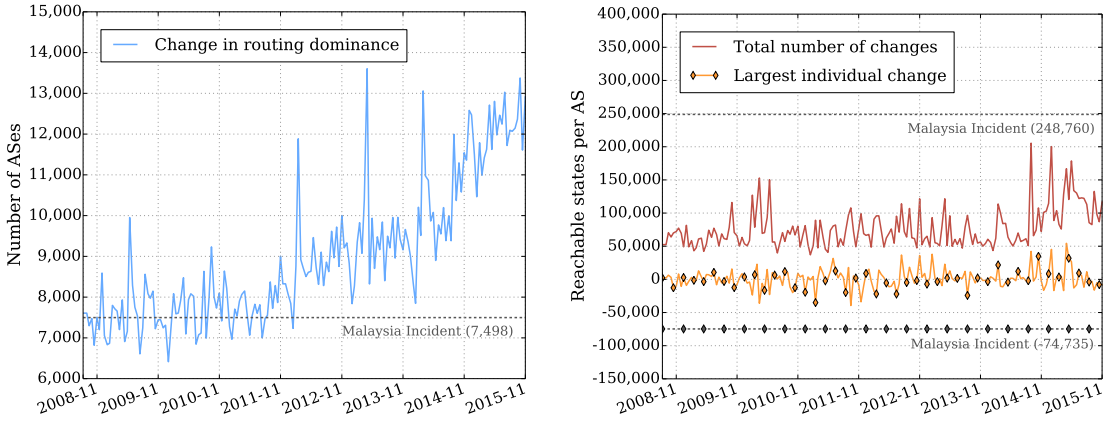
11.3 Case Study: "The Malaysia Route Leak"

To further demonstrate the expressiveness of CAIR, we derive a simple yet effective metric to assess route leaks. In its most basic form, a route leak is given by a multi-homed AS, in the following called the *originator*, who (accidentially) re-advertises its full routing table into BGP. Upstream ISPs often prefer customer routes [229] and thus redistribute the corresponding BGP announcements. The result is a major shift in global Internet traffic such that packet flows are now redirected towards the originator. Although a route leak leads to significant re-routing activity in BGP, it leaves individual AS links intact. As a consequence, network graphs are ill-suited for the detection and analysis of such events.

A Measure for Routing Dominance

In order to asses the routing changes imposed by route leaks, we introduce a metric for *routing dominance*. Recall that each state $q \in Q$ in our minimized automaton represents an equivalence class, i.e. a set of partial routes $\vec{\mathscr{L}}(q)$ accepted by q (see Subsection 8.2.1). We extend this abstract concept to ASes $u \in \Sigma_{AS}$ such that $\vec{Q}(u)$ yields the set of all states that are reachable via transitions labeled with u as given by

$$\vec{Q}(u) \;=\; \{q \in Q \mid \delta^*(\vec{q}, uw) = q,\ \vec{q} \in Q,\ w \in \Sigma_{AS}^*\}\,.$$



(a) Routing changes by affected ASes.

(b) Routing changes by total amount.

Figure 11.6: Analysis of routing changes with CAIR.

The size of $|\vec{Q}(u)|$ implicitly represents the number of (partial) routes over AS u. The larger this set is for a given AS, the more of its advertised routes are redistributed by its peers, i.e. the more it dominates routing in BGP. Contrary to intuition, this metric does not yield a particular high rank for the peers of our BGP collector. They exhibit an average rank of 3,808. The top-5 ASes ranked by $|\vec{Q}|$ are `AS2914` (*NTT*), `AS3356` (*Level3*), `AS3257` (*Tinet*), `AS1299` (*Telia*), and `AS174` (*Cogent*), which all are prominent *tier-1* providers.

Regular route updates and route leaks

We evaluate regular changes in $|\vec{Q}|$ for individual ASes over the course of seven years based on our RouteViews data set and compare the results to a recent route leak caused by `AS4788` (*Telekom Malaysia*) on June 12, 2015 between 08:00 and 10:00. On average, we observe 9,019 ASes that experience any change in $|\vec{Q}|$ during the periods of two weeks. Note that the average number of ASes that newly appear in BGP—and thus regularly change $|\vec{Q}|$ for their upstream ISPs—is 364, i.e. significantly lower. Taking into account this growth in BGP participants, we obtain a corresponding average of 19.61% of ASes with changes in $|\vec{Q}|$. For the Malaysia route leak, we observed a change in routing dominance for 7,498 ASes (14.67%) in an interval of less than two hours. Figure 11.6a illustrates these observations.

Figure 11.6b further shows the total amount of changes in $|\vec{Q}|$ over all ASes. We observe an average of 76,785 states that are added to or removed from the set $\vec{Q}(u)$ of any AS $u \in \Sigma_{AS}$. In contrast to that, the Malaysia route leak led to a total change in reachability of 248,760 states. Such a significant and abrupt re-routing is unprecedented during nor-

mal operations. In addition, Figure 11.6b illustrates the maximum and minimum changes imposed to a single AS. The largest (regular) individual increase of 54,402 newly reachable states over two weeks attributes to AS3257 (*Tinet*). For AS2914 (*NTT*), we find the largest decrease of 39,442 states. During the route leak, in contrast, we can observe that a single AS, namely AS577 (*Bell Canada*) loses reachability of 74,735 states (-98.57%) within the duration of the event. At the same time, the size of $|\vec{Q}|$ for Telekom Malaysia increased by 32,621 entries (+2,239%). Such a sudden significant increase in global routing dominance is unlikely the result of a regular change in peering agreements.

The Malaysia Route Leak (2015)

The incident started on June 12, 2015, at 08:40, and began to gradually settle down about two hours later. We split the event in four intervals and consequently utilize four route automata to study this event in retrospective. For T_2–T_3 (08:00–10:00), we evaluate the top-10 ASes that exhibit the highest absolute change in reachable states $|\vec{Q}|$. We compare our results to the as-is state before and after the event, i.e. at T_1 (06:00) and T_4 (12:00). Note that the overall size of the automata hardly changed during the time frame of the route leak (-3.96% of states and -10.47% of transitions). Table 11.4 shows further results. For the sake of completeness, we also analyzed corresponding network graphs, which changed even less (-0.92% of nodes and -1.23% of links).

We already observed a remarkably high shift in globally dominating routes that lies well above the average during normal operations. We see a vast increase in reachable states $|\vec{Q}|$—which notably correlates to inbound traffic—for the originator AS4788 and correspondingly for his upstream ISPs AS3549 (*Level3*) and AS6695 (*DE-CIX*). Note that the latter does not provide upstream connectivity under normal circumstances. Its high increase in dominant routes is rather an artifact: Telekom Malaysia peers with the public route servers at DE-CIX, which redistribute routes from 456 connected ASes to a total of 14,542 different ASes. Apparently, AS4788 leaked these routes such that the AS path of corresponding BGP updates comprised AS6695. This is surprising since route servers at DE-CIX Internet Exchange Point operate transparently and effectively hide their own AS (see Subsection 5.2.2).

The largest decrease in $|\vec{Q}|$—both in relative and absolute terms—attributes to AS577. A similar observation can be made for AS1267 (*Wind Telecomunicazioni*) and also to some extent for AS174 (*Cogent*). Since the originator's routes are globally prefered during the incident, these ASes consequently lose a major part of their inbound traffic in exchange for an

			ongoing route leak		
Top-10 ASes		06:00 (T_1)	08:00 (T_2) – 10:00 (T_3)		12:00 (T_4)
	$u \in \Sigma_{AS}$	$\lvert\vec{Q}_{T_1}(u)\rvert$	$\lvert\vec{Q}_{T_3}(u)\rvert - \lvert\vec{Q}_{T_2}(u)\rvert$		$\lvert\vec{Q}_{T_4}(u)\rvert$
1.	AS577	75,811	-74,735	(*-98.57%*)	75,660
2.	AS4788	**1,555**	**+32,621**	(**+2,239%**)	**1,412**
3.	AS1267	32,514	-32,218	(*-99.11%*)	32,541
4.	AS174	94,702	-16,770	(*-17.73%*)	93,905
5.	AS3549	**16,455**	**+11,523**	(**+71.1%**)	**13,359**
6.	AS3356	101,547	-4,522	(*-4.45%*)	101,262
7.	AS6453	72,213	-3,965	(*-5.48%*)	71,918
8.	AS6695	**713**	**+3,782**	(**+531%**)	**719**
9.	AS2914	107,226	-3,733	(*-3.48%*)	106,862
10.	AS1299	117,673	-3,416	(*-2.90%*)	117,387

Most affected: AS577 Bell, AS1267 Wind, AS174 Cogent
Propagators: AS4788 Tel. Malaysia, AS3549 Level3, AS6695 DE-CIX

Table 11.4: ASes with newly (un-)reachable states. Malaysia Route Leak, June 12, 2015.

increase in outbound traffic towards AS4788. Note that only a smaller number of all ASes (14.67%) propagate routes towards AS47888; the greater part (85.33%) is affected outbound, i.e. only receives corresponding routes. Within two hours after the event, routing converged back to its original state (see right column in Table 11.4).

Towards a Reliable Detection Scheme

With CAIR, we are able to precisely identify the role of each party involved in a route leak. We already showed that only a moderate number of ASes is directly affected by re-routing, which in turn can be classified into groups that either gain or lose in reachable states $|\vec{Q}|$. The highest absolute increase in reachable states is observed for the originator of a leak followed by his upstream ISPs. Hence, we are able to reliably identify both the offender and *catalyst* upstreams, which unwittingly propagate the leak. Note that the measure $|\vec{Q}|$ directly correlates to the amount of traffic attracted by an AS. As a consequene, we can further assess the impact of a route leak on individual ISPs.

Based on our observations, we can easily derive a threshold to detect emerging route leaks as indicated in Figure 11.6b. CAIR can then be used to monitor BGP for route leaks in real-time in order to provide live data that can support quick mitigation. By studying more incidents from the past, we might also derive safety measures to prevent upcoming route leaks. This is part of our future work.

11.4 Lessons Learned

Commonly, graph-based data models are used to represent the Internet topology from a given set of BGP routing tables but fall short of explaining policy contexts. As a consequence, routing anomalies such as route leaks and interception attacks cannot be explained with graphs. To cope with this situation, we introduced formal languages to represent the global routing system in a rigorous model. Our CAIR framework translates BGP announcements into a finite route language that allows for the incremental construction of minimal route automata. This novel data structure preserves route diversity and is well-suited to detect routing anomalies. In this respect, we improved on state-of-the-art by providing formalized anomaly patterns to search our route automata for abnormal routing behaviour. In practical experiments, we analyzed publicly available BGP data over the last seven years. We showed the great potential of CAIR to study routing in general, and confirmed that it outperforms common data structures such as graphs and tries in terms of expressiveness.

In our evaluation, we focused on two intricate anomalies. First, we addressed the challenge of detecting interception attacks. We validated our approach with well-known incidents and identified 22 so far unknown cases of interception. We learned that the affected ASes mostly belong to the R&D sector and to the medium-sized ISP business. Second, we studied (ab)normal routing changes, i.e. so-called route leaks, and gained insight into customer ASes that erroneously advertise transit. Our technique is suitable to reliably detect route leaks and provides a variety of information to analyze such incidents. With CAIR, we can identify the actual offender as well as catalyst ISPs that amplify the route leaks. In addition, we are able to estimate the impact on arbitrary ASes with respect to changes in their incoming and outgoing traffic volumes.

12 CHAPTER TWELVE
Evaluation of *PHEW*

NOTE *This chapter contains prior publication.*

Published in **Proceedings of the International Workshop on Traffic Monitoring and Analysis (TMA), 2015.**
Section 12.1 is based on previous work [6]. No substantial changes are made.

Published in **ACM SIGCOMM Computer Communication Review (CCR), 2013.**
Section 12.2 is based on previous work [4]. No substantial changes are made.

SUMMARY In Chapter 9, the author presented PHEW, a framework to detect abandoned Internet resources that are vulnerable to hidden takeover attacks (refer to Subsection 5.2.3). When corresponding DNS names expire, an attacker is enabled to hijack these resources both from an administrative and operational point of view. In Section 12.1, we use the PHEW system to identify such abandoned Internet resources and learn that a total of 73 /24 networks and 7 AS numbers are currently at risk. A forensic case study on a real incident illustrates the risk potential in great detail. In Section 12.2, we analyze a malicious case of AS hijacking that was carried out in order to send spam. Our findings provide unique insight into the attacker's proceeding and evidence that the threat is real.

12.1 Overall Results

In Chapter 9, we derived an activity metric to assess the risk potential for individual Internet resources. Our metric correlates information about resources extracted from IRR databases with DNS domain expiration dates queried via the WHOIS system (refer to Subsection 3.3.1). A complete picture of the activity of these resources is obtained by taking into account long-term archived BGP tables. In the following, we briefly summarize our findings.

So far, we learned that 65 groups of Internet resources, each maintained by a single entity, reference expired DNS names. Our study on resource activity further indicated that expired groups show lesser activity in IRR databases and BGP than valid groups. By combining these measures, we were able to infer resources that are inactive from both an administrative and an operational point of view. In this respect, 13 of the aforementioned cases did not show any activity for more than 6 months. For the European service region, we find that currently 15 `INETNUM` objects with an equivalent of 73 `/24` IP prefixes and 7 `AUT_NUM` objects, i.e. AS numbers, are effectively abandoned. Note that due to ethical concerns, we do not publish further details about these endangered resources.

The identified resources are highly vulnerable and can be stealthily attacked. Anecdotal evidence exists that such attacks have been carried out in the past. According to [230], for instance, an attacker fraudulently authorized an upstream provider to announce a victim's prefix. In 2003, a large US defence company failed to renew the registration of a DNS domain associated with one of its prefixes. The attacker re-registered that domain to gain control over the mail address that was provided in ARIN's IRR database. After using it to prove prefix ownership to a U.S. upstream provider, the attacker massively sent spam from the hijacked prefix. It took the victim more than two months to notice and counter the attack. Note that the burden of proof for such attacks lies with the victim. In the following, we study a more recent attack in greater detail.

12.2 Case Study: "The LinkTel Incident"

To put our results into perspective, we study a real case of a long-term takeover attack on abandoned Internet resources, which was carried out in order to send spam. So far, no such incident has been studied in detail, in contrast to well-known IP prefix hijacking incidents. We thoroughly investigate the attack using data from both the control and the data plane. Our analysis yields insights into how an attacker proceeded in order to covertly

hijack an abandoned autonomous system, how he misled an upstream provider, and how he abused unallocated address space. Our findings complement a recent study on spammers [32] that connects spamming activities to short-lived prefix hijacking attacks. More importantly, the incident shows that there is a real threat emerging from abandoned Internet resources.

Course of Events

On August 20, 2011 a representative of the Russian ISP *Link Telecom* (AS31733) sent a distress mail [231] to the North American Network Operators' Group (NANOG). The subject of this message was a suspected prefix hijacking attack observed via the upstream ISP *Internap* (AS12812) located in the USA. The author explained that Link Telecom had struggled with the financial crisis and got almost bankrupt, but was now on the verge of recovery due to a new investor. While trying to get their prefixes back online, Link Telecom's operator recognized massive blocking of the company's traffic and learned that all prefixes were listed on Spamhaus [232] spam blacklists. We refer to this event as the *"LinkTel Incident"*.

According to the author, Internap received a forged letter of authorization from Link Telecom on June 9, 2011 and started to redistribute routes for AS31733 and its prefixes. By end of July, 2011, Link Telecom closely investigated the issue, found its prefixes announced via Internap, and complained to this ISP. Internap consequently referred to a valid letter of authorization and refused to take actions. In addition, Link Telecom apparently was contacted by a person claiming ownership of the prefixes in question:

> *"(...) someone named Michael Lindsay contacted us and said it is his network!"* [233]

On August 25, Link Telecom informed Internap's upstream providers (which at that time were *Telia, Cogent, NTT, Global Crossing* and *Tinet*) and started announcing more specific prefixes while redelegating reverse DNS servers. After Tinet (AS3257) and NTT (AS2914) started to filter out illicit announcements on August 29, the attack ended on August 30.

While this story already gives an overview of the attacker's proceeding in general, we are able to disclose the full course of action by studying archived control plane data for the corresponding period of time. We further analyze manually obtained meta data to complement our findings. Finally, the evaluation of data plane information allows us to understand the attacker's intention in great detail.

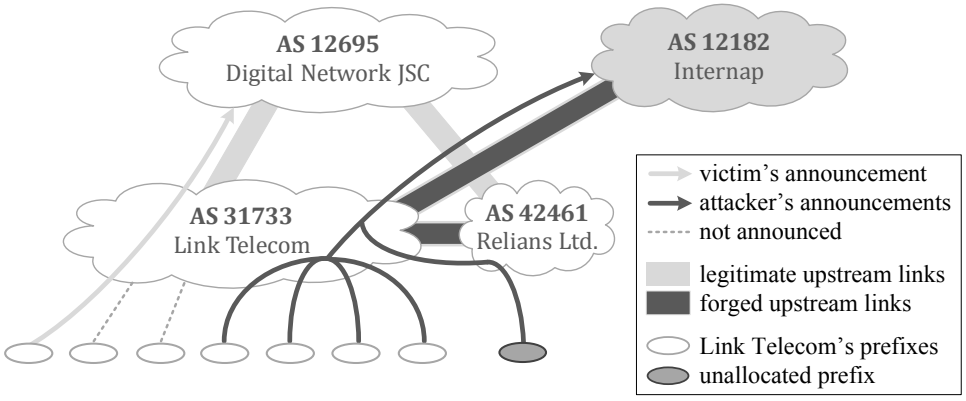


Figure 12.1: Changes in the AS-level topology during the *LinkTel Incident* (2011).

Control-plane Analysis

We evaluated archived data feeds from RouteViews Oregon's BGP router [222] for the time between April 1 and September 30, 2011. To this end, we parsed a total of 17,573 individual dump files containing BGP update messages. Out of 2,633,764,529 messages globally received within this time frame, we extracted 12,751 messages that were related to the *LinkTel Incident*. 409 advertisements originated at the time before the attack, 1,441 route updates could be traced back to the attacker, and 9,449 messages arose in Link Telecom's attempt to regain control over its prefixes. We also found 1,452 withdraw messages, from which 1,411 relate to the regain attempt. The remaining 41 withdraw messages directly reflect actions taken by the attacker. In the following, we describe our findings in detail.

We learned that the attack started on April 15, 2011 with a single announcement of `188.164.0.0/16`. Within eight weeks, the attacker gradually took over further parts of Link Telecom's networks (namely the prefixes `94.250.128.0/18`, `83.223.224.0/19`, and `46.96.0.0/16`). The last announcement was observed on June 9, which is the same day when Internap supposedly received the forged letter of authorization. This finding indicates that Internap redistributed the attacker's route updates for two months without formal authorization. The attacker spared Link Telecom's remaining prefixes `86.59.224.0/17`, `79.174.128.0/18`, and `94.250.192.0/18`, which were never touched by the attacker. Note that the first of these prefixes was partially announced by Link Telecom before and during the whole attack. Figure 12.1 shows the resulting AS-level topology.

On May 12, 2011, the attacker further attempted to announce unallocated address space (`89.145.168.0/21`). For that, a second AS, *Relians Ltd.* (`AS42461`), was hijacked and used to originate routes via the already hijacked `AS31733`. At that time, Relians Ltd. was con-

nected to the same Russian upstream provider as the one employed by the victim, *Digital Network JSC* (AS12695). Figure 12.1 shows the respective relations between all parties involved. We assume that AS42461 was used as a decoy to test announcing unallocated space without risking the ongoing attack. On July 11, the prefix was globally withdrawn. This indicates that the attacker—or someone else along the topological way—suddenly decided to stop routing that prefix. Our finding confirms the result of previous work [166] in that bogus routes to unallocated address space can still leak into the global routing system.

Link Telecom started to recover on August 24 by re-advertising its prefixes at full size, and also via announcements of more specific prefixes on August 29. As a consequence, some of the routes to the hijacked networks were globally withdrawn. On August 30, major upstream providers (including NTT and Tinet) actively withdrew the remaining forged routes. The last traces of the attack finally vanished on September 3, 2011, i.e. 141 days after the first illicit announcement. The attacker never tried to announce more specific prefixes and did not fight back in any other observable way.

Analysis of Meta Data

Beside for BGP data, there are few historical archives concerning Internet-specific information, which makes it difficult to carry out forensic analyses. For instance, the holder of Link Telecom's AS number and prefixes has long changed since the attack. The same is true for the unallocated address space and for the company operating the decoy AS. For the case at hand, however, we were able to obtain an accurate view of events related to the attack by searching the Internet for evidential data. We found a RIPE database dump from June 3, 2011 at the UK mirror service [234], which archives a collection of popular FTP and web sites. This RIPE database dump provided us with the necessary information on resource holders to assess the full course of events. To allow for future analyses, we suggest to continuously archive IRR database dumps within the PHEW framework.

We found further evidence that the attacker deceived Internap of being authorized to advertise Link Telecom's resources. From the report on NANOG's mailing list, we learned that the domain `link-telecom.biz` was taken over by the attacker. The aforementioned RIPE database dump revealed corresponding email addresses associated to Link Telecom's resources, implicitly given in the resources' *changed* attributes. A DNS whois lookup showed that the domain expired on March 11, 2011, and was re-registered 6 hours later by a proxy registrar protecting the buyer's identity. Note again that for incidents lying further in the past, DNS registration data is inaccessible due to the lack of historical archives.

Hijacking Phase		
Mar	12	The attacker immediately re-registers the expired DNS domain `link-telecom.biz` to take over `AS31733` (Link Telecom)
Apr	15	The attacker announces prefix `188.164.0.0/16` originating at AS31733 via AS12812 (Internap)
May	06	The attacker announces prefix `94.250.128.0/18` (via 2x `/19`)
May	12	The attacker takes over `AS42461` (Relians Ltd.) and uses it to announce the unallocated prefix `89.145.168.0/21`
May	28	The attacker announces prefix `83.223.224.0/19`
Jun	09	The attacker announces prefix `46.96.0.0/16`
Jun	09	Internap receives a faked letter of authorization
Productive Phase		
Jul	11	Global withdrawing of unallocated prefix `89.145.168.0/21`
Jul	28	Spamhaus blacklists all remaining hijacked prefixes
Recovery Phase		
Aug	11	Link Telecom sends complaints to Internap
Aug	24	Spamhaus starts to close spam cases
Aug	24	Link Telecom announces all prefixes at full size
Aug	25	Link Telecom sends complaints to upstream ISPs and redelegates reverse DNS
Aug	28	Link Telecom announces more specific prefixes
Aug	29	Link Telecom receives responses to complaints
Aug	30	NTT and Tinet withdraw routes to hijacked prefixes
Sep	03	BGP converges, no further related events

Table 12.1: Full disclosure of the *LinkTel Incident* (2011).
Grayed out entries are not confirmed by our analysis.

The RIPE database dump also revealed that reverse DNS was delegated to name servers with a host name in the re-registered domain. By assigning `{ns1,ns2}.link-telecom.biz` to a malicious host, all reverse DNS queries for the hijacked networks could have been intercepted by the attacker. This implies that the attacker had the ability to pass Forward-confirmed reverse DNS (FCrDNS). In Section 9.2, we discussed that passing such checks is highly beneficial for an attacker with respect to evading spam detection.

Finally, we discovered the exact point in time at which the prefixes first appeared on Spamhaus blacklists. Note that this information is not available anymore due to limited availability of long-term archives. Nevertheless, we found a discussion on the RIPE anti-abuse working group's mailing list [235]. Its subject is beyond the scope of this paper—the

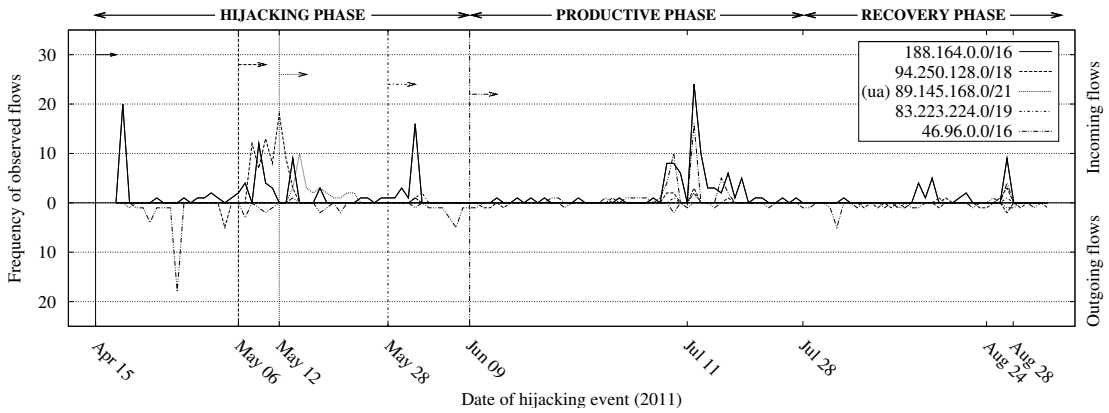


Figure 12.2: Traffic flows in our academic network related to the *LinkTel Incident* (2011).
Vertical lines indicate the specific date of illicit BGP announcements.
The prefix marked with (ua) was unallocated at that time.

initial message, however, held a copy of the latest Spamhaus listings. This excerpt shows that all hijacked prefixes were blacklisted on July 28, 2011. According to the Spamhaus Register Of Known Spam Operations (ROKSO) [236], all spam cases were closed between August 24 and September 8, 2011. This indicates that the attacker did not send spam after August 24, which correlates with our control plane results and the victim's efforts to counteract. Up to this point, we have collected a large body of evidence, which confirms the attacker's moves to take over Link Telecom's AS. Table 12.1 summarizes the full course of events divided into specific attack phases.

Data-plane Analysis

In order to better understand the attacker's objectives and to confirm malicious use of the hijacked networks, it is helpful to analyze traffic flows related to the attack. To this end, we leverage archived netflow data from our academic network, the *Münchner Wissenschaftsnetz (MWN)*, as described in Subsection 7.3.2. From April 19 till September 2, 2011, we extracted a total of 603 bidirectional flows related to the LinkTel incident. No further flows were observed for at least one month before and after this time frame, i.e. an observation of random backscatter traffic resulting from spoofed IP addresses is unlikely. In Figure 12.2, we see a correlation between the attack phases identified in Table 12.1 and flows originating from the hijacked prefixes. Note that outgoing traffic that is directed towards unannounced prefixes before the attack (e.g. by end of April) has been analyzed and traced back to research activities on one of MWN's PlanetLab nodes. Not a single flow left the MWN in response to incoming traffic from the unallocated prefix (marked with *(ua)* in Figure 12.2), which indi-

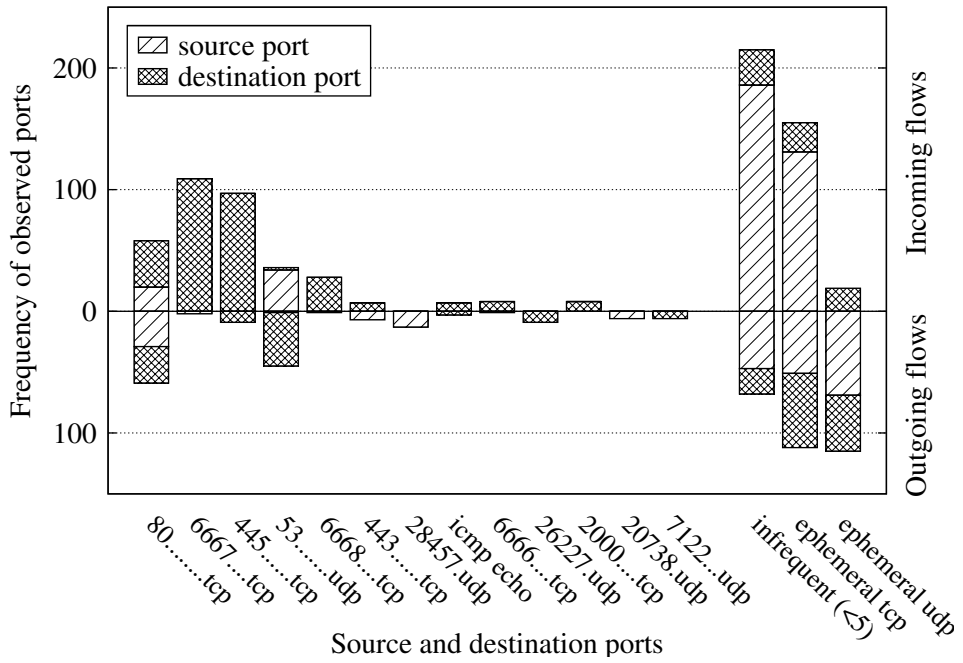


Figure 12.3: Ports involved in traffic flows related to the *LinkTel Incident* (2011).

cates blocking of traffic to bogus destinations at our site. Massive BGP withdraw messages were received for the unallocated space on July 11. On that day, we also observed the last incoming flow from the corresponding prefix.

We already know that the attacker sent spam from the hijacked networks since all prefixes were blacklisted by Spamhaus on July 28. The attacker's first BGP announcement, however, was observed on April 15, i.e. more than 3 months earlier, and we observed the first traffic flow related to this announcement on April 18. It might be the case that Spamhaus' blacklisting techniques take a significant amount of time to detect spam operations, but the attacker could also have carried out other activities without attracting attention for the time between the initial hijacking and the blacklisting. To reconstruct such actions, we take a closer look on the ports involved in the attacker's traffic flows.

Figure 12.3 shows the frequency of ports observed in all extracted flows. These ports were equally observed for all hijacked prefixes. A significant fraction of flows represents bidirectional traffic to port 80 (HTTP). Note that incoming flows to ports 6667 (IRC) and 445 (NetBIOS) were hardly answered. IRC traffic might indicate connections to chat channels or even activities related to botnet command and control. Traffic to port 445 is often used to exploit vulnerabilities of Windows for remote code excecution. Traffic with source port

53 (DNS) implies services hosted in the hijacked prefixes, which is backed up by outgoing HTTP traffic. We further observed incoming flows to port 443 (HTTPS). One of these HTTPS connections was established to a webserver under our control. By analyzing its log files, we were able to extract the content of the connection. We learned that the attacker had created a ticket in our project management system with the following message:

> **currency trading**
> *http://theforexsoftwaretrader.com currency trading*

We queried the Internet Archive [237] for that web site and found a corresponding snapshot from July 23, 2011. Back then, the web site advertised software and tutorials for trading beginners in the currency market. Among others, this included tips on using *"auto pilot software"* to automatically make funds and stock trading strategies based on the utilization of *"Fibonacci retracement" algorithms.*

Summary

The analysis presented above revealed a great deal about the attacker's proceeding. In particular, we observed a hidden takeover attack that was enabled by several preconditions. First, the victim's DNS domain was going to expire, and it was referenced by resource objects in the RIPE database. Second, sending spam from the victim's networks was possible due to suitable reverse DNS delegations. And lastly, most of the victim's prefixes were unannounced and did not show recent activity in BGP. An ideal AS for long-term abuse, we conclude that the Link Telecom has been carefully selected. This was unlikely a manual operation: Various data sources had to be combined to assess the victim's eligibility, which suggests that the attacker had access to automated tools for spotting vulnerable targets.

12.3 Lessons Learned

With PHEW, we introduced a new detection framework to assess the vulnerability of abandoned Internet resources. By studying orthogonal data sources over a period of more than 30 months, we could give evidence of a high risk potential of corresponding attacks. We learned that a significant number of Internet resources is abandoned at the very moment and thus vulnerable to be attacked. Furthermore, we learned that there is an imminent threat by studying the *LinkTel Incident*, a real-world hijacking event. We showed that hidden takeover attacks are feasible with little effort and effectively hide the attacker's iden-

tity. With the evaluation of BGP control plane data and additional meta data, we were able to disclose the attacker's activities and to reconstruct the full sequence of events during the attack. We saw that the attack was carried out in a professional manner in order to send spam from the hijacked networks. Detailed studies of data plane information revealed further objectives. The attacker hosted services in the hijacked prefixes, scanned for client vulnerabilities, and placed adverts for questionable products on web sites and possibly in chat rooms. We assume that the attacker's ability to pass FCrDNS checks supported the abusive use of the hijacked networks.

During our analysis, we repeatedly faced the fact that archives for certain historical data were not available. For forensic analyses on attacks disclosed in the future, access to IRR databases, spam blacklists, and DNS registration data stemming from the time of an attack is indispensable. We suggest to periodically monitor and archive these data sources within the PHEW framework and to add further supplemental information, e.g. by analyzing spam campaigns. In addition, we encourage RIRs to provide unanonymized database snapshots including full details about resource holders. By reflecting on the technical insights we gained from the *LinkTel Incident*, we further profiled the attacker and understood that he must have had access to automated tools in order to detect potential victims. This fact confirms our approach to continuously operate the PHEW framework in order to pre-empt future attackers and to protect Internet resources by informing vulnerable ISPs to deploy countermeasures in time.

13 CHAPTER THIRTEEN
A Real-Time Monitoring System

NOTE *This chapter does not contain prior publication.*

SUMMARY In this chapter, we outline the design of a flexible monitoring framework that combines the author's techniques to assess different types of hijacking attacks. It is based on lessons learned from operational practice and aimed at a continuous monitoring of the global routing system. The framework further supports close monitoring of individual networks and forensic analyses of incidents from the past. Its design is highly extensible: Due to a modular principle, the framework can be enriched by additional functionality such as the analysis of new data sets or new types of attacks that may emerge in the future.

In Section 13.1, we outline functional requirements and specific use cases pertaining to the operation of our combined monitoring framework. Further details on the architectural design are presented in Section 13.2 together with a specification of individual framework modules and their interoperability. In Section 13.3, we discuss implementational aspects for a future deployment of the framework in productive environments.

13.1 Requirements and Objectives

So far, we developed three independent analysis frameworks that address different types of routing attacks. With HEAP (Chapter 7), we can reliably identify false positive subprefix hijacking alarms. CAIR (Chapter 8) provides the necessary tools to detect BGP-based man-in-the-middle attacks. PHEW (Chapter 9) enables us to anticipate hidden takeover attacks by identifying vulnerable Internet resources. We thoroughly evaluated the performance of these systems and showed their applicability in practice along a variety of case studies (Chapters 10 to 12). In the following, we outline the design of a comprehensive monitoring framework that combines the strengths of all three systems. Its main purpose is to continuously monitor the global routing system for imminent or ongoing hijacking attacks. At the same time, operators should be able to register their networks for close monitoring to raise alarms in case of unexpected routing changes.

With our framework, we intend to provide a versatile tool set that allows for an in-depth analysis of past and present routing anomalies. On the one hand, we strive to support quick detection, analysis, and mitigation of routing attacks. On the other hand, insights gained from operational practice and from individual case studies can lead the way towards reliable prevention of hijacking attacks in the future. The potential user's area thus comprises network operators, the scientific community, standardization task forces, and developers of security solutions.

Continuous Operation

All networks connected to the Internet should be continuously monitored for routing attacks. In addition, an interface to receive alarms raised by third-party hijacking detection systems shall be in place. Routing changes in BGP need to be evaluated in real-time in order to identify emerging attacks in a timely manner and to assess the validity of hijacking alarms fed into the framework by external systems. Active measurements as required by individual detection techniques are to be carried out with a tight coupling to the analysis of BGP updates. Confirmed hijacking alarms should be publicly reported, responsibly disclosed, and kept up-to-date on a web-based alarming portal. This portal can further provide detailed statistics on past and present threat levels of hijacking attacks in the Internet. The monitoring system should also be capable to assess the risk exposure of individual networks as well as hijacking trends and developments, which allows to anticipate changes in attack tactics as well as new types of attacks early in time.

Network Monitoring

Potential users of the framework should be able to register their network for close monitoring. An alarming system shall be in place to notify operators about imminent threats and ongoing attacks. Depending on the particular type of attack, recommendations about suitable mitigation strategies shall be provided to the users. To improve the overall detection results of our framework, network providers should be encouraged to provide additional data such as BGP feeds from their own networks and to opt-in for SSL/TLS monitoring (see Section 10.5) by setting up an SSL/TLS-enabled host. A security mechanism shall be in place to prevent attackers from monitoring a potential victim's networks.

Forensic Analysis

Another important use case of our combined monitoring framework is to study routing incidents from the past that become known at a later time. It shall be possible to assess the preconditions of an attack as well as the course of actions in order to anticipate the attacker's goals. To this end, the framework needs to maintain a comprehensive data archive that allows for a root cause analysis of past (and present) attacks. The corresponding query interface should also allow registered users to request live, event-specific data in order to manually assess ongoing network problems and routing anomalies during daily operations.

13.2 Framework Design

In the following, we outline the design of a flexible monitoring framework that meets the functional requirements as outlined above. Our framework utilizes separate modules for input/output, data archiving, and analysis, which can be easily extended by additional submodules as the need arises. We present an initial set of modules that is sufficient to start operating the framework, which fully provides the functionality of the hijacking detection schemes developed by the author.

13.2.1 System Modules

The design of our monitoring framework is built upon five module types that each serve a specific purpose. *Input modules* receive live data feeds, like BGP updates, and conjectural alarms of external detection systems. Similar modules support the input of specific routing

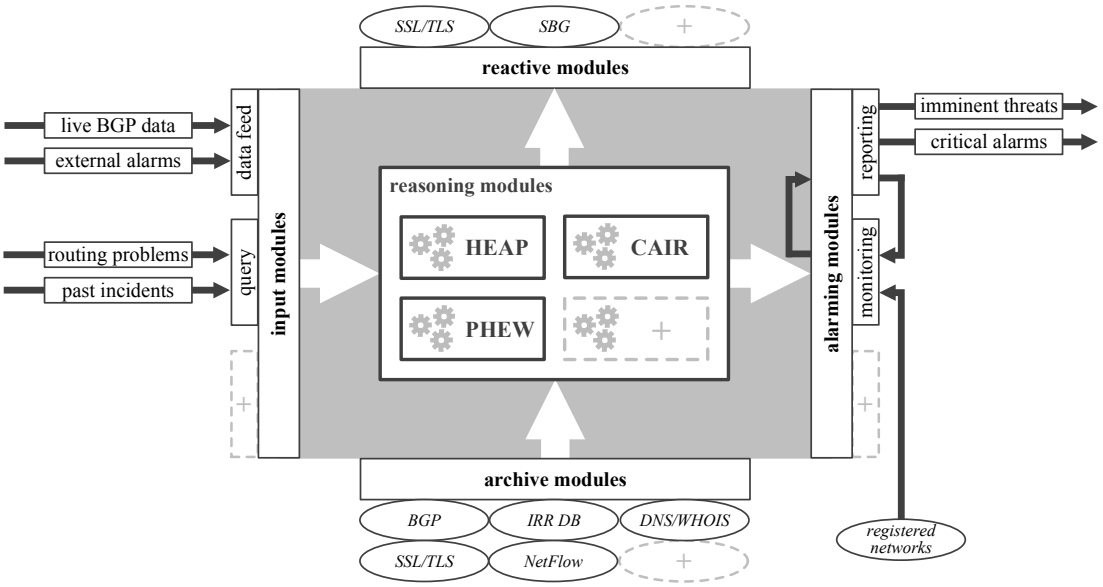


Figure 13.1: A combined real-time monitoring framework.

incidents to selectivly query for data sets on past and present events. The *archive modules* integrate comprehensive data sets into the framework, including historical data as required by individual detection techniques. These modules are asynchronously updated on an as-needed basis. With our *reactive modules*, we support active measurements to collect additional information for the analysis of particular events. *Alarming modules* provide means to report imminent threats and critical alarms to the public or to individual users. Note that specific networks can be (re)inserted into the system for close monitoring. In this respect, a feedback loop allows to continuously monitor suspicious events and to escalate threat levels if necessary. The *reasoning modules* finally provide the actual detection techniques developed in this thesis, thereby connecting all other modules together. Figure 13.1 presents the resulting architectural diagram.

Input Modules

Our monitoring framework is designed to processes data feeds in real-time. The corresponding module expects a live stream of BGP updates, which can be obtained from BGP routers or from projects like BGPmon [223], for instance. Such a data feed consists of AS paths towards IP prefixes including a timestamp of the observation and a flag indicating if the reported routes are newly advertised or withdrawn. In addition, the data feed module accepts external hijacking alarms that concern a hijacked (sub)prefix and its origin ASes.

With a separate query module, we provide access to our rich data sets to study day-to-day routing problems as well as newly discovered incidents from the past. To this end, a user can feed past or present events defined by arbitrary AS numbers or prefixes into the system. The result is a full report of all activity related to the Internet resources. Corresponding (forensic) analyses enable a root cause analysis of unspecific routing problems and an assessment of vulnerabilities in particular networks. As a consequence, this interface needs to be protected against abuse by restricting access to authorized users only.

Archive Modules

With the help of various archive modules, we provide access to all data sources that were utilized throughout this thesis. First, we archive BGP data that is continuously fed into the monitoring framework. Second, we maintain archives of IRR database snapshots (Section 3.3.1), which proved to be a valuable source of administrative information about Internet resources. We also issue corresponding WHOIS queries for additional DNS information (Subsection 3.3.1) to supplement this data set. Furthermore, we collect a comprehensive set of SSL/TLS fingerprints (Subsection 3.3.3) that serves as a ground truth for the validation of hijacking alarms. Last, but not least, we incorporate archived netflow data (Subsection 7.3.2) into our system that can provide further insights into an attacker's motivation.

The data sets presented above are kept up-to-date in an asynchronous way. Each data source requires specific actions to obtain new data, which need to be carried out in different intervals, like monthly scans to update the SSL/TLS ground truth or daily FTP downloads to obtain newly archived IRR database snapshots, for instance. These modules thus need to take care of updating transparently by themselves and are not directly managed by our monitoring framework. In this respect, we assume that the data provided by individual archive modules is always up-to-date. Note that this approach fosters the integration of additional data sources as provided by framework users or fellow researchers.

Reactive Modules

During the process of evaluating routing anomalies and attacks, the need for targeted measurements may arise. To assess certain types of hijacking attacks, we thus provide a reactive component that allows us to carry out active measurements tightly coupled to the analysis process. With HEAP, for instance, we need to scan hosts for SSL/TLS certificates in a timely manner and compare them to an initial ground truth data set. Furthermore, our

IP geolocation system (Section 7.3) can provide additional details on suspicious networks upon request. Such supplemental measurements can be easily added to and controlled by our framework. Note that stand-alone measurement operations that target a large set of networks without initial suspicion are beyond the scope of a reactive module. If needed, such measurement campaigns can be carried out independently and integrated into our framework via appropriate archive modules, as is the case with our SSL/TLS ground truth measurements, for instance.

Alarming Modules

With our alarming modules, we provide a user-facing component that serves as a publicly accessible hijacking archive. Note that we practice responsible disclosure such that no harm is caused to the affected parties by publishing corresponding alarms. We also offer the possibility to register as a user. Interested network operators may create an account in our framework in order to receive tailored reports and real-time notifications about imminent threats to their networks. Corresponding assessment reports and early warnings may come along with recommendations on how to solve network problems or mitigate ongoing attacks. For registered users, we also provide access to our rich data sets to allow for a manual analysis of endangered or attacked networks. Our framework further supports post-mortem analyses of incidents lying in the past. Beyond that, suspicious events as identified by our detection schemes can be reinserted into the framework in order to be continuously monitored for peculiar changes. If such a change occurs, the threat level of a suspicious event can be raised. By triggering a corresponding alarm, involved parties, like the victim and the attacker's upstream ISPs, for instance, can be immediately notified about the emerging attack and advised to deploy countermeasures in time.

Reasoning Modules

The central part of our framework is given by three independent reasoning modules that provide an implementation of the author's detection systems HEAP, CAIR, and PHEW. With these techniques, we can assess the legitimacy of subprefix and AS hijacking alarms, detect ongoing interception attacks, and identify abandoned Internet resources that are prone to be taken over. Our system further addresses operational anomalies such as global route leaks that affect the Internet as a whole. If needed, targeted measurements can be initiated to obtain further data on particular events. More details about the individual analysis steps carried out with these modules can be found in Chapters 7 to 9.

13.2.2 Extensibility

As outlined above, a key design goal of our monitoring framework is its extensibility at all levels. In Figure 13.1, we accordingly illustrate numerous possibilities for future extension. Our framework provides built-in support for additional sources of input data, as is the case with further data archives that can be easily incorporated to the system. Newly emerging detection techniques can be encapsulated and directly integrated as such into the framework, even different kinds of active measurements that are controlled by individual reasoning modules can be added. Furthermore, the presentation of analysis results can be extended as required by future users, too. Improved alarming modules could implement emergency notification via email and SMS, for instance, or might provide additional tools for visualization like those described in Section 7.3.

13.3 Implementation Roadmap

So far, our techniques to assess hijacking attacks, and corresponding reasoning modules respectively, have been thoroughly implemented, tested, and deployed in practice. These modules process BGP data feeds as well as external alarms and allow for individual case studies by querying for event-specific data. All archive modules as described above are readily available within a prototype implementation. We also developed a reactive module to carry out SSL/TLS measurements and operated it over the course of several months. For our evaluation, we further implemented a wealth of reporting modules that generate comprehensive statistics and diagrams in an automated way. Vulnerable networks and critical hijacking alarms are reported to log files and can be manually analyzed by querying the system for additional data. A fully functional proof-of-concept implementation of our combined framework to be deployed in real-time environments is thus within reach.

The prototypic components of our monitoring framework are implemented in Python. Hence, the most straightforward way to extend the system is by supplying customized modules written in the same programming language, though other types of software components can be integrated with little effort as well. In general, our framework is suitable to be operated on commodity hardware. Specific modules, however, might require a more extensive amount of storage or computing capacity. In this respect, particular modules can be easily distributed onto different servers, since all communication within the framework is already realized with secure SSH connections.

The release of a reference implementation of our monitoring framework is part of the author's recommendation on future work. With respect to a productive deployment, the existing system parts could be strengthened as follows. First, the integration of individual components can be improved in terms of state management and error handling. Second, some computationally intensive parts of the framework should be reimplemented in a programming language that is tied more closely to the underlying hardware, like C, for instance. The author's work on measurement-based IP geolocation could be reactivated and integrated into the framework to provide additional information on suspicious network activities. Finally, the presentation of evaluation results to the public, and to individual users respectively, can still be improved. To this end, the implementation of an interactive web portal that further includes a component for user management is considered reasonable to foster broad acceptance of our monitoring framework in the network community.

PART FOUR

Discussion and Conclusion

14 CHAPTER FOURTEEN
Author's Contributions

This thesis documents the author's commitment to a thorough evaluation of architectural threats to Internet routing. During his research, the author made significant contributions [1, 2, 3, 4, 5, 6, 7, 8, 9] to the scientific community. He developed innovative concepts to study routing attacks and advanced the state-of-the-art in several directions. In this chapter, we summarize his contributions and compare them to related work.

14.1 Achieved Research Objectives

The author's work is structured around three main objectives (see Section 2.1 for details). First, he analyzed security flaws in interdomain routing from a theoretical point of view (**Objective O-1**). Second, he developed innovative techniques to detect and assess sophisticated types of routing attacks (**Objective O-2**). Finally, the author applied his techniques in practice to conduct an empirical threat analysis, leading to the design of a combined real-time monitoring system (**Objective O-3**). In the following, the author's contributions are revisited and compared to his declared objectives.

O–1 Security Flaws in Interdomain Routing

In Chapter 3, the author discussed current procedures for the global management of Internet resources and established a well-founded basis for subsequent analysis *(Objective O-1.1)*. The non-profit organisation IANA is responsible for the distribution of Internet resources and delegates operational responsibility to Regional Internet Registries and Network Information Centers. Trusted Third Parties serve to manage cryptographic certificates that are used to secure communications in the Internet. The author discussed how to obtain corresponding data sets on resource delegation and their utilization in the Internet ecosystem. Such data sets can be leveraged to study routing anomalies in greater detail.

The Border Gateway Protocol, a de-facto monopolistic interdomain routing protocol, plays a most significant role in Internet routing. In Chapter 4, the author unraveled its history of origins, which dates back almost four decades. He assessed shortcomings in its design *(Objective O-1.2)* and showed that the protocol still exhibits severe flaws with respect to security. Even though several extensions were proposed in the 2000s, interdomain routing relies on mutual trust and is thus susceptible to hijacking attacks. A full protection against such attacks within BGPsec is out of reach for many years.

The author further derived a comprehensive attacker model and thoroughly analyzed BGP-based attacks in Chapter 5 *(Objective O-1.3)*. To this end, he developed a novel concept of Finite Route Languages providing a rigorous model to formalize various aspects of Internet routing. Applied to BGP hijacking attacks, the author is first to present a systematic and exhaustive classification including a detailed analysis of their impact. He formalized different types of attacks comprising origin relocation, route diversion, and hidden takeover, and discussed suitable attack tactics as well as the motivation behind such attacks.

In Chapter 6, the author assessed state-of-the-art techniques to detect ongoing hijacking attacks *(Objective O-1.4)* and studied their applicability in practice. Such frameworks utilize data from the control-plane, i.e. information extracted from live routing tables, or data-plane measurements to obtain a forwarding-based view on Internet routing. Several systems combine the strengths of both approaches. It is surprising, though, that corresponding techniques focus on rather basic types of attacks while neglecting more sophisticated variants for the utmost part.

O–2 Detection and Analysis of Routing Attacks

With his own work, the author strived to address challenging types of hijacking attacks. In Chapter 7, he developed HEAP, a system to assess the validity of hijacking alarms in an effort to reduce high rates of false positives raised by current techniques *(Objective O-2.1)*. HEAP combines several unique data sources to identify a legitimate cause behind suspicious routing anomalies. First, IRR databases are searched for legitimizing business relationships between an alleged attacker and his victim. Second, SSL/TLS measurements are carried out to obtain cryptographic assurance about network holdership. And lastly, a heuristic algorithm to assess legitimate topological constellations is added to the system. Taking into account the author's earlier work on traffic flow analysis and IP geolocation, HEAP provides a rich set of data sources to consult for the analysis of hijacking alarms.

In Chapter 8, the author presented a novel technique to study sophisticated path manipulation in BGP *(Objective O-2.2)*. With the CAIR framework, he put his formal routing model to practical use by introducing the concept of route automata. This data structure is a most efficient representation of network paths and significantly outperforms common approaches in terms of expressiveness. CAIR uniquely enables the analysis of routing policies on top of static network topologies and allows to search for policy violations therein. In this respect, the author's formalization of routing attacks can be translated into practical search patterns that are implementable in corresponding route automata. The author consequently derived and implemented a detection scheme for BGP-based interception attacks and route leaks, which were so far impractical to detect. He further showed that the applicability of CAIR reaches well beyond the detection of routing anomalies.

Anecdotal evidence exists that abandoned Internet resources have been hijacked and abused by attackers. In Chapter 9, the author presented the PHEW system to comprehensively assess the actual threat imposed by such resources *(Objective O-2.3)*. PHEW is designed as an early warning system to identify vulnerable networks. By leveraging IRR databases, WHOIS queries, and archived BGP data, several activity measures were derived to assess the utilization of Internet resources. The author showed that idle resources exist and bear a high risk potential for being attacked. With the PHEW framework, vulnerable resource holders can be reliably identified and protected by informing inattentive operators about imminent threats in time.

O–3 Empirical Threat Analyses

The author thoroughly evaluated all detection frameworks presented in this thesis and carried out a variety of forensic case studies to put the results into perspective *(Objective O-3.1)*. In Chapter 10, he analyzed the HEAP framework to assess its capabilities of legitimizing routing anomalies. By studying common day-to-day events, he established an encouraging base line for practical validation of hijacking alarms. In this respect, we learned that HEAP is able to legitimize up to 56.93% of such events. Restricting the input events to more valuable targets for an attack, HEAP yields an even higher rate of 81.15% legitimate events. More importantly, the author evaluated a set of publicly reported hijacking alarms and showed that HEAP can still identify nearly 10% false positives in such a highly focused set of alarms. The author further conducted a case study on a suspected hijacking incident taking place in Bulgaria, which involved abusive use of corresponding networks to send large amounts of spam. By taking into account all data sources provided by HEAP,

we discovered substantial evidence against a real hijacking attack. HEAP is thus suitable to address hijacking alarms both in terms of automated reasoning and manual investigation.

In Chapter 11, the author evaluated the CAIR framework and applied it to detect ongoing path manipulations in BGP. By studying performance properties of CAIR, we learned that the approach is highly effective and well-suited to monitor routing in real-time. With an evaluation of BGP routing tables from more than seven years, the author demonstrated great potential in using CAIR for comprehensive routing analyses. The evaluation revealed 22 new cases of interception attacks, of which two cases were studied in close detail. Corresponding results for the well-known DEFCON incident confirmed that CAIR can correctly identify man-in-the-middle attacks while providing useful background on such incidents at the same time. In addition, a case study on an interception incident taking place in Iran showed that such attacks actually occur in practice. We further gained insight into normal and abnormal routing changes and studied a known route leak incident in thorough detail. In this respect, the author derived a novel measure for routing dominance to assess the significance of individual ASes in the global routing system.

The PHEW system was evaluated and applied in practice in Chapter 12. PHEW is designed to detect inactive resource holders at large with a measure to rate their utilization. By studying orthogonal data sources over a period of more than 30 months, the author could give evidence that abandoned Internet resources are a real phenomenon. In this respect, we learned that a total of 73 /24 networks and 7 ASes were at risk at the time of writing. A thorough case study on a real incident was carried out to illustrate the actual threat. The author studied an attacker's proceeding to takeover a victim's Internet resources in order to send spam from the corresponding networks. We learned how expired DNS names enabled the attack and that countermeasures are hard to implement for ongoing attacks. The findings of this forensic analysis emphasize the need for an early warning system.

Based on lessons learned from the evaluation, the author developed the design of a comprehensive and flexible monitoring framework *(Objective O-3.2)*. This framework combines all detection techniques presented in this thesis in an effort to increase the overall security of interdomain routing. To this end, requirements and use cases for such a system were discussed in detail. The architectural design is highly modular and allows for an easy integration of additional data sources as well as future techniques to detect and analyze hijacking incidents. The author further specified an initial set of input/output and analysis modules and their interaction within the framework. He also provided an implementation roadmap to put the system into operation within a short period of time.

	Origin Relocation		**Route Diversion**		**Hidden Takeover**	**Applicability**
	Prefix Hijacking	*Subprefix Hijacking*	*AS Hijacking*	*Man-in-the-Middle*	*Abandoned Resources*	
control-plane detection						
PHAS [176]	+	o	–	–	+	o *(operator-level)*
Bogus Routes [177]	+	o	o	o	o	+ *(global scale)*
data-plane detection						
LWDS [181]	+	+	o	o	–	– *(incident-level)*
iSpy [184]	+	–	–	–	–	o *(operator-level)*
hybrid techniques						
Fingerprints [185]	+	o	o	–	–	+ *(global scale)*
Argus [188]	+	o	o	–	–	+ *(global scale)*
author's contributions						
HEAP [5, 7, 8]	**o**	**++**	**++**	**–**	**–**	**+** *(global scale)*
CAIR [9]	**o**	**o**	**o**	**++**	**–**	**+** *(global scale)*
PHEW [3, 4, 6]	**–**	**–**	**–**	**–**	**++**	**+** *(global scale)*

++/+ practical usefulness o theoretic applicability – not supported

Table 14.1: Comparison of the author's contributions to state-of-the-art.

14.2 Comparison to State-of-the-Art

The author strived to fill a gap in current network research that mostly neglects the detection and analysis of more challenging routing attacks. To this end, we studied the applicability of related work for the detection of different types of hijacking attacks (see Chapter 6). We assessed prominent representatives of control-plane [176, 177] and data-plane [181, 184] approaches as well as hybrid techniques [185, 188] and learned that, on the one hand, all of these techniques can reliably detect ordinary prefix hijacking attacks. On the other hand, only one technique [181] is capable to detect attacks on subprefixes in practice. It is, however, ill-suited to be deployed on a global scale. More importantly, neither AS hijacking nor man-in-the-middle attacks are addressed by previous work in a satisfying manner due to infeasible requirements or limiting assumptions of the corresponding techniques. Finally, one single technique [176] is suitable to detect attacks on unannounced address spaces, but it is limited to a deployment by individual network operators.

Table 14.1 summarizes the strenghts and limitations of these detection techniques (as presented with Table 6.1) and adds a comparison to the author's contributions. HEAP is particularly well-suited to assess subprefix hijacking attacks possibly combined with AS hijacking. The framework can be deployed on a global scale to monitor routing anomalies in

BGP. More importantly, it supports the assessment of hijacking alarms as raised by external systems [176, 177, 181, 185, 188]. CAIR is a generic framework to study routing characteristics. The author demonstrated its capabilities to analyze routing attacks by implementing a search pattern for man-in-the-middle attacks. He is first to solve this challenge in a fundamental way. PHEW is a system to detect vulnerable resources prior to an attack in order to subsequently monitor these resources for suspicious routing changes. It is capable to address hidden takeover attacks on a global scale.

15 CHAPTER FIFTEEN
Conclusion and Perspectives

During his research, the author studied Internet operations and discussed shortcomings in interdomain routing with respect to security. He derived a formal routing model and applied it to describe corresponding attacks, developed innovative detection frameworks, and thoroughly evaluated their practical usefulness. In addition, he conducted a variety of case studies and demonstrated that his work addresses a real threat. All of the presented techniques are self-contained, fully functional, and ready to be deployed in practice. Moreover, these approaches can be combined into a comprehensive real-time framework to continuously monitor the global routing system.

In summary, the author fully succeeded in implementing his research agenda. At the same time, his work offers great potential for future research. On the one hand, his systems can be technically improved with respect to methodology and input data. In this respect, the author already identified several open tasks to further increase the effectiveness of individual detection frameworks. On the other hand, his work leads the way to future research on routing attacks and, more importantly, opens up new scientific opportunities to study Internet routing in general.

15.1 Further Development

The author suggests to further develop his detection systems in future work. To this end, the existing prototypes should be enhanced with additional functionality as follows.

Hijacking Event Analysis Program (HEAP) New data sources can be easily added to the framework to support the assessment of hijacking alarms in greater detail. The author's work on IP geolocation [1] and traffic flow analysis [2] is a natural choice for the incorporation of orthogonal data sets. Commercial IRR databases might provide a complementary view on business relationships between Internet participants. In this respect, a cross-validation of IRR databases should be carried out to improve overall data quality. In addition

to SSL/TLS measurements, other network protocols, like SSH [71] and DNSSEC [42], can be leveraged to obtain further cryptographic assurance on legitimate routing events. Finally, HEAP should be interfaced with state-of-the-art and future detection systems of fellow researchers to evaluate their conjectural hijacking alarms.

Constructible Automata for Internet Routes (CAIR) The detection of interception attacks, and hijacking attacks in general, can still be improved. First, the presented search pattern for man-in-the-middle attacks may yield false positive alarms for sibling ASes. While manually assessed during the evaluation, such constellations can also be identified in an automated way based on BGP data or IRR databases. Second, more data sets should be analyzed and further incidents ought to be studied in detail. Finally, new search patterns for (sub)prefix hijacking and AS hijacking attacks as well as for other types of routing anomalies can be derived. Beyond that, structural properties of route automata should be analyzed in more detail with respect to performance and expressiveness. More sophisticated concepts to represent formal languages, like so-called *deterministic finite-state cover automata* [238, 239], may offer further potential to increase effectiveness.

Prefix Hijacking Early Warning (PHEW) At the moment, PHEW is solely based on data sets that are published by RIPE. It is a straightforward task, however, to incorporate the databases of all RIRs as well as those of commercial IRRs. PHEW is currently limited to monitor Internet resources that are bound to a single domain name in order to prevent false positive alarms. This limitation can be addressed by disregarding well-known domain names that are unlikely to expire. Additional data sets can be leveraged to improve the activity metric for Internet resources. Since PHEW is solely based on the evaluation of historic resource utilization, corresponding data archives should be set up and maintained as soon as possible. Individual resources that are at risk need to be protected quickly, either by contacting their holders or by taking over such resources on purpose to pre-empt an attack.

Real-time Monitoring Framework The architectural design of a comprehensive monitoring system directly serves as a blueprint for its future implementation. While its vital parts, i.e. the reasoning and data archiving modules, are well advanced and ready for integration, supplementary modules still need to be developed. To this end, a controller component is required to orchestrate the existing modules. Input interfaces are to be created for receiving external alarms and to allow for selective data queries to conduct manual analyses. It is imperative to connect the system to a live stream of BGP data for continuous operation. The reactive component to collect additional measurement data on an as-needed basis can be extended to deploy counter measures in an automated way. Moreover, user-facing

alarming modules need to be developed to report imminent threats and ongoing attacks to registered subscribers. A feedback loop should be installed to closely monitor suspicious events over time. After the implementation is complete, the system should be continuously deployed and announced to network operators and researchers.

15.2 Future Research Directions

A broad area of research can benefit from the author's work on routing threats. With practical insights gained on attacks, his evaluation lays the ground for reliable mitigation strategies. More importantly, his detection schemes can help to develop and sustain a secure routing architecture in the future. Last, but not least, the author's novel routing model may lead the way to a new understanding of policy-aware routing analysis.

15.2.1 Mitigating the Threat of Routing Attacks

In this thesis, the author showed that the Internet is vulnerable to routing attacks and that such attacks occur in practice. It is thus an important goal to develop effective mitigation strategies in the future. The author's work can support this development in several ways. First, his analysis of different attack vectors in BGP serves to establish a solid understanding of the potential attack surface. In this respect, an attacker's goals can be anticipated by taking into account the author's assessment of suitable attack tactics. Second, the detection frameworks presented in this thesis can provide the necessary background to select appropriate countermeasures during an attack. Reliable information on preconditions of an attack and an attacker's course of actions is indispensable for effective mitigation. It is worth mentioning that the automated deployment of reactive countermeasures is still an open research question. Finally, post-mortem analyses can be conducted using the author's frameworks to timely adapt mitigation concepts to evolving attack scenarios.

Having suitable countermeasures in place can significantly increase deterrence against attacks. At the same time, future research should be directed to develop preventive measures to further reduce the risk of attacks. The author's work on early warning systems can be continued and extended to inform network operators about potential risks in good time. It is an open issue to identify such vulnerable networks. Proactive measures to pre-empt attacks should be considered in case of imminent threats. Future work should also provide recommendations on best practice network operations to hinder attacks.

15.2.2 Towards a Secure Routing Architecture

The development of mitigation strategies is arguably an intermediate solution only. A more sustainable goal is to secure the global routing system in order to fully prevent attacks in the future. Much effort has already been put into the improvement of security in interdomain routing. BGPsec [115], the most promising approach, still lacks a complete specification and the support by router vendors though. By documenting a real threat, the author's analysis of routing attacks may help to push the development of such solutions. Corresponding case studies on real attacks can further help to engineer their security requirements and eventually foster their deployment by providing a strong incentive. Until then, the author's concept of route automata could be applied to monitor the global routing table for compliance with routing policies as intended by individual providers. In case of violations, network operators could then refuse to propagate illicit route announcements.

While BGPsec is designed to ensure the technical validity of routing changes, other threats exist on an administrative level. Routing attacks can be enabled by social engineering as network operators generally arrange peering relations based on a weak authentication scheme. Restoring trust in such provider interrelations is an important task for future research. To this end, we need concepts, tools, and procedures for resource ownership validation. The author's work on SSL/TLS measurements to cryptographically assure network authenticity can be a first step towards such a solution. A possible outcome of future work might be a notary system that transparently attests network ownership. Such a system would enable Internet service providers to authenticate their customers, thereby effectively preventing attacks. Moreover, it could strengthen today's Internet ecosystem by providing the means to unambiguously identify resource holders with respect to accounting and law enforcement.

15.2.3 Advancing the Field of Routing Analysis

With this thesis, the author presented a novel model for Internet routing based on concepts of formal languages. While he applied this model exclusively to study anomalies, it can be naturally adopted for arbitrary routing scenarios. Future work can be conducted on numerous fields of application, including but not limited to the following areas of research. First, structural properties of route automata should be analyzed in more detail. The adaption of well-known network graph algorithms to operate on route automata is particularly promising. In this context, further insights into the nature of Internet routing might

be gained with respect to convergence, resilience, and policy adherence. Second, common techniques to assess AS characteristics and the type of relationship between ASes could greatly benefit from an analysis with route automata. In this respect, the author's metric on routing dominance should be further investigated. Third, route automata can also be applied to the field of IP-level routing analysis. The identification of Internet exchange points and points of presence might profit from the natural representation of equivalent routing paths in the automata. Techniques to map individual IP addresses to routers lend themselves for further analysis in a similar way. Finally, the author's route model could be applied to enhance control-plane operations. To this end, software routers as wells as controller elements in software-defined networking might benefit from a rigorous representation of policy-induced network paths. Beyond that, cross-sectional topics, like network diagnosis, offer further potential for future work. The author invites fellow researchers to put his model to the test and apply it to research questions that go beyond his own.

It is worth noting that the definition of finite route languages is consistent in itself and solely based on formal languages. The author assumes, though, that a partial algebra [240] defined over such a language could be embedded into Sobrinho's routing algebra [241, 242, 243]. Given the existence of such a link, we would be able to approach the mathematical theory of policy-based routing [244, 245] from an experimental point of view.

15.3 Closing Remarks

The author played a leading role in developing the concepts presented in this thesis to advance the state-of-the-art on Internet routing analysis. Nonetheless, he closely collaborated with other researchers. His work is tied to that of Quentin Jacquemart and Pierre-Antoine Vervier at Institut Eurécom, Sophia Antipolis, France, and Matthias Wählisch at Freie Universität Berlin, Germany. During his research at Technische Universität München, Germany, the author further supervised the work of 49 student researchers. Writing their thesis under the author's guidance, 21 of them received their Bachelor's degree and another 15 students their Master's degree. Eventually, 9 of his former students followed the author's way to academia.

PART FIVE

APPENDIX

APPENDIX

List of Figures

APPENDIX

List of Tables

APPENDIX

List of Prior Publications

[1] S. Günther, **J. Schlamp**, and G. Carle, "Spring-based Geolocation", in *Proceedings of the IEEE/IFIP Symposium on Network Operations and Management*, NOMS '12, April 2012.

[2] L. Braun, M. Volke, **J. Schlamp**, A. von Bodisco, and G. Carle, "Flow-Inspector: A Framework for Visualizing Network Flow Data using Current Web Technologies", in *Proceedings of the First IMC Workshop on Internet Visualization*, WIV '12, November 2012.

[3] **J. Schlamp**, G. Carle, and E. W. Biersack, "How to prevent AS hijacking attacks", in *Proceedings of the ACM CoNEXT Student Workshop*, CoNEXT Student '12, December 2012.

[4] **J. Schlamp**, G. Carle, and E. W. Biersack, "A Forensic Case Study on AS Hijacking: The Attacker's Perspective", *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 43, pp. 5–12, April 2013.

[5] P.-A. Vervier, Q. Jacquemart, **J. Schlamp**, O. Thonnard, G. Carle, G. U. Keller, E. W. Biersack, and M. Dacier, "Malicious BGP Hijacks: Appearances can be deceiving", in *Proceedings of the IEEE ICC Communications and Information Systems Security Symposium*, ICC CISS '14, June 2014.

[6] **J. Schlamp**, J. Gustafsson, M. Wählisch, T. C. Schmidt, and G. Carle, "The Abandoned Side of the Internet: Hijacking Internet Resources When Domain Names Expire", in *Proceedings of the International Workshop on Traffic Monitoring and Analysis*, TMA '15, April 2015.

[7] **J. Schlamp**, R. Holz, O. Gasser, A. Korsten, Q. Jacquemart, G. Carle, and E. W. Biersack, "Investigating the Nature of Routing Anomalies: Closing in on Subprefix Hijacking Attacks", in *Proceedings of the International Workshop on Traffic Monitoring and Analysis*, TMA '15, April 2015. Best Paper Award.

[8] **J. Schlamp**, R. Holz, Q. Jacquemart, G. Carle, and E. W. Biersack, "HEAP: Reliable Assessment of IP Subprefix Hijacking Attacks", *IEEE Journal on Selected Areas in*

Communications–Special Issue on Measuring and Troubleshooting the Internet (JSAC-SI-MT), vol. 33, May 2016.

[9] **J. Schlamp**, M. Wählisch, T. C. Schmidt, G. Carle, and E. W. Biersack, "CAIR: Using Formal Languages to Study Policy Violations in BGP", *IEEE/ACM Transactions on Networking (TON)*, 2016. (submitted in December 2016).

APPENDIX

Bibliography

[10] B. Huffaker, M. Fomenkov, and k. Claffy, "Internet Topology Data Comparison", tech. rep., Cooperative Association for Internet Data Analysis (CAIDA), May 2012.

[11] "Yearbook of Statistics: Chronological Time Series 2005–2014", tech. rep., International Telecommunication Union (ITU), November 2015.

[12] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions", *Future Generation Computer Systems*, vol. 29, pp. 1645–1660, September 2013.

[13] F. A. Azevedo, L. R. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. Ferretti, R. E. Leite, R. Lent, S. Herculano-Houzel, *et al.*, "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain", *Journal of Comparative Neurology*, vol. 513, pp. 532–541, April 2009.

[14] C. Zimmer, "100 trillion connections: New efforts probe and map the brain's detailed architecture", *Scientific American*, vol. 304, pp. 58–63, January 2011.

[15] B. Rezabakhsh, D. Bornemann, U. Hansen, and U. Schrader, "Consumer Power: A Comparison of the Old Economy and the Internet Economy", *Journal of Consumer Policy*, vol. 29, pp. 3–36, March 2006.

[16] P. DiMaggio, E. Hargittai, W. R. Neuman, and J. P. Robinson, "Social Implications of the Internet", *Annual Review of Sociology*, pp. 307–336, August 2001.

[17] H. Farrell, "The Consequences of the Internet for Politics", *Annual Review of Political Science*, vol. 15, pp. 35–52, March 2012.

[18] A. Dainotti, C. Squarcella, E. Aben, K. C. Claffy, M. Chiesa, M. Russo, and A. Pescapé, "Analysis of Country-wide Internet Outages Caused by Censorship", in *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, IMC '11, 2011.

[19] E. Stepanova, "The Role of Information Communication Technologies in the "Arab

Spring"", *Ponars Eurasia*, vol. 15, pp. 1–6, May 2011.

[20] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. D. Johnson, "M2M: From Mobile to Embedded Internet", *IEEE Communications Magazine*, vol. 49, pp. 36–43, April 2011.

[21] J. Van Dijck, *The Culture of Connectivity: A Critical History of Social Media*. Oxford University Press, 2013.

[22] J. S. Turner and D. E. Taylor, "Diversifying the Internet", in *Global Communications Conference*, GLOBECOM '05, 2005.

[23] E. C. Rosen, "Exterior Gateway Protocol", RFC 209, DARPA, October 1982.

[24] J. Li, M. Guidero, Z. Wu, E. Purpus, and T. Ehrenkranz, "Bgp routing dynamics revisited", *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 37, pp. 5–16, March 2007.

[25] G. Huston, "Analyzing the Internet's BGP routing table", *The Internet Protocol Journal*, vol. 4, pp. 2–15, March 2001.

[26] T. Bu, L. Gao, and D. Towsley, "On characterizing BGP routing table growth", *Computer Networks*, vol. 45, pp. 45–54, May 2004.

[27] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1654, IETF, July 1994.

[28] M. Brown, "Pakistan hijacks YouTube". `http://research.dyn.com/2008/02/pakistan-hijacks-youtube/`, February 2008. Dyn Research.

[29] M. S. Siddiqui, D. Montero, M. Yannuzzi, R. Serral-Gracia, and X. Masip-Bruin, "Route Leak Identification: A Step Toward Making Inter-Domain Routing More Reliable", in *International Conference on the Design of Reliable Communication Networks*, DRCN '14, 2014.

[30] B. Al-Musawi, P. Branch, and G. Armitage, "Detecting BGP Instability Using Recurrence Quantification Analysis (RQA)", in *Proceedings of the IEEE Performance Computing and Communications Conference*, IPCCC '15, 2015.

[31] J. Cowie, "China's 18-Minute Mystery". `http://research.dyn.com/2010/11/chinas-18-minute-mystery/`, November 2010. Dyn Research.

[32] A. Ramachandran and N. Feamster, "Understanding the Network-Level Behavior of Spammers", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '06, 2006.

[33] J. Postel, "Proposed Standard Socket Numbers", RFC 349, DARPA, May 1972.

[34] J. Postel, “Assigned Numbers”, RFC 790, DARPA, September 1981.

[35] J. Postel, “Assigned Numbers”, RFC 1060, IETF, March 1990.

[36] B. Carpenter, F. Baker, and M. Roberts, “Memorandum of Understanding Concerning the Technical Work of the Internet Assigned Numbers Authority”, RFC 2860, IETF, June 2000.

[37] J. Reynolds, “Assigned Numbers: RFC 1700 is Replaced by an On-line Database”, RFC 3232, IETF, January 2002.

[38] S. Deering, “Host Extensions for IP Multicasting”, RFC 1112, IETF, August 1989.

[39] Y. Rekhter, R. Moskowitz, D. Karrenberg, G. de Groot, and E. Lear, “Address Allocation for Private Internets”, RFC 1918, IETF, February 1996.

[40] P. Mockapetris, “Domain Names - Concepts and Facilities”, RFC 1034, IETF, November 1987.

[41] J. Postel, “Domain Name System Structure and Delegation”, RFC 1591, IETF, March 1994.

[42] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, “DNS Security Introduction and Requirements”, RFC 4033, IETF, March 2005.

[43] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer, “OpenPGP Message Format”, RFC 4880, IETF, November 2007.

[44] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and T. Polk, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”, RFC 5280, IETF, May 2008.

[45] E. Rescorla, “HTTP Over TLS”, RFC 2818, IETF, May 2000.

[46] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP”, RFC 8960, IETF, June 2013.

[47] D. Eastlake, “Transport Layer Security (TLS) Extensions: Extension Definitions”, RFC 6066, IETF, January 2011.

[48] A. Freier, P. Karlton, and P. Kocher, “The Secure Sockets Layer (SSL) Protocol Version 3.0”, RFC 6101, IETF, August 2011.

[49] T. Dierks and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2”, RFC 5246, IETF, August 2008.

[50] I. USC, “Transmission Control Protocol”, RFC 793, DARPA, September 1981.

[51] J. Postel and J. Reynolds, “File Transfer Protocol (FTP)”, RFC 959, DARPA, October 1985.

[52] P. Ford-Hutchinson, “Securing FTP with TLS”, RFC 4217, IETF, October 2005.

[53] R. Fielding, J. Gettys, J. Mogul, H. Nielsen, L. Masinter, P. Leach, and T. Berners-Lee, “Hypertext Transfer Protocol – HTTP/1.1”, RFC 2616, IETF, June 1999.

[54] R. Khare and S. Lawrence, “Upgrading to TLS Within HTTP/1.1”, RFC 2817, IETF, May 2000.

[55] M. Crispin, “Internet Message Access Protocol - Version 4rev1”, RFC 2060, IETF, December 1996.

[56] C. Newman, “Using TLS with IMAP, POP3 and ACAP”, RFC 2595, IETF, June 1999.

[57] J. Oikarinen, “Internet Relay Chat Protocol”, RFC 1459, IETF, May 1993.

[58] “IRC STARTTLS (Open Standard)”. `https://wiki.inspircd.org/STARTTLS_Documentation`, 2008. InspIRCd.

[59] M. Wahl, T. Howes, and S. Kille, “Lightweight Directory Access Protocol (v3)”, RFC 2251, IETF, December 1997.

[60] J. Hodges, R. Morgan, and M. Wahl, “Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security”, RFC 2830, IETF, May 2000.

[61] B. Kantor and P. Lapsley, “Network News Transfer Protocol”, RFC 977, DARPA, February 1986.

[62] K. Murchison, J. Vinocur, and C. Newman, “Using Transport Layer Security (TLS) with Network News Transfer Protocol (NNTP)”, RFC 4642, IETF, October 2006.

[63] M. Rose, “Post Office Protocol - Version 3”, RFC 1081, IETF, November 1988.

[64] J. Postel, “Simple Mail Transfer Protocol”, RFC 821, DARPA, August 1982.

[65] P. Hoffman, “SMTP Service Extension for Secure SMTP over Transport Layer Security”, RFC 3207, IETF, February 2002.

[66] R. Gellens and J. Klensin, “Message Submission”, RFC 2476, IETF, December 1998.

[67] P. Saint-Andre, “Extensible Messaging and Presence Protocol (XMPP): Core”, RFC 3920, IETF, October 2004.

[68] E. Rescorla and N. Modadugu, “Datagram Transport Layer Security Version 1.2”, RFC 6347, IETF, January 2012.

[69] S. Kent and K. Seo, “Security Architecture for the Internet Protocol”, RFC 4301, IETF,

December 2005.

[70] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen, “Internet Key Exchange Protocol Version 2 (IKEv2)”, RFC 7296, IETF, October 2014.

[71] T. Ylonen and C. Lonvick, “The Secure Shell (SSH) Transport Layer Protocol”, RFC 4253, IETF, January 2006.

[72] B. Ramsdell and S. Turner, “Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification”, RFC 5751, IETF, January 2010.

[73] L. Daigle, “WHOIS Protocol Specification”, RFC 3912, IETF, September 2004.

[74] K. Harrenstein and V. White, “NICNAME/WHOIS”, RFC 812, DARPA, March 1982.

[75] “The Internet Routing Registry”. `http://www.irr.net/`, 2010. Merit Network Inc.

[76] D. Barr, “Common DNS Operational and Configuration Errors”, RFC 1912, IETF, February 1996.

[77] M. Kucherawy, “Message Header Field for Indicating Message Authentication Status”, RFC 5451, IETF, April 2009.

[78] D. Senie, A. Sullivan, and W. Salmon, “Considerations for the use of DNS Reverse Mapping”, Internet-Draft 06, IETF, March 2008.

[79] V. Strazisar, “Gateway Routing, An Implementation Specification”, IEN 30, Internet Project, April 1978.

[80] V. Strazisar, “How To Build A Gateway”, IEN 109, Internet Project, August 1979.

[81] R. Hinden and A. Sheltzer, “DARPA Internet gateway”, RFC 823, DARPA, September 1982.

[82] D. L. Mills, “Exterior Gateway Protocol formal specification”, RFC 904, DARPA, April 1984.

[83] J. Rekhter, “EGP and policy based routing in the new NSFNET backbone”, RFC 1092, IETF, February 1989.

[84] H.-W. Braun, “NSFNET routing architecture”, RFC 1093, IETF, February 1989.

[85] K. Lougheed and J. Rekhter, “Border Gateway Protocol (BGP)”, RFC 1105, IETF, June 1989.

[86] K. Lougheed and Y. Rekhter, “Border Gateway Protocol (BGP)”, RFC 1163, IETF, June 1990.

[87] K. Lougheed and Y. Rekhter, “Border Gateway Protocol 3 (BGP-3)”, RFC 1267, IETF,

October 1991.

[88] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, IETF, March 1995.

[89] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, IETF, January 2006.

[90] Q. Vohra and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", RFC 6793, IETF, December 2012.

[91] O. Nordström and C. Dovrolis, "Beware of BGP attacks", *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 34, pp. 1–8, April 2004.

[92] N. Feamster, H. Balakrishnan, and J. Rexford, "Some foundational problems in interdomain routing", in *Proceedings of the ACM Workshop on Hot Topics in Networks*, HotNets '04, 2004.

[93] S. Murphy, "BGP Security Vulnerabilities Analysis", RFC 4272, IETF, January 2006.

[94] A. Barbir, S. Murphy, and Y. Yang, "Generic Threats to Routing Protocols", RFC 4593, IETF, October 2006.

[95] G. Huston, M. Rossi, and G. Armitage, "Securing bgp – a literature survey", 2010.

[96] D. R. Kuhn, K. Sriram, and D. C. Montgomery, "Sp 800-54. border gateway protocol security", tech. rep., National Institute of Standards & Technology, 2007.

[97] K. Butler, T. Farley, P. Mcdaniel, and J. Rexford, "A survey of bgp security issues and solutions", *Proceedings of the IEEE*, vol. 98, pp. 100–122, January 2010.

[98] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra, "Routing Policy Specification Language (RPSL)", RFC 2622, IETF, June 1999.

[99] L. Blunk, J. Damas, F. Parent, and A. Robachevsky, "Routing Policy Specification Language next generation (RPSLng)", RFC 4012, IETF, March 2005.

[100] S. Kent, C. Lynn, J. Mikkelson, and K. Seo, "Secure Border Gateway Protocol (S-BGP)", *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 18, pp. 582–592, April 2000.

[101] Y.-C. Hu, A. Perrig, and M. Sirbu, "SPV: Secure Path Vector Routing for Securing BGP", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, 2004.

[102] R. White, "Securing BGP Through Secure Origin BGP", *The Internet Protocol Journal*, vol. 6, September 2003.

[103] P. v. Oorschot, T. Wan, and E. Kranakis, "On Interdomain Routing Security and Pretty Secure BGP (psBGP)", *ACM Transactions on Information and System Security*, vol. 10, p. 11, July 2007.

[104] Q. Li, M. Xu, J. Wu, X. Zhang, P. P. C. Lee, and K. Xu, "Enhancing the Trust of Internet Routing With Lightweight Route Attestation", vol. 7, pp. 691–703, February 2012.

[105] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin, "Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing", in *Proceedings of the ISOC Symposium on Network and Distributed Systems Security*, NDSS '03, 2003.

[106] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "Detection of invalid routing announcement in the Internet", in *Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks*, DSN '02, 2002.

[107] E. Wong and V. Shmatikov, "Get Off My Prefix! The Need for Dynamic, Gerontocratic Policies in Inter-domain Routing", in *Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks*, DSN '11, 2011.

[108] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, "Topology-based detection of anomalous BGP messages", in *Proceedings of the International Conference on Recent Advances in Intrusion Detection*, RAID '03, 2003.

[109] S. Y. Qiu, F. Monrose, A. Terzis, and P. D. McDaniel, "Efficient Techniques for Detecting False Origin Advertisements in Inter-domain Routing", in *Proceedings of the IEEE Workshop on Secure Network Protocols*, NPSec '06, 2006.

[110] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. Katz, "Listen and whisper: Security mechanisms for BGP", in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation*, NSDI '04, 2004.

[111] G. Siganos and M. Faloutsos, "Neighborhood watch for Internet routing: Can we improve the robustness of Internet routing today?", in *Proceedings of the IEEE International Conference on Computer Communications*, INFOCOM '07, 2007.

[112] J. Karlin, "Pretty Good BGP: Improving BGP by cautiously adopting routes", in *Proceedings of the International Conference on Network Protocols*, ICNP '06, 2006.

[113] M. Zhang, B. Liu, and B. Zhang, "Safeguarding Data Delivery by Decoupling Path Propagation and Adoption", in *Proceedings of the International Conference on Com-*

puter Communications, INFOCOM '10, 2010.

[114] T. Bates, R. Bush, T. Li, and Y. Rekhter, "DNS-based NLRI origin AS verification in BGP", Internet-Draft – work in progress 00, IETF, July 1998.

[115] M. Lepinski and S. Turner, "An Overview of BGPSEC", Internet-Draft – work in progress 05, IETF, July 2014.

[116] "RPKI Validator". `http://www.ripe.net/lir-services/resource-management/certification/tools-and-resources`, July 2014. RIPE NCC.

[117] M. Wählisch, F. Holler, T. C. Schmidt, and J. H. Schiller, "RTRlib: An Open-Source Library in C for RPKI-based Prefix Origin Validation", in *Proceedings of the USENIX Security Workshop*, CSET '13, 2013.

[118] M. Wählisch, "RPKI Validator Firefox Add-On". `https://addons.mozilla.org/en-US/firefox/addon/rpki-validator/`, August 2013.

[119] M. Wählisch, F. Holler, T. C. Schmidt, and J. H. Schiller, "One Day in the Life of RPKI". `https://labs.ripe.net/Members/waehlisch/one-day-in-the-life-of-rpki`, December 2011. RIPE Labs.

[120] "RPKI ROA Certification Statistics". `http://certification-stats.ripe.net/`, January 2015. RIPE NCC.

[121] H. Chan, D. Dash, A. Perrig, and H. Zhang, "Modeling Adoptability of Secure BGP Protocol", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '06, 2006.

[122] S. Goldberg, S. Halevi, A. D. Jaggard, V. Ramachandran, and R. N. Wright, "Rationality and Traffic Attraction: Incentives for Honest Path Announcements in BGP", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '08, 2008.

[123] P. Gill, M. Schapira, and S. Goldberg, "Let the Market Drive Deployment: A Strategy for Transitioning to BGP Security", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '11, 2011.

[124] A. Boldyreva and R. Lychev, "Provable Security of S-BGP and Other Path Vector Protocols: Model, Analysis and Extensions", in *Proceedings of the ACM International Conference on Computer and Communications Security*, CCS '12, 2012.

[125] R. Lychev, S. Goldberg, and M. Schapira, "BGP Security in Partial Deployment: Is the Juice Worth the Squeeze?", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '13, 2013.

[126] "The Implications of RPKI Certificate Revokation". `https://www.ripe.net/ripe/mail/archives/address-policy-wg/2011-May/005747.html`, May 2011. RIPE Address Policy Working Group (mailing list).

[127] G. Huston and R. Bush, "Securing BGP and SIDR", *IETF Journal*, vol. 7, no. 1, pp. 1815–1828, 2011.

[128] E. Jasinska, N. Hilliard, R. Raszuk, and N. Bakker, "Internet Exchange Route Server", Internet-Draft 05, IETF, June 2014.

[129] "Exa-Networks/exabgp". `https://github.com/Exa-Networks/exabgp/`, 2010. Exa Networks.

[130] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, SIGCOMM '00, 2000.

[131] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding BGP Misconfiguration", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, SIGCOMM '02, pp. 3–16, 2002.

[132] M. Caesar, L. Subramanian, and R. H. Katz, "Towards root cause analysis of internet routing dynamics", in *Proceedings of the Berkeley EECS Annual Research Symposium*, 2004.

[133] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs, "Locating Internet routing instabilities", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, SIGCOMM '04, 2004.

[134] J. Wu, Z. M. Mao, J. Rexford, and J. Wang, "Finding a needle in a haystack: Pinpointing significant BGP routing changes in an IP network", in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation*, NSDI '05, 2005.

[135] Y. Huang, N. Feamster, A. Lakhina, and J. J. Xu, "Diagnosing network disruptions with network-wide analysis", in *Proceedings of the ACM International Conference on Mea-*

surement and Modeling of Computer Systems, SIGMETRICS '07, 2007.

[136] N. Feamster and J. Rexford, "Network-wide prediction of BGP routes", *IEEE/ACM Transactions on Networking*, vol. 15, pp. 253–266, April 2007.

[137] R. Oliveira, B. Zhang, D. Pei, and L. Zhang, "Quantifying Path Exploration in the Internet", *IEEE/ACM Transactions on Networking*, vol. 17, pp. 445–458, April 2009.

[138] B. Eriksson, R. Durairajan, and P. Barford, "RiskRoute: A Framework for Mitigating Network Outage Threats", in *Proceedings of the International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '13, 2013.

[139] M. Luckie, "Spurious Routes in Public BGP Data", *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 44, pp. 14–21, July 2014.

[140] D. G. Anderson, H. Balakrishnan, M. F. Kaawhoek, and R. Morrisi, "Resilient Overlay Networks", in *Proceedings of the ACM Symposium on Operating Systems Principles*, SOSP '01, 2001.

[141] F. Georgatos, F. Gruber, D. Karrenberg, M. Santcroos, A. Susanj, U. H., and W. R., "Providing active measurements as a regular service for ISPs", in *Proceedings of the International Conference on Passive and Active Measurement*, PAM '01, 2001.

[142] R. Kompella, J. Yates, A. Greenberg, and A. Snoeren, "Detection and localization of network black holes", in *Proceedings of the IEEE International Conference on Computer Communications*, INFOCOM '07, 2007.

[143] I. Cunha, R. Teixeira, N. Feamster, and C. Diot, "Measurement Methods for Fast and Accurate Blackhole Identification with Binary Tomography", in *Proceedings of the ACM SIGCOMM Conference on Internet Measurements*, IMC '09, 2009.

[144] I. Cunha, R. Teixeira, D. Veitch, and C. Diot, "Predicting and Tracking Internet Path Changes", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, SIGCOMM '11, 2011.

[145] I. Cunha, R. Teixeira, and C. Diot, "Measuring and Characterizing End-to-End Route Dynamics in the Presence of Load Balancing", in *Proceedings of the International Conference on Passive and Active Measurement*, PAM '11, 2011.

[146] E. Katz-bassett, I. Cunha, H. V. Madhyastha, C. Scott, V. Valancius, T. Anderson, D. R. Choffnes, N. Feamster, and A. Krishnamurthy, "LIFEGUARD: practical repair of persistent route failures", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures and Protocols for Computer Communica-*

tions, SIGCOMM '12, 2012.

[147] T. Flach, E. Katz-Bassett, and R. Govindan, "Quantifying Violations of Destination-based Forwarding on the Internet", in *Proceedings of the ACM SIGCOMM Conference on Internet Measurements*, IMC '12, 2012.

[148] L. Quan, J. Heidemann, and Y. Pradkin, "Trinocular: Understanding Internet Reliability Through Adaptive Probing", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, SIGCOMM '13, 2013.

[149] H. Yang and S. S. Lam, "Collaborative Verification of Forward and Reverse Reachability in the Internet Data Plane", in *Proceedings of the IEEE International Conference on Network Protocols*, ICNP '14, 2014.

[150] N. Feamster, D. G. Andersen, H. Balakrishnan, and M. F. Kaashoek, "Measuring the effects of Internet path faults on reactive routing", in *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '03, 2003.

[151] J. Teixeira and J. Rexford, "A measurement framework for pin-pointing routing changes", in *Proceedings of the ACM SIGCOMM workshop on Network Troubleshooting*, NeT '04, 2004.

[152] M. Zhang, C. Zhang, V. Pai, L. Peterson, and R. Wang, "PlanetSeer: Internet path failure monitoring and characterization in wide-area services", in *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*, OSDI '04, 2004.

[153] F. Wang, Z. M. Mao, J. Wang, L. Gao, and R. Bush, "A measurement study on the impact of routing events on end-to-end Internet path performance", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, SIGCOMM '06, 2006.

[154] A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot, "NetDiagnoser: Troubleshooting network unreachabilities using end-to-end probes and routing data", in *Proceedings of the International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '07, 2007.

[155] E. Katz-Bassett, H. V. Madhyastha, J. P. John, A. Krishnamurthy, D. Wetherall, and T. E. Anderson, "Studying Black Holes in the Internet with Hubble", in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation*, NSDI '08, 2008.

[156] R. Bush, O. Maennel, M. Roughan, and S. Uhlig, "Internet Optometry: Assessing the Broken Glasses in Internet Reachability", in *Proceedings of the ACM SIGCOMM Conference on Internet Measurements*, IMC '09, 2009.

[157] U. Javed, I. Cunha, D. Choffnes, E. Katz-Bassett, T. Anderson, and A. Krishnamurthy, "PoiRoot: Investigating the Root Cause of Interdomain Path Changes", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, SIGCOMM '13, 2013.

[158] W. W. Fok, X. Luo, R. Mok, W. Li, Y. Liu, E. W. Chan, and R. K. Chang, "MonoScope: Automating network faults diagnosis based on active measurements", in *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management*, IM '13, 2013.

[159] N. Feamster and H. Balakrishnan, "Detecting BGP configuration faults with static analysis", in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation*, NSDI '05, 2005.

[160] R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren, "IP fault localization via risk modeling", in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation*, NSDI '05, 2005.

[161] D. Turner, K. Levchenko, A. C. Snoeren, and S. Savage, "California Fault Lines: Understanding the Causes and Impact of Network Failures", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, SIGCOMM '10, 2010.

[162] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "An Analysis of BGP Multiple Origin AS (MOAS) Conflicts", in *Proceedings of the ACM SIGCOMM Workshop on Internet Measurement*, IMW '01, 2001.

[163] K.-W. Chin, "On the characteristics of BGP multiple origin AS conflicts", in *Proceedings of the Australasian Telecommunication Networks and Applications Conference*, ATNAC '07, 2007.

[164] Jacquemart, Quentin and Urvoy-Keller, Guillaume and Biersack, Ernst W., "A longitudinal study of BGP MOAS prefixes", in *Proceedings of the International Workshop on Traffic Monitoring and Analysis*, TMA '14, 2014.

[165] V. Khare, Q. Ju, and B. Zhang, "Concurrent Prefix Hijacks: Occurrence and Impacts", in *Proceedings of the ACM SIGCOMM Conference on Internet Measurements*, IMC '12, 2012.

[166] N. Feamster, J. Jung, and H. Balakrishnan, "An Empirical Study of "Bogon" Route Advertisements", *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 35, pp. 63–70, January 2005.

[167] R. Hiran, N. Carlsson, and P. Gill, "Characterizing Large-Scale Routing Anomalies: A Case Study of the China Telecom Incident", in *Proceedings of the International Conference on Passive and Active Measurement*, PAM '13, 2013.

[168] M. Wählisch, O. Maennel, and T. C. Schmidt, "Towards Detecting BGP Route Hijacking Using the RPKI", *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 42, pp. 103–104, August 2012.

[169] H. Ballani, P. Francis, and X. Zhang, "A study of prefix hijacking and interception in the Internet", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '07, 2007.

[170] Clint Hepner and Earl Zmijewski, "Defending against BGP Man-in-the-middle attacks". `https://www.renesys.com/tech/presentations/pdf/blackhat-09.pdf`, 2009. Presentation at Blackhat, Dyn Research.

[171] A. Pilosov and T. Kapela, "Stealing the Internet: An Internet-scale Man in the Middle Attack". `http://www.defcon.org/images/defcon-16/dc16-presentations/defcon-16-pilosov-kapela.pdf`, 2008. Presentation at DEFCON 16.

[172] Y. Zhang and M. Pourzandi, "Studying impacts of prefix interception attack by exploring BGP AS-PATH prepending", in *Proceedings of the IEEE International Conference on Distributed Computing Systems*, ICDCS '12, 2012.

[173] M. Lad, R. Oliveira, B. Zhang, and L. Zhang, "Understanding Resiliency of Internet Topology Against Prefix Hijack Attacks", in *Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks*, DSN '07, 2007.

[174] Z. Zhang, Y. Zhang, Y. C. Hu, and Z. M. Mao, "Practical Defenses Against BGP Prefix Hijacking", in *Proceedings of the International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '07, 2007.

[175] P.-A. Vervier, O. Thonnard, and M. Dacier, "Mind Your Blocks: On the Stealthiness of Malicious BGP Hijacks", in *Proceedings of the ISOC Symposium on Network and Distributed Systems Security*, NDSS '15, 2015.

[176] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, "PHAS: A prefix hijack alert system", in *Proceedings of the USENIX Security Symposium*, vol. 15 of *USENIX-SS '06*,

2006.

[177] J. Qiu and L. Gao, "Detecting bogus BGP route information: going beyond prefix hijacking", in *Proceedings of the 3rd International Conference on Security and Privacy in Communication Networks*, SecureComm '07, 2007.

[178] T. Qiu, L. Ji, D. Pei, J. Wang, J. J. Xu, and H. Ballani, "Locating Prefix Hijackers using LOCK", in *Proceedings of the USENIX Security Symposium*, USENIX-SS '09, 2009.

[179] J. Li, T. Ehrenkranz, and P. Elliott, "Buddyguard: A buddy system for fast and reliable detection of IP prefix anomalies", in *Proceedings of the International Conference on Network Protocols*, ICNP '12, 2012.

[180] "Routing anomalies". `http://www.bgpstream.com`, 2015. BGPmon Network Solutions Inc.

[181] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis, "A light-weight distributed scheme for detecting IP prefix hijacks in real-time", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '07, 2007.

[182] M. Tahara, N. Tateishi, T. Oimatsu, and S. Majima, "A Method to Detect Prefix Hijacking by Using Ping Tests", in *Proceedings of the Asia-Pacific Symposium on Network Operations and Management: Challenges for Next Generation Network Operations and Service Management*, APNOMS '08, 2008.

[183] J. W. Mickens, J. R. Douceur, W. J. Bolosky, and B. D. Noble, "StrobeLight: Lightweight Availability Mapping and Anomaly Detection", in *Proceedings of the USENIX Annual Technical Conference*, USENIX '09, 2009.

[184] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush, "iSPY: Detecting IP prefix hijacking on my own", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '08, 2008.

[185] X. Hu and Z. M. Mao, "Accurate real-time identification of IP prefix hijacking", in *Proceedings of the IEEE Symposium on Security and Privacy*, IEESSP '07, 2007.

[186] S.-C. Hong, H.-T. Ju, and J. W. Hong, "IP prefix hijacking detection using idle scan", in *Proceedings of the Asia-Pacific Symposium on Network Operations and Management: Management Enabling the Future Internet for Changing Business and New Computing Services*, APNOMS '09, 2009.

[187] S.-C. Hong, J.-K. Hong, and H. Ju, "IP prefix hijacking detection using the collection

of AS Characteristics", in *Proceedings of the Asia-Pacific Symposium on Network Operations and Management: Management Enabling the Future Internet for Changing Business and New Computing Services*, APNOMS '11, 2011.

[188] X. Shi, Y. Xiang, Z. Wang, X. Yin, and J. Wu, "Detecting prefix hijackings in the Internet with argus", in *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, IMC '12, 2012.

[189] P.-A. Vervier and O. Thonnard, "SpamTracer: How Stealthy Are Spammers?", in *Proceedings of the International Workshop on Traffic Monitoring and Analysis*, TMA '13, 2013.

[190] A. Khan, H. chul Kim, T. Kwon, and Y. Choi, "A Comparative Study on IP Prefixes and Their Origin Ases in BGP and the IRR", *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 43, pp. 16–24, July 2013.

[191] N. Vratonjic, J. Freudiger, V. Bindschaedler, and J.-P. Hubaux, "The inconvenient truth about Web certificates", in *Proceedings of the Workshop on the Economics of Information Security*, WEIS '11, 2011.

[192] R. Holz, L. Braun, N. Kammenhuber, and G. Carle, "The SSL landscape—a thorough analysis of the X.509 PKI using active and passive measurements", in *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, IMC '11, 2011.

[193] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast Internet-wide Scanning and Its Security Applications", in *Proceedings of the USENIX Security Symposium*, USENIX-SS '13, 2013.

[194] R. Graham, "Masscan: the entire Internet in 3 minutes". `http://blog.erratasec.com/2013/09/masscan-entire-internet-in-3-minutes.html`, September 2013. Blog post.

[195] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman, "Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices", in *Proceedings of the USENIX Security Symposium*, USENIX-SS '12, 2012.

[196] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman, "Analysis of the HTTPS certificate ecosystem", in *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, IMC '13, 2013.

[197] R. Holz, J. Amann, O. Mehani, M. Wachs, and M. A. Kafaar, "TLS in the wild—An Internet-wide analysis of TLS-based protocols for electronic communication", in *Proceedings of the ISOC Symposium on Network and Distributed Systems Security*, NDSS

'16, 2016.

[198] B. H. Trammell and E. Boschi, "Bidirectional Flow Export Using IP Flow Information Export (IPFIX)", RFC 5103, IETF, January 2008.

[199] R. Holz, *Empirical analysis of Public Key Infrastructures and investigation of improvements*. PhD thesis, Technische Universität München, May 2014.

[200] M. Roughan, W. Willinger, O. Maennel, D. Perouli, and R. Bush, "10 Lessons from 10 Years of Measuring and Modeling the Internet's Autonomous Systems", *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 29, no. 9, pp. 1810–1821, 2011.

[201] R. Motamedi, R. Rejaie, and W. Willinger, "A Survey of Techniques for Internet Topology Discovery", *IEEE Communications Surveys and Tutorials*, vol. 17, no. 2, pp. 1044–1065, 2015.

[202] L. C. Freeman, "A set of measures of centrality based on betweenness", *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[203] R. Tarjan, "A hierarchical clustering algorithm using strong components", *Information Processing Letters*, vol. 14, pp. 26–29, March 1982.

[204] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks", *Nature*, vol. 393, pp. 440–442, June 1998.

[205] M. E. J. Newman, "The Structure and Function of Complex Networks", *SIAM Review*, vol. 45, pp. 167–256, March 2003.

[206] K. L. Calvert, M. B. Doar, A. Nexion, and E. W. Zegura, "Modeling Internet Topology", *IEEE Communications Magazine*, vol. 35, pp. 160–163, June 1997.

[207] E. W. Zegura, K. L. Calvert, and M. J. Donahoo, "A quantitative comparison of graph-based models for Internet topology", *IEEE/ACM Transactions on Networking*, vol. 5, pp. 770–783, December 1997.

[208] H. Haddadi, S. Uhlig, A. Moore, R. Mortier, and M. Rio, "Modeling Internet topology dynamics", *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 38, pp. 65–68, April 2008.

[209] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology", in *Proceedings of the ACM SIGCOMM International Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, SIGCOMM '99, 1999.

[210] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "Network

topologies, power laws, and hierarchy", *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 32, pp. 76–76, January 2002.

[211] D. Alderson, L. Li, W. Willinger, and J. C. Doyle, "Understanding Internet topology: principles, models, and validation", *IEEE/ACM Transactions on Networking*, vol. 13, pp. 1205–1218, February 2005.

[212] P. Mahadevan, D. Krioukov, M. Fomenkov, X. Dimitropoulos, k c claffy, and A. Vahdat, "The Internet AS-level topology: three data sources and one definitive metric", *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 36, pp. 17–26, January 2006.

[213] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, kc claffy, and G. Riley, "AS relationships: inference and validation", *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 37, pp. 29–40, January 2007.

[214] C. Hu, K. Chen, Y. Chen, and B. Liu, "Evaluating Potential Routing Diversity for Internet Failure Recovery", in *Proceedings of the IEEE International Conference on Computer Communications*, INFOCOM '10, 2010.

[215] R. Anwar, H. Niaz, D. Choffnes, Ítalo Cunha, P. Gill, and E. Katz-Bassett, "Investigating Interdomain Routing Policies in the Wild", in *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, IMC '15, 2015.

[216] J. E. Hopcroft and J. D. Ullman, *Introduction To Automata Theory, Languages, And Computation*. Addison-Wesley, 1990.

[217] R. D. L. Briandais, "File searching using variable length keys", in *Western joint computer conference*, IRE-AIEE-ACM '59, 1959.

[218] B. W. Watson, "A Taxonomy of Finite Automata Minimization Algorithms", Computing Science Note 44, Eindhoven University of Technology, 1993.

[219] J. Daciuk, S. Mihov, B. Watson, and R. Watson, "Incremental Construction of Minimal Acyclic Finite-State Automata", *Computational Linguistics–Special issue on finite-state methods in NLP*, vol. 26, no. 1, pp. 3–16, 2000.

[220] P. Gill, M. Schapira, and S. Goldberg, "A Survey of Interdomain Routing Policies", *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 44, no. 1, pp. 28–34, 2013.

[221] RIPE NCC, "RIPE Database Update Reference Manual". `http://www.ripe.net/data-tools/support/documentation/RIPEDatabaseUpdateManual20140425_edit.pdf`, April 2014. Manual.

[222] D. Meyer, "University of Oregon RouteViews Project". `http://www.routeviews.org`,

2005. Archived data.

[223] H. Yan, R. Oliveira, K. Burnett, D. Matthews, L. Zhang, and D. Massey, "BGPmon: A real-time, scalable, extensible monitoring system", in *Proceedings of the Cybersecurity Applications and Technologies Conference for Homeland Security*, CATCH '09, 2009.

[224] I. Alexa Internet, "The top 1 million sites on the web". `http://s3.amazonaws.com/alexa-static/top-1m.csv.zip`, 2015. Archived data.

[225] UCEPROTECT-Orga, "UCEPROTECT-Level 1". `http://www.uceprotect.net`, August 2015. Archived data.

[226] X. Cai, J. Heidemann, B. Krishnamurthy, and W. Willinger, "Towards an AS-to-Organization Map", in *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, IMC '10, 2010.

[227] J. Cowie, "The New Threat: Targeted Internet Traffic Misdirection". `http://research.dyn.com/2013/11/mitm-internet-hijacking/`, November 2013. Blog post.

[228] R. F. Guilmette, "AS22558 - Routing apparently hijacked space". `http://seclists.org/nanog/2010/Oct/412`, October 2010. NANOG mailing list archive.

[229] T. Griffin and G. Huston, "BGP Wedgies", RFC 4264, IETF, November 2005.

[230] L. Benkis, "Practical BGP security: architecture, techniques and tools". `http://www.renesys.com/tech/notes/WP_BGP_rev6.pdf`, 2008.

[231] D. Spirin, "Prefix hijacking by Michael Lindsay via Internap". `http://mailman.nanog.org/pipermail/nanog/2011-August/039379.html`, August 2011. NANOG mailing list archive.

[232] S. P. Ltd., "The Spamhaus Project". `http://www.spamhaus.org/`, 2015.

[233] D. Spirin, "Prefix hijacking by Michael Lindsay via Internap". `http://mailman.nanog.org/pipermail/nanog/2011-August/039568.html`, 2011. NANOG mailing list archive.

[234] U. of Kent School of Computing, "The UK Mirror Servcie". `http://www.mirrorservice.org/sites/ftp.ripe.net/ripe/dbase/split/`, 2013. Archived data.

[235] RIPE, "Anti-Abuse Working Group". `http://lists.ripe.net/pipermail/anti-abuse-wg/2011-July/000838.html`, 2011. Mailing list archive.

[236] S. P. Ltd., "The Register of Known Spam Operations (ROKSO)". `http://www.`

spamhaus.org/rokso/sbl_archived/SPM792/zombies, 2015.

[237] RIPE, "The Internet Archive". http://www.archive.org/, 2011. Archived web sites.

[238] C. Câmpeanu, A. Paun, and S. Yu, "An efficient algorithm for constructing minimal cover automata for finite languages", *International Journal of Foundations of Computer Science*, vol. 13, no. 1, pp. 83–97, 2002.

[239] C. Câmpeanu, A. Paun, and J. R. Smith, "Incremental construction of minimal deterministic finite cover automata", *Theoretical Computer Science*, vol. 363, no. 2, pp. 135–148, 2006.

[240] E. S. Ljapin and A. E. Evseev, *The theory of partial algebraic operations*, vol. 414. Springer Science & Business Media, 2013.

[241] J. L. Sobrinho, "Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet", in *Proc. of IEEE INFOCOM*, 2001.

[242] J. L. Sobrinho, "Network routing with path vector protocols: Theory and applications", in *Proc. of ACM SIGCOMM*, pp. 49–60, 2003.

[243] J. L. Sobrinho, "An algebraic theory of dynamic network routing", *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 1160–1173, 2005.

[244] T. G. Griffin and J. L. Sobrinho, "Metarouting", *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 35, no. 4, pp. 1–12, 2005.

[245] C.-K. Chau, R. J. Gibbens, and T. G. Griffin, "Towards a Unified Theory of Policy-Based Routing.", in *Proc. of IEEE INFOCOM*, 2006.

A Appendix A
Spring-Based Geolocation

NOTE *This chapter contains prior publication.*

Published in **Proceedings of the IEEE/IFIP Symposium on Network Operations and Management (NOMS), 2012.**
The following paper constitutes prior publication [1]. No changes are made.

SUMMARY We propose *Spring-Based Geolocation (SBG)*, a measurement-based approach to estimate the geographic location of IP addresses and corresponding hosts connected to the Internet. Our technique utilizes a set of carefully calibrated measurement nodes that carry out latency measurements towards target hosts. The resulting delay data is translated to distances and mapped onto springs in a spring system. An equilibrium point in this system represents an estimate for the actual location of target hosts.

In our evaluation, we utilize 20 well-known landmark hosts connected to the PlanetLab network and localize each of them. The resulting mean error of 73.4 km outperforms that of state-of-the-art by 19.4%–51.3%. Based on lessons learned from our experiment, we outline an extended approach that accounts for intermediate routers and indirect routing paths.

Spring-Based Geolocation

Stephan M. Günther *
Dept. of Computer Science
Technische Universität München
Munich, Germany
Email: guenther@in.tum.de

Johann Schlamp
Dept. of Computer Science
Technische Universität München
Munich, Germany
Email: schlamp@in.tum.de

Georg Carle
Dept. of Computer Science
Technische Universität München
Munich, Germany
Email: carle@in.tum.de

***Abstract*—Given an IP address, it is a challenging task to obtain its geographic location. Besides approaches which associate coordinates with IP addresses in a predominantly static way, there are also measurement based approaches that exploit the correlation between the propagation delay of signals and round trip times of probe packets. We analyze multiple approaches solely based on delay measurements, i. e. without the use of third-party knowledge, and obtain mean errors of just under 100 km.**

In this paper, we propose a new model for IP geolocation which combines the strengths of different previous techniques and reduces IP geolocation to the problem of finding equilibrium points in a spring system. Our approach, called *Spring-Based Geolocation (SBG)*, is able to reduce the mean error to less than 75 km in our experiments without adding significant complexity. In fact, our model allows for additional data sources in a natural way, which has the potential to further improve results.

I. Introduction

There are static and measurement based approaches for IP geolocation. Static approaches rely on positioning information obtained from databases, i. e. geographic coordinates of the target itself or of neighboring hosts needed during the geolocation process. *HostIP* [1] is an important community-based approach providing a publicly accessible database. A fully vendor-based service is *MaxMind GeoIP* [2]. The major drawback of such approaches is their dependency on third-party knowledge, which is in part unreliable [9]. For instance, many IP addresses that belong to the German academic research network (DFN) are mapped to its administrative headquarter although their actual location is elsewhere. With actively measured round trip times (RTTs) from a set of nodes with known location – called *landmarks* – to an unknown target, it is possible to estimate the distance between these landmarks and the target. In this paper we present a new model for geolocation and show that it is capable to outperform similar fundamental techniques. For transparency, we provide our data set and implementations [3].

The rest of this paper is organized as follows: Section II gives an overview of related work, mostly focused on measurement-based approaches. The theoretical model of our approach is explained in Section III. Empirical results are given in Section IV and compared to related work. An extension to our model that allows for topological information is outlined in Section V. Conclusions and perspectives of future research are given in Section VI.

* Author's work partially funded by EU grant FP7-224619 (ResumeNet)

II. Related Work

Two basic approaches are *Shortest Ping (SP)* and *GeoPing (GP)* [8], which essentially determine the landmark lying closest to a target in terms of delay. This landmark's position is returned as an estimate. For SP the minimum RTT is decisive, whereas GP compares delay vectors resulting in a combined vote how a target is seen by all landmarks.

Many other techniques use a calibration phase to determine parameters of a *distance estimator*, which is used to map measured RTTs to geographic distances. For instance, *Constraint-Based Geolocation (CBG)* [6] measures RTTs to nodes whose locations are known. In this way a set of RTTs and corresponding distances is obtained. These are used to determine slope and intercept of a linear distance estimator that bounds all samples above. The slope of this estimator basically represents a conversion factor between speed of light and the effective signal propagation speed as observed by delay measurements. When a new target is to be located, each landmark measures its RTT to the unknown node and obtains an upper bound for the distance. This defines a circular feasible region centered at the position of the respective landmark. By intersecting the regions of all landmarks, a comparatively small geographic area is obtained wherein the target must reside.

A modification of CBG is known as *Speed of Internet (SOI)* [7]. Here, a constant conversion factor is assumed without previous calibration. One would suspect that SOI is clearly inferior to CBG. However, results in Section IV indicate that SOI yields competitive results. CBG can be further improved by leveraging buffering delay estimation – a technique known as *GeoBuD* [5]. For that purpose, path measurements to the target are used to obtain delay estimates between intermediate hops. These are used to estimate the overall buffering delay along paths from landmarks to a target. Subtracting these estimates from measured end-to-end delays further constrains the geographic area wherein the target has to be located. As a side effect, GeoBuD provides an approximation for the locations of all intermediate routers.

Such per-hop delays are also used by *Topology-Based Geolocation (TBG)* [7]. Here, estimates for the distances between hops along a path from some landmark to the target are derived. Given initial position estimates for those hops, relaxed equality constraints are imposed on the corresponding per-hop distances. Minimizing the sum of errors jointly optimizes the position estimates of all nodes, including the actual target.

TABLE I: Data sources used by various geolocation approaches.

Data source	SP, GP, CBG, SOI, **SBG**	GeoBuD, TBG	Octant, Wang et al.
End-to-end delays	+	+	+
Topological information	-	+	+
Previous localizations	-	+	+
DNS naming patterns	-	-	+
Third-party databases	-	-	+

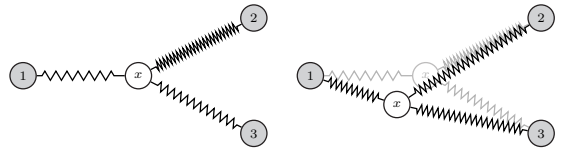
Linear distance estimators as used by CBG cannot exploit non-linear correlations between delay and distance. A different class of estimators was introduced in [12] with *Statistical Geolocation (SG)*: a kernel density estimator is applied to the data obtained from calibration to estimate the joint probability mass function of delay and distance. To estimate the target's position, a custom steepest ascend algorithm is used to jointly optimize the mass functions of all landmarks conditioned on RTTs obtained by probing the respective target. This algorithm relies on an adaptive stepsize method that is not given. Due to this problem, we were unable to obtain reproducible results with SG. However, we adapted the idea of kernel density estimators and combined it with our approach. Results are given in Section IV.

The *Octant* framework [11] takes another approach in estimating distances. By introducing negative constraints, geographical regions are defined wherein a target *cannot* be located. With help of different data sources like DNS naming patterns and by considering last-hop delays through the concept of height vectors, this leads to a more precise description of geographically bounded regions around targets, and thereby to significantly better results.

Hybrid geolocation techniques based on the evaluation of databases, augmented with active measurements, achieve the best results so far, e. g. Wang et al. in [10], but also use more information sources. Table I summarizes relevant data sources and groups the presented approaches accordingly. In this paper, we compare our approach to those that incorporate the same data sources (namely measured end-to-end delays only). In particular, we try to fully explore the potential of such fundamental techniques. Aware of the improvements achieved by approaches that use additional sources of information, we outline a natural extension of our model in Section V.

III. Model

The idea of our approach is to reduce the problem of IP geolocation to finding equilibrium points in a spring system. Using a spring system for modeling network delays was first proposed by the authors of Vivaldi [4]. Here, a spring system is used to *predict* time variant delays in peer-to-peer networks. By contrast, our approach translates *measured* delays to distances and determines a stable solution. In our model, each landmark $i \in \mathcal{L}$ is the anchor of a spring (i, x) which is connected to target x. The equilibrium length of (i, x) would correspond to the great circle distance d_{ix} which is unknown. Therefore, each landmark uses a distance estimator $\tilde{d}_{ix}(\rho_{ix})$. Furthermore, each spring is assigned an individual spring constant k_i indicating the reliability of the distance estimate. The target's position estimate $\tilde{\boldsymbol{r}}_x$ is given by an equilibrium point of the spring system, i. e. the point where forces of displaced springs mutually cancel out (see Figure 1).



(a) Instable initial position (b) Equilibrium position

Fig. 1: Spring system which consists of target x and three landmarks $\mathcal{L} = \{1, 2, 3\}$. The springs are displaced (a) and thus push the loose mass x away. At an equilibrium point (b), the forces on x mutually cancel out.

Note, that our model is not restricted to a specific estimator. In Section IV, we also provide results for distance estimators based on probability mass functions using a Gaussian kernel. Description thereof can be found in [12]. In the following we present a method to determine the spring parameters based on the least squares method. Afterwards, we briefly outline how the spring system can be solved efficiently.

A. Spring parameters

Our approach consists of two phases: calibration and localization. During the calibration phase, landmarks $i \in \mathcal{L}$ measure RTTs ρ_{ij} to all other landmarks $j \in \mathcal{L}\backslash\{i\}$. Since the position of landmarks is known, we obtain a vector of RTTs $\boldsymbol{\rho}_i$ and associated distances $\boldsymbol{d}_i$ for each landmark. For notational brevity we omit the landmark index i wherever applicable.

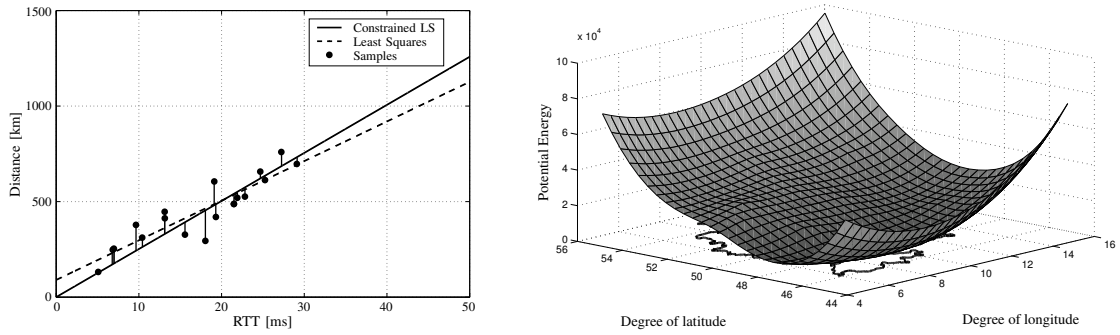
With the vectors $\boldsymbol{d}$ and $\boldsymbol{\rho}$ each landmark determines a linear distance estimator

$$\tilde{d}_x(\rho_x) = \alpha\rho_x + \beta, \tag{1}$$

which is unique to this landmark. To find the vector of parameters $\boldsymbol{y} = [\alpha\ \beta]^T$ we define the matrix $\boldsymbol{A} = [\boldsymbol{\rho}\ \boldsymbol{1}]$ and solve the constrained least squares problem

$$\boldsymbol{y}^* = \arg\min_{\boldsymbol{y}} \|\boldsymbol{A}\boldsymbol{y} - \boldsymbol{d}\|_2^2, \quad \text{s.t.} \quad \beta \leq 0. \tag{2}$$

The constraint $\beta \leq 0$ is a necessary condition: the dashed line in Figure 2a indicates a distance of roughly $80\,\mathrm{km}$ at a delay of zero which violates laws of physics. Additional constraints on α are superfluous since values larger than speed of light would indicate an error in measurement whereas values close to zero and negative values are precluded by the constraint on β. The solid line in Figure 2a represents the constrained estimator. Note that $\boldsymbol{y}^*$ can be determined analytically, removing the need for LP solvers (see supplemental material provided online [3]). In contrast to [6], we do not impose strict bounds but determine an estimator which is designed to minimize the absolute errors. This is reasonable since the estimates are used as equilibrium length of springs. Regardless of whether the real distance is over- or underestimated, the resulting displacement of a spring gives a quadratic penalty term that is minimized while solving the spring system. Furthermore,



(a) Calibration (b) Localization

Fig. 2: (a) Linear distance estimator and empirical standard deviation as measure of quality. (b) Plot of the objective function over a geographic map.

the estimator accounts for a constant source of delay specific to each landmark through the intercept β. The fact that the signal propagation delay makes up only a fraction of the RTT is captured by the slope α.

Regarding the stiffness of springs, consider the following situation: If samples (ρ_j, d_j) are close to the linear estimator, this is an indication that the landmark yields valuable estimates. The vertical solid lines in Figure 2a indicate the deviation of samples from the estimator. This motivates the usage of the reciprocal of the empiric standard deviation between the estimator and data samples that is given by

$$k_i = \sqrt{\frac{|\mathcal{L}| - 1}{\sum_{j \neq i} (d_{ij} - \tilde{d}_{ij})^2}}. \tag{3}$$

The stiffness of a spring can be thought of a weighting factor which allows to influence the contribution of individual springs according to the reliability of measurements.

B. Solving the spring system

Given the parameterization of the spring system for an actual target, its position estimate is given by an equilibrium point of the spring system, i. e. coordinates $\boldsymbol{r}_x$ for the target such that forces mutually cancel out. Consider Figure 1 again: the total force $\boldsymbol{F}$ on the loose target at position $\boldsymbol{r}_x$ is given by

$$\boldsymbol{F}(\boldsymbol{r}_x) = \sum_{i \in \mathcal{L}} \left(\tilde{d}_{ix}(\rho_{ix}) - \|\boldsymbol{r}_i - \boldsymbol{r}_x\|_2 \right) \frac{\boldsymbol{r}_i - \boldsymbol{r}_x}{\|\boldsymbol{r}_i - \boldsymbol{r}_x\|_2}. \tag{4}$$

$\boldsymbol{F}(\boldsymbol{r}_x)$ is a vectorial function and has singularities at $\boldsymbol{r}_x = \boldsymbol{r}_i$. Instead of looking for roots of $\boldsymbol{F}$ we can minimize its primitive, which represents the potential energy stored in the system. The position estimate for a target is then given by

$$\boldsymbol{r}_x^* = \arg\min_{\boldsymbol{r}_x} \sum_{i \in \mathcal{L}} k_{ix} \left(\|\boldsymbol{r}_i - \boldsymbol{r}_x\|_2 - \tilde{d}_{ik}(\rho_{ix}) \right)^2. \tag{5}$$

The objective function in (5) is non-convex and can have local minima rather than a globally unique one. A trivial approach to check for local minima is to start the optimization of (5) at different initial values for $\boldsymbol{r}_x$. So far, we encountered local minima only in rare cases. An example for a system's potential energy is shown in Figure 2b.

IV. Evaluation

We evaluated our approach using 14 active and 6 passive landmarks located in Germany and Switzerland, which we obtained from PlanetLab. Note that due to the calibration phases, all approaches evaluated in this section (except for SOI) are robust against additive delays, e. g. delays incurred by PlanetLab's virtualization. To assure comparability measurement was conducted once and evaluated offline. As the position of our landmarks is known exactly, each of them was considered an unknown target to be located. To avoid corruption of results we excluded the calibration data for the respective landmark from the complete data set while estimating its position. Statistical key data are given in Table II. The results are shown as cumulative distribution in Figure 3.

SBG with the linear distance estimator achieves the smallest mean error and standard deviation of all approaches. The best competitive approach (SOI) exhibits a $24\,\%$ larger mean error. For CBG, the mean error is $33\,\%$ larger. Keeping in mind that SBG uses exactly the same information as CBG does, this is a significant improvement. It is very interesting that SOI slightly outperforms CBG and that GP performs worse than SP. Both observations are confirmed by the results in [7].

Regarding SBG and probabilistic estimators the expectation and median of the conditional PDFs yield very similar results to the linear distance estimators while the mode appears to be a bad choice. These results suggest that probabilistic estimators do not offer substantial benefits compared to linear estimators. However, this demonstrates that SBG is capable to use virtually any distance estimator and does not pose strict demands on it, e. g. expecting upper bounds for the distance.

TABLE II: Statistical results for different approaches.

Approach	Mean	Stddev	minimum	maximum
SBG linear	73.4 km	40.4 km	12.3 km	151.5 km
SBG expect.	74.0 km	37.7 km	29.0 km	137.2 km
SBG median	74.7 km	38.7 km	28.4 km	142.9 km
SBG mode	80.7 km	44.7 km	17.0 km	165.4 km
SOI 4/9	91.1 km	62.0 km	15.6 km	226.8 km
SOI 2/3	95.7 km	56.1 km	10.9 km	237.4 km
CBG	97.2 km	67.1 km	17.2 km	259.0 km
SP	119.7 km	50.7 km	59.4 km	249.5 km
GP	150.6 km	82.5 km	66.5 km	313.6 km

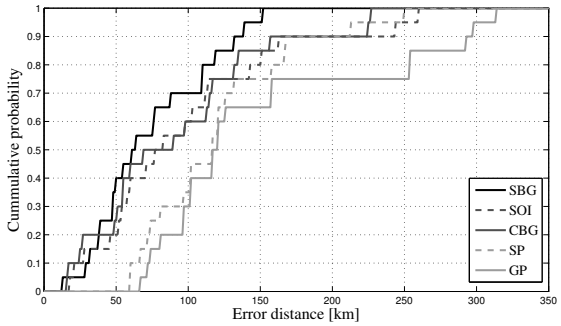


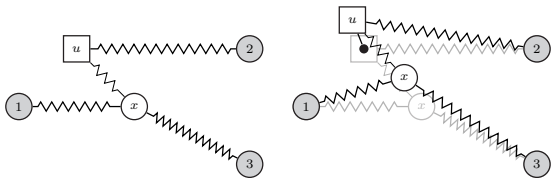
Fig. 3: Cumulative distribution of evaluated approaches. SBG was run with linear distance estimators. For SOI we assumed $4/9 \cdot c$ as propagation speed.

As demonstrated by the results, SBG performs very well using end-to-end delay measurements only. However, we are aware that PlanetLab is a mostly homogenous network and thus an idealized environment. The next section outlines natural extensions of our model to deal with the problem of indirect routing paths in heterogenous networks.

V. Extended Model

The main limitation of our approach is the assumption of rather direct routing paths, which are modeled by springs pointing directly from landmarks to a target. It proved valuable for other approaches to include intermediate nodes. This idea can be adopted by SBG. For instance, a compressed spring like $(2, x)$ in Figure 1 is the result of an overestimated distance between landmark and target. The position estimate is considerably influenced by this spring. Now assume that path measurements, e. g. using traceroute, revealed a node u along the path from 2 to x. Further, assume that unreliable positioning information are available for u, e. g. by previous localization attempts or analysis of DNS naming patterns. We include u into the spring system, which yields the situation depicted in Figure 4. Node u is anchored at its estimated position but allowed to be pushed away given a sufficiently large force. Compared to Figure 1 this topological enhancement changes the resulting equilibrium point significantly.

The spring parameters for $(2, u)$ can be determined directly by probing u. The anchor point of u is given by its position estimate. The spring $(\bullet, u)$ has zero length and its stiffness is chosen according to the reliability of u's position estimate. Finding meaningful parameters for (u, x) is not trivial. Assume that routing decisions do not depend on whether a packet is a request or reply. Then the RTT between two nodes is hardly influenced by the direction of measurement. This also holds for asymmetric routing paths. First experiments indicated that this assumption is reasonable for stateless traffic, e. g. ICMP or UDP probes. Thus, a calibration for u can be obtained by probing it from all other landmarks. This yields all data necessary to derive a distance estimator. The stiffness for the spring originating at u is readily obtained as described in Section III-A. Finally, to determine the equilibrium length of (u, x) we need an estimate of the unknown value ρ_{ux} which



(a) Instable initial position (b) Equilibrium position

Fig. 4: Information about an intermediate node is used (cf. Figure 1).

can be obtained by the difference $\rho_{2x} - \rho_{2u}$ as used in [5]. As this can result in negative values, we propose to bound it below by zero. Further analyses are subject to future work.

VI. Conclusion and Future Work

In this paper we proposed a new model for IP geolocation based on spring systems. By now it utilizes end-to-end delay measurements only and is thus comparable with fundamental approaches like CBG. In our evaluation within PlanetLab we showed that SBG yields more accurate results than those techniques. This suggests that existing approaches do not fully exploit this basic source of information.

Current state-of-the-art approaches use further sources of information, in particular topological data. With a natural extension of our model we outlined how SBG can incorporate this data as well. Further work can be conducted on testing in heterogenous network environments, finding more efficient distance estimators, and also on investigating the applicability of our model to areas beyond geolocation.

References

[1] HostIP GeoTargeting Community. http://www.hostip.info.
[2] MaxMind GeoIP Service. http://www.maxmind.com.
[3] Supplemental material and data: http://geolocation.net.in.tum.de.
[4] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A Decentralized Network Coordinate System. *SIGCOMM Comput. Commun. Rev.*, 34(4):15–26, Aug. 2004.
[5] B. Gueye, S. Uhlig, A. Ziviani, and S. Fdida. Leveraging Buffering Delay Estimation for Geolocation of Internet Hosts. *Proceedings of Networking (IFIP-TC '06)*, pages 319–330, 2006.
[6] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. Constraint-Based Geolocation of Internet Hosts. *IEEE/ACM Transactions on Networking*, 14(6):1219–1232, 2006.
[7] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe. Towards IP Geolocation using Delay and Topology Measurements. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC '06)*, pages 71–84, 2006.
[8] V. N. Padmanabhan and L. Subramanian. An Investigation of Geographic Mapping Techniques for Internet Hosts. *ACM SIGCOMM Computer Communication Review*, 31(4):173–185, 2001.
[9] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye. IP Geolocation Databases: unreliable? *SIGCOMM Comput. Commun. Rev.*, 41:53–56, April 2011.
[10] Y. Wang, D. Burgener, M. Flores, A. Kuzmanovic, and C. Huang. Towards Street-Level Client-Independent IP Geolocation. In *Proceedings of the 8th USENIX conference on Networked systems design and implementation (NSDI '11)*, pages 27–27, 2011.
[11] B. Wong, I. Stoyanov, and E. G. Sirer. Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts. *Proceedings of the 4th USENIX conference on Networked systems design & implementation (NSDI '07)*, pages 313–326, 2007.
[12] I. Youn, B. L. Mark, and D. Richards. Statistical Geolocation of Internet Hosts. In *Proceedings of the 18th International Conference on Computer Communications and Networks (ICCCN '09)*, pages 1–6, 2009.

B Appendix B
Flow-Inspector

NOTE *This chapter contains prior publication.*

Published in **Proceedings of the First IMC Workshop on Internet Visualization (WIV), 2012.**
The following paper constitutes prior publication [2]. No changes are made.

SUMMARY We present *Flow-Inspector*, a versatile toolkit for the visualization of network flow data. This framework utilizes modern web technologies to provide a highly interactive interface that displays traffic information in real-time. Flow-Inspector can be used to study generic network properties as well as specific operational problems. It is extensible and enables both network operators and the scientific community to create new and innovative visualizations that can be immediately put into practice.

Our prototype implementation utilizes a document-based storage backend and provides several ready-to-use visualization modules. These modules include volume-based display formats for traffic flows, force-directed network graphs, and also more sophisticated types of diagrams such as edge-bundled hierarchies or hive plots.

Flow-Inspector: A Framework for Visualizing Network Flow Data using Current Web Technologies

Lothar Braun, Mario Volke, Johann Schlamp, Alexander Klein, Georg Carle
Technische Universität München
Chair for Network Architectures and Services
{braun,volke,schlamp,klein,carle}@net.in.tum.de

ABSTRACT

New web technologies led to the development of browser applications for data analysis. Modern browser engines allow for building interactive real-time visualization applications that enable efficient ways to understand complex data. We present Flow-Inspector, a highly interactive open-source web framework for visualizing network flow data using latest web technologies.

Flow-Inspector includes a backend for processing and storing large-scale network flow data, as well as a JavaScript-based web application capable to display and manipulate traffic information in real-time. This work provides operators with a toolkit to analyze their networks and enables the scientific community to create new and innovative visualizations of traffic data with an extensible framework. We demonstrate the applicability of our approach by implementing several different visualization components that help to identify topological characteristics in network flows.

1. INTRODUCTION

The increasing popularity of web applications has led to numerous W3C standards that specify functionality to build dynamic and interactive web applications. Prominent representatives of such technologies include HTML5 and JavaScript. Those standards provide mechanisms for real-time rendering of 2D graphics and are favored by browser vendors, which leads to advances in rendering speed as browser implementations improve.

As a result of this technological progress, JavaScript-based frameworks for data visualization emerged [1]. Such frameworks already implement a variety of algorithms useful for the data visualization community, which can be particularly applied to study network measurement data. While such algorithms are capable to display large data sets in a human-understandable way, the use of JavaScript enables highly interactive analyses by providing means to manipulate rendered images. This flexibility can lead to additional insights compared to common static visualization approaches.

We present Flow-Inspector, a JavaScript-based web application that applies modern web technologies to visualize network flow data. The systems consists of a backend for preprocessing, aggregation and storage of data, and a frontend that allows for interactive querying and rendering.

This paper targets two audiences alike: operators and researchers. First, operators benefit from our framework when confronted with analyzing traffic flows in their own networks. Several built-in visualization components are available for different use cases, including volume-based and node-based visualization of traffic.

Flow-Inspector provides users with a new traffic visualization approach: hive plots [2] enable novel analyses of network flow data. The analysis frontend supports drill down methods to filter data and an interface for interacting with rendered images. Additional pieces of information can be provided, e.g. tool tips can be requested by hovering over objects.

Second, researchers and developers can profit by Flow-Inspector's extensible framework. It is straightforward to integrate new visualization algorithms while relying on the backend to provide the necessary data. It is even possible to extend the data model without interfering with other visualization components. This technical flexibility allows for rapid development and early visualization of novel data sets.

We organize our paper as follows: Section 2 introduces the design of our framework, with focus on easy extensibility. With Section 3, we discuss built-in visualization algorithms shipped with Flow-Inspector. We compare our approach to related work in Section 4, and conclude the paper in Section 5.

2. DESIGN AND IMPLEMENTATION

Flow-Inspector consists of two main components. A backend that is responsible for preprocessing, aggregating and storing of flow data. While preprocessing the data, it also generates several pre-computed statistics on the data and makes all data available via an HTTP-API.

The second part is a JavaScript application that displays the pre-processed flow information using the methods of current web browsers.

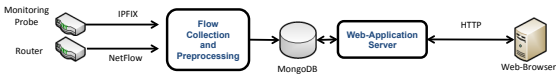


Figure 1: Flow-Inspector data processing chain

2.1 Data Model

Flow-Inspector's primary goal is to visualize traffic measurements that describe how hosts (or networks) communicate with each other. Relevant characteristics of such communication patterns depend on the environment and the user's intended results, i.e. necessary data is potentially unknown from the design's point of view. Traditional flow information, such as the information transported by NetFlow v5 messages might not be sufficient for certain visualization tasks.

Additional information like the results of QoS measurements, application specific information, or additional structural information could be interesting for a user to visualize. Since we cannot anticipate future use cases and do not want to restrict users, we focus our design on extensibility. We take the definition of a flow given by the IPFIX standard as a base for our data model:

A flow is defined as a set of IP packets passing an observation point in the network during a certain time interval. All packets belonging to a particular flow have a set of common properties. [3]

The IPFIX flow definition allows for different types of flows and arbitrary supplemental information. Properties that define a flow could be the IP five tuple but are not limited to these keys.

We provide implementational flexibility by allowing the developer to individually specify flows based on a set of arbitrary keys. Supplemental flow properties can be specified as well. Adding new properties does not require any changes to the core code for implementing new visualization components.

2.2 Backend

The previous section outlined the need for a flexible data model that needs to handle potentially unknown data types. Flow-Inspector's backend supports such indetermined data objects with a document-based database called MongoDB [4]. Documents thereby consist of key-value pairs of arbitrary types, and allow to import any flow information from NetFlow or IPFIX messages into the database. Interaction between the frontend application and the database backend is realized using a JSON data model, that allows the frontend to query any types of data as lists of key-value pairs.

The choice of MongoDB was also motivated by some of its built-in functionality: Mongo supports so-called *sharded* databases which allows to distribute the database onto multiple machines. Furthermore, the database includes built-in support for the MapReduce programming model which allows to parallelize complex database requests onto a sharded database. Besides MapReduce, MongoDB also provides a system called "Aggregation Framework", which provides simple ways to perform aggregation on the stored data. The aggregation framework is compatible with sharded databases which allows to execute the aggregation in parallel on multiple machines.

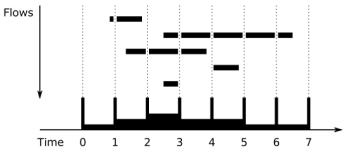


Figure 2: Slicing flows into buckets

All flow data is pre-processed before it is stored in the database, which results in an overall architecture as shown in Figure 1. This pre-processing step includes temporal and spatial flow data aggregation as described in [5], and flow indexing for quick data access. Aggregation techniques are applied based on a given backend configuration, where operators can specify aggregation flow keys for spatial aggregation, and time intervals for temporal aggregation.

Time aggregation is essential in order to allow users to choose visualization intervals and analyze traffic over time. Flow-Inspector thereby integrates mechanisms for temporal aggregation and interval distribution [5]. Each flow is associated with a start and end time, i.e. the time of the first and last packet observed within a flow. The system configuration contains time intervals that define the visualization granularity (e.g. 5/10/30/60 minutes). Flows are sliced to match those intervals, called buckets throughout the rest of this paper, as shown in Figure 2. Flows that share the same aggregation flow keys in the same interval on the other handy aggregated into a single flow.

The stored data can be queried by an HTTP API provided by the server. This API further provides filtering mechanisms to purge data sets based on a client's request. Data rendering is exclusively performed on the client.

2.3 Frontend

Flow-inspector's frontend is implemented as an interactive JavaScript application that is automatically delivered when loading the website in a browser.

The core of this application utilizes D3.js [6], a library that allows data-driven manipulations of the website's document object model (DOM). With D3.js, complex interactive visualizations of arbitrary data can be efficiently build using HTML5, SVG and CSS. Additional JavaScript libraries, e.g. from the projects listed in [1], that provide specialized types of visualization can be

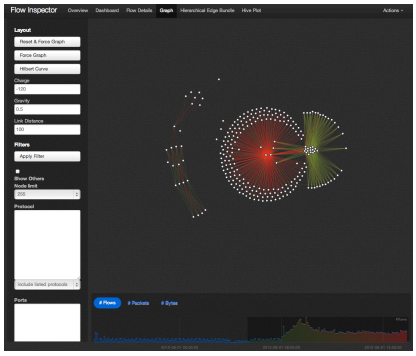


Figure 3: Flow-Inspector: Control Interface

integrated into the application as well.

Flow-Inspector also relies on BackBone.js [7], which provides a model-view-controller design architecture that facilitates extending the system with new visualizations. While providing new view classes that perform data rendering, model classes responsible for fetching data can be re-used. We created several visualization components based on this approach, which will be discussed in the following section.

3. VISUALIZATIONS

One of the most important steps for a visualization process is the proper selection of the parts of the data that should be displayed to the user. Figure 3 shows the control interface of Flow-Inspector that is shared by most of the views of the system. Each view includes three major regions.

The first region is the side bar on the left, and contains the filtering and control options of the view. These include common controls shared by all visualizations, such as fields for filtering for ports, protocols, or IP addresses. Furthermore, each visualization can add it's own controls. The example shows the Force Graph visualization, presented in Section 3.2, which provides some view specific operations that control the graph layout. Each control field provides information on its proper usage with tool tips that are displayed when the component is highlighted.

Time-based selection of traffic is performed in the bottom field of the view. This field contains an time-based overview of the traffic volumes. A time-interval can be selected by clicking on a start-interval and dragging the mouse over the available intervals.

The final and major component displays the flows that have been selected by the controls using one of the implemented visualization techniques. These are described in the remainder of this section.

3.1 Volume-based Visualization

Volume-based representations are the most common technique for presenting the current and past state of network traffic to an operator. They can be used to obtain an initial understanding of the time-dependent dynamics in a network. Time-series-based graphs that show the overall volume of traffic divided per transport protocols are generally available in systems that show network flows. Flow-Inspector supports such standard time-series-based views as well as views that allow to identify dominant hosts or services in the traffic data.

Typical time-series volume graphs, as shown in Figure 4, provide an overview about the number of flows, packets, and bytes observed in the user-defined buckets. They provide timeline views on the x-axes and display the amounts of flows, packets, or bytes split by transport layer protocol on the y-axes.

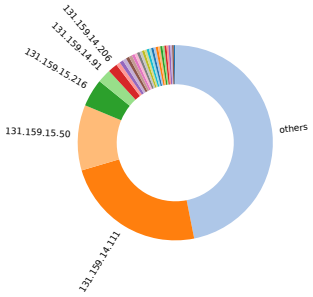


Figure 5: Distributions of host with highest flow counts

Other pieces of helpful information are the global distributions of flows, packets or bytes among IP addresses (hosts) and ports (services). These distributions show the most active hosts and most heavily-used services in the network. Flow-Inspector uses donut charts to represent the share of individual IP addresses or ports in the total amount of traffic, as shown in Figure 5. All IP addresses or ports are sorted by one of the sums available in the pre-computed buckets (flows, packets or bytes). The most active IPs or ports are then shown as individual segments, while all others are summarized into an "others" segment in order to avoid an overcrowded visualization.

An additional host overview graph displays the most active hosts sorted by bytes, packets or flows in table form. Figure 6 shows an example of a flow-based host overview graph. The host overview graph complements the donut view by breaking down its visualization into the different transport layer protocols and comparing only the most active IP addresses or ports.

All views are associated with additional information that is only provided upon interaction between the user and the graphs. The graphs are connected to buttons that allow the user to choose a metric he wants to observe, e.g. distributions by number of flows, packets or bytes. Furthermore, every bar in the time-series and

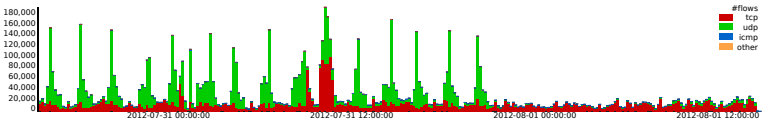


Figure 4: Time series: number of flows over time

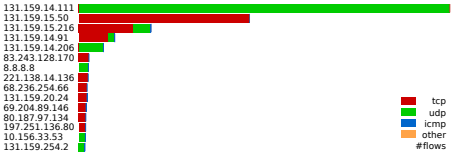


Figure 6: TOP 15 hosts (most flows)

host overview graphs is provided with a tool tip that holds additional information. It is shown as soon as the user hovers over a portion of the graph.

Volume-based graphs can help to find time spans with unusual traffic, e.g. the time of service outages or network attacks. The views are especially helpful in order to identify time intervals that should be further investigated in other views. While such graphs provide oversight of overall traffic volumes, they do not provide insight into communication patterns between nodes.

3.2 Force Graphs

Node-flow graphs show these patterns by representing hosts or networks as nodes, and the communication flows between them as lines that connect the nodes. By adopting the line color depending on the time a given flow was observed, communication patterns and changes in those patterns over time can be made visible.

The human eye requires a meaningful layout in order to derive useful information from such a flow graph. Corresponding layouts should minimize line crossings by grouping the nodes in a way such that connections overlap as little as possible.

One of the most popular approaches to generate graph layouts with respect to visualizations of relationships between entities with little overlap are force-directed layouts. There are various algorithms available, but most of them share the idea of a physics-based, iterative simulation until a power equilibrium is reached.

Flow-Inspector uses the force-graph layout algorithm provided by D3.js, which uses position Verlet integration [8]. Links between nodes are considered as a weak geometric constraint that have a desired length. Initially, nodes are randomly positioned on the canvas resulting in a non-optimal layout. The force-graph algorithm then tries to iteratively optimize the position of all nodes under their geometric constraints, computes new positions and then optimizes them again. An example of a node-flow graph with a force-directed layout can be seen in Figure 7.

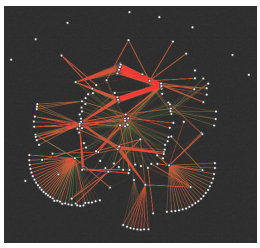


Figure 7: Force Graph

In Flow-Inspector, rendering and calculating of the force graph layout is carried out after a user selects a time interval. All observed flows during this time interval define connections between nodes. The graph is calculated and drawn in multiple iterations, until a steady state is reached or the user aborts the iterative algorithm by clicking onto the graph.

If the user modifies the selected time range changes, which selects other flows between the nodes, all new flows are drawn into the previously calculated force-graph image. This enhances the comparability between the time ranges. By clicking the force button in the control area, the user can derive a new force-layout from the newly loaded flows. This new force-graph can be created based on the previous layout, or based on a new randomized placement of the nodes.

Force-directed layouts can often reveal structures in a graph which might be hard to recognize in trivial layouts. They minimize the length of connections and line crossings and usually lead to graph drawings with a natural look. In contrast to static visualizations of such graphs, Flow-Inspector can attach additional information to its nodes: Hovering over nodes in interesting communication patterns reveals their IP address which can be used for further investigations.

Filtering of data to an interesting subset is nevertheless a crucial part of this visualization technique. If the number of rendered nodes and flows grows to large, the graph will most likely contain many overlapping connections, which decreases the visibility and readability of communication patterns.

3.3 Hierarchical Edge Bundles

A drawback of force graphs is that two nodes are generally connected using a straight line, which can result in unwanted line crossings.

Large-scale data sets with many nodes and connections tend to lead to massive line crossings. Bundle

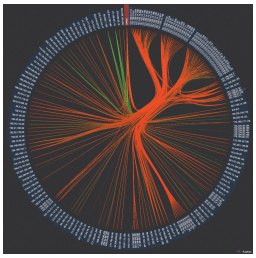


Figure 8: Hierarchical Edge Bundle

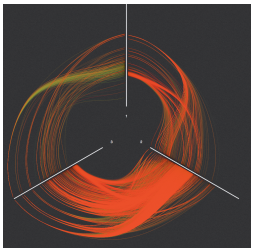


Figure 9: Hive Plot

graphs are one way to group the connections between entities in a more comprehensible way.

The basic idea behind edge bundle graphs is to create bundles of similar connections to increase the visibility of connection structures between end systems. An example for those edge bundles is shown in Figure 8, where entities are organized on a circle. Flow-Inspectors supports this technique with entities that are either IP addresses or networks in CIDR notation.

The layout on the circle is defined by a radial tree layout that groups IP addresses based on membership of a network. IP addresses in the same networks are mapped close to each other with additional space between networks.

Flows between these IPs are drawn as bundled lines using hierarchical edge bundling. The lines are interpolated with a piecewise cubic B-spline. A tension parameter between zero and one allows to control the attraction of the line to its control points, which affects the tightness of the flow bundles.

The graph itself is an interactive graph with several ways to filter and display additional information upon user request: A timeline at the bottom of the page allows a user to select the time range that should be visualized. If multiple buckets are selected, the color of the lines represent the buckets in which corresponding flows have been observed. When rendering the image, all available information for an entity (i.e. number of flows, packets, bytes) is loaded from the database and provided as a tooltip. In addition, if the user hovers over a node, only flows that involve the corresponding IP address are rendered.

3.4 Hive Plots

A relatively new approach to visualize large-scale data are Hive Plots [2]. An example for a hive plot is shown in Figure 9. To the best of our knowledge, no one has previously used hive plots for visualizing Netflow or IPFIX data.

The developers of the hive plot concept criticize force-directed methods for visualizing large data sets as "hairballs", which do not provide much insight into the data. Unreasonable positioning of nodes is criticized as well: the resulting graph is often confusing. Hive plots are a completely different and highly customizable way to make trends visible in a huge set of relations.

A hive plot consists of a predefined number of axes arranged on a radial layout. Flow-Inspector uses three axes for mapping IP addresses or networks. When rendering data, each node needs to be assigned to one of the axes (node-to-axis mapping). Nodes are thereby positioned on axes using some feasible heuristic (position mapping). The links of the graph are drawn as lines between two axes starting and ending at the corresponding node positions. This approach is especially useful in analysis scenarios that aim at getting insight into large communication structures.

Mapping and position of nodes to axes is crucial for the insight that a Hive Plot can provide. Furthermore, any interpretation heavily depends on the specific node-to-axis and position mapping. In Flow-Inspector, those mappings are therefore configurable by the user.

The configuration sidebar contains one input field for each axis in the hive plot. Those input fields expect lists of IP addresses/networks in CIDR notation.

The flows between the specified nodes are drawn as B-splines between two adjacent axes, starting and ending at the position of the nodes. Hive plots can be used to display the traffic relationship between networks mapped to axes.

Operators can asses the amount of traffic flows exchanged between networks. The visualization can also be used to determine whether there are unexpected or interesting traffic flows between parts of the networks. For example, flows can be unexpected due to access rules that should not allow those traffic streams. This allows to identify errors in firewall configurations.

The plots can also help to understand the traffic flows in complex applications with frontend and backend traffic: By mapping clients to one axis, frontend servers to the second axis, and backend servers to the third axis, communication patterns that are caused by frontend traffic can be observed.

Hive plots can also be used to analyze complex load-balancing environments, which often use multiple layers of load-balancing. Clients can be assigned to one or more load-balancing servers by a DNS-based round-robin scheme. The load-balancing services can relay

the incoming connections to a large number of back-end servers, which then respond to the actual client requests. By visually analyzing corresponding scenarios, the quality of a load-balancing process can be estimated.

4. RELATED WORK

The need for visualization of traffic data resulted in many systems with different analysis purposes. Some of them display information on traffic volumes and constituencies. NfSen [9] and FlowScan [10] provide mechanisms for visualizing Netflow data that is generated by monitoring probes, routers, or switches. Their visualization methods concentrate on presenting traffic volumes including information about the used transport protocols.

Ntop [11] is a web-based system for analysing and visualizing traffic information and also focuses on flow traffic analysis. It provides tools for collecting and generating flow information for its visualization process. In addition to standard volume-based visualizations, ntop provides tools for combining flow information with other data sources such as BGP data or IP-based geographical information.

Our system differs from these approaches: NfSen, FlowScan and ntop perform the visualization at server side, providing fixed images to the user. Flow-Inspector on the other hand renders its images in a client's browser. This allows for providing users with a higher level of interaction.

Other approaches do not provide web interfaces for visualizations tasks. Instead, they created full-fledged client applications that perform the rendering. These, usually platform-depended applications, can make use of 3D capabilities of the client systems' graphic cards.

In [12], the authors present the client-based visualization tool FlowVis, which uses the SiLK tools for processing NetFlow data. They provide proof-of-concept visualizations like activity plots, flow edge bundles, and network bytes viewer. Lakkaraju et al. presented NVisionIP [13], a tool for displaying traffic patterns in class-B networks using scatter plots and volume-based visualizations. Yin et al. presented an animated link analysis tool for Netflow data [14], which leverages a parallel coordinate plot to highlight dependencies in the network.

Flow-Inspector has several advantages compared to those approaches. Due to its web-based nature, any computer with a modern browser can be used to interact with Flow-Inspector without any additional software installations required. Another major advantage is Flow-Inspector's extensibility. An active community of JavaScript developers is working on visualization libraries for various purposes. Although such libraries might aim at implementing new visualization techniques for tasks beyond network flow analysis, corresponding approaches can often be adopted for displaying traffic data. Flow-Inspector users can benefit from such developments due to its extensible framework that allows to easily integrate these new visualization libraries.

5. CONCLUSION

In this paper we introduced Flow-Inspector, an interactive web application for dynamic network flow visualization. For network operators, visualization of traffic data is an important tool that can help to understand network characteristics and to identify immediate and long-term problems. We therefore encourage network operators to use our framework.

Our paper further aims at making Flow-Inspector available to the research community. Researchers can use it to build new and innovative visualization tools for network traffic data. The code is available for download at http://flow-inspector.net.in.tum.de, and we invite others to use and extend the system.

6. REFERENCES

[1] "Tools for Data Visualization," http://selection.datavisualization.ch, Visited: Nov. 2012.
[2] M. Krzywinski, I. Birol, S. J. Jones, and M. Marra, "Hive Plots - Rational Approach to Visualizing Networks," *Briefings in Bioinformatics*, Dec. 2011.
[3] B. Claise, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information," RFC 5101 (Proposed Standard), Internet Engineering Task Force, Jan. 2008.
[4] "Mongo DB website," http://www.mongodb.org, Visited: Nov. 2012.
[5] B. Trammell, A. Wagner, and B. Claise, "Flow Aggregation for the IP Flow Information Export (IPFIX) Protocol," Work-in-progress document, http://tools.ietf.org/html/draft-ietf-ipfix-a9n-05, Jul. 2012.
[6] "D3 JS – Data-Driven Documents, Website," http://d3js.org, Visited: Nov. 2012.
[7] "Backbone.js Website," http://backbonejs.org, Visited: Nov. 2012.
[8] L. Verlet, ""Computer Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules," *Physical Review*, vol. 159, pp. 98–103, 1967.
[9] "NfSen homepage," http://nfsen.sourceforge.net, Visited: Nov. 2012.
[10] D. Plonka, "Flowscan: A Network Traffic Flow Reporting and Visualization Tool," in *USENIX LISA'00*, New Orleans,LA, Dec. 2000.
[11] "Ntop Website," http://www.ntop.org, Visited: Nov. 2012.
[12] T. Taylor, D. Paterson, J. Glanfield, C. Gates, S. Brooks, and J. McHugh, "Flovis: Flow Visualization System," in *Conference For Homeland Security 2009. CATCH'09*, Mar. 2009.
[13] K. Lakkaraju, W. Yurcik, and A. J. Lee, "NVisionIP: NetFlow Visualizations of System State for Security Situational Awareness," in *Proceedings of VizSEC/DMSEC '04*, Washington, DC, Oct. 2004.
[14] X. Yin, W. Yurcik, and A. Slagell, "VisFlowConnect-IP: An Animated Link Analysis Tool for Visualizing Netflows," *FLOCON 2005*.

ISBN 978-3-937201-53-5
DOI 10.2313/NET-2016-07-2
ISSN 1868-2634 (print)
ISSN 1868-2642 (electronic)