Towards uncovering BGP hijacking attacks

Quentin Jacquemart

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2015-ENST-0063

École doctorale Informatique,
Télécommunications et Électronique
(Paris)

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Informatique et Réseaux »

présentée et soutenue publiquement par

Quentin Jacquemart

le 06 octobre 2015

Déceler les attaques par détournement BGP

Directeur de thèse : **Prof. Ernst Biersack**
Co-directeur de thèse : **Prof. Guillaume Urvoy-Keller**

Jury

M. Chadi Barakat, Dr.	INRIA Sophia-Antipolis, France	Rapporteur
M. Bruno Quoitin, Prof.	Université de Mons, Belgique	Rapporteur
M. Michael Behringer	Cisco Systems, France	Examinateur
M. Hervé Debar, Prof.	Télécom SudParis, France	Examinateur et Président

T H È S E

TELECOM ParisTech
école de l'Institut Mines-Télécom – membre de ParisTech

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2015-ENST-0063

École doctorale Informatique,
Télécommunications et Électronique
(Paris)

THESIS

submitted to

TELECOM ParisTech

in partial fullfillment of the requirements for the degree of

Doctor of Philosophy

Specialty: Computer Science

defended by

Quentin Jacquemart

on October 06, 2015

Towards Uncovering BGP Hijacking Attacks

Advisor: **Prof. Ernst Biersack**
Co-advisor: **Prof. Guillaume Urvoy-Keller**

Jury members

M. Chadi Barakat, Dr.	INRIA Sophia-Antipolis, France	Reviewer
M. Bruno Quoitin, Prof.	University of Mons, Belgium	Reviewer
M. Michael Behringer	Cisco Systems, France	Examiner
M. Hervé Debar, Prof.	Telecom SudParis, France	Examiner and President

THÈSE

TELECOM ParisTech
member college of Institut Mines-Télécom, and of ParisTech

Abstract

The Internet is composed of tens of thousands independant **Autonomous Systems** (ASes) owned and administred by organizations such as private corporations, Internet Service Providers (ISPs), universities, research labs, etc. These ASes exchange reachability information (i.e. IP prefixes) using the **Border Gateway Protocol** (BGP). BGP was codified with realiability and scalability as the main design goals. Security elements were considered unnecessary and would have added a non-negligible overhead on BGP-enabled systems. Consequently, there is an *implicit* **mutual trust** among ASes because they are unable to verify the validity of the routes they receive from others.

Prefix hijacking abuses this mutual trust in order to introduce *fallacious routes* in the Internet. This phenomenon can be *accidental*, i.e. resulting from a router misconfiguration. On the other hand, the phenomenon can also be the result of a *deliberate* against the global routing infrastructure, or a particular network. Such an attack can be used to blackhole the victim network, impersonate the victim, eavesdrop (including man-in-the-middle attacks), and other malicious activity (e.g. spam).

Many proposals attempt to *secure BGP* by introducing cryptographic verification of the information transmitted to ASes. Unfortunately, they introduce an important overhead on BGP routers, and are therefore unlikely to be deployed in the short-to-mid term future. Consequently, a number of prefix hijacking **detection techniques** have also been proposed, which are deployable today. These detection techniques raise a rather large number of alerts, most of which are **false positives** resulting from benign routing practices. Consequently, the detection techniques are not suitable to study prefix hijacking because an **external observer** typically **lacks** the **ground-truth** needed in order to easily identify false positive alerts, and therefore has to manually inspect and investigate each alert.

The main focus of this Dissertation is to seek the root cause of routing events (i.e. of prefix hijacking alerts) beyond reasonable doubts. In order to do so, we first, we **reduce the global the number of raised alerts** by analysing a large number of false positive alerts. From this analysis, we extract constructs that are composed of patterns and trends that reflect standard, but varied, routing practices. We then analyse the security threat associated with these constructs by considering their impact if they were used in prefix hijacking scenaros. Second, we **circumvent the lack of ground-truth** by analysing suspicious events from as many facets as possible. We achieve this by using a variety of auxiliary datasets that provide additional information about the involved networks, such as registration information, known malicious activity, and application-level activity.

Specifically, we investigate three distinct prefix hijacking scenarios. First, we look at **Multiple Origin AS** (MOAS) prefixes, i.e. prefixes that are simultaneously announced by multiple ASes. We introduce a taxonomy of MOAS prefixes, and we create a series of filters that automatically **discard over 80% of false positive alerts**. We also analyse a real-world case where a MOAS coincided with spam and web scam traffic. We show that the current approach of correlating routing events with malicious activity is *insufficient* to evidence harmful BGP hijacks, and illustrate that the ground-truth substitutes we use enable a better understanding of routing events. Then, we look at **prefix overlaps**, namely at more specific prefixes. We analyse the global BGP routing table and clarify the engineering practices behind the use of such announcements, and present a prototype that currently **discards around 50% of false positive alerts**. Finally, we explore the **IP blackspace**, which is composed of the IP space that is actively globally announced on the Internet, even though it has not been assigned for use. We study the routing-level characteristics of these networks and identify some benign reasons why these networks are announced on the Internet. We focus on the security threat associated with these networks by looking at their application-level footprint, identify live IP addresses, and uncover a large amount of spam and scam activities carried out from the blackspace.

List of Publications

1. Ernst Biersack; Quentin Jacquemart; Fabian Fischer; Johannes Fuchs; Olivier Thonnard; Georgios Theodoridis; Dimitrios Tzovaras; and Pierre-Antoine Vervier.
   "Visual analytics for BGP monitoring and prefix hijacking identification".
   In IEEE Network Magazine, Special Issue on Computer Network Visualization, Volume 26 #6, November/December 2012.

2. Quentin Jacquemart; Guillaume Urvoy-Keller; and Ernst Biersack.
   "A longitudinal study of BGP MOAS prefixes".
   In 6th International Workshop on Traffic Monitoring and Analysis (TMA 2014), April 2014.

3. Pierre-Antoine Vervier; Quentin Jacquemart; Johann Schlamp; Olivier Thonnard; Georg Carle; Guillaume Urvoy-Keller; Ernst Biersack; and Marc Dacier.
   "Malicious BGP hijacks: appearances can be deceiving".
   In IEEE International Conference on Communications (ICC) – Communications and Information Systems Security (CISS) Symposium, June 2014.

4. Johann Schlamp; Ralph Holz; Oliver Gasser; Andreas Korsten; Quentin Jacquemart; Georg Carle; and Ernst Biersack.
   "Investigating the nature of routing anomalies: closing in on subprefix hijacking attacks".
   In 7th International Workshop on Traffic Monitoring and Analysis (TMA 2015), April 2015.

5. Quentin Jacquemart; Pierre-Antoine Vervier; Guillaume Urvoy-Keller; and Ernst Biersack.
   "Demystifying the IP Blackspace".
   In 18th International Symposium on Research in Attacks, Intrusions and Defense (RAID 2015), November 2015.

6. Quentin Jacquemart; Guillaume Urvoy-Keller; and Ernst Biersack.
   "Behind IP Prefix Overlaps in the BGP Routing Table".
   In 17th International Passive and Active Measurements (PAM 2016) Conference, March 2016.

7. Johann Schlamp; Georg Carle; Ralph Holz; Quentin Jacquemart; and Ernst Biersack.
   "HEAP: Reliable Assessment of IP Subprefix Hijacking Attacks".
   In IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Measuring and Troubleshooting the Internet: Algorithms, Tools, and Applications, 2016.

Contents

Introduction 1

In this Chapter, we give a general introduction to the challenges that will be discussed in this Dissertation. First, we introduce BGP (Border Gateway Protocol), which assures routing between Autonomous Systems (ASes) across the entire Internet. We describe the context in which the protocol was designed, thereby illustrating its design goals, and explaining its deficiencies. Then, we then introduce prefix hijacking, the main topic of this Dissertation, an attack on the global routing infrastructure that is possible because of design choices that were made in BGP. We illustrate why this phenomenon is a problem and show the repercussions it can have. We show how existing techniques are lacking, and introduce the steps that we follow in order to tackle this problem. Finally, we give an overview of the improvements made by this work compared to current state-of-the-art systems.

1.1 The Internet and the Border Gateway Protocol

Nowadays, the Internet is composed of over 50,000 computer networks, known as **ASes** (Autonomous Systems). ASes globally interconnect hundreds of millions of diverse end systems, such as data centers and consumer devices, which exchange varied information. As a result, the Internet is now arguably the largest system ever engineered by mankind. However, the first iteration of the Internet, called **ARPAnet** (Advanced Research Projects Agency Network), was only meant to provide communications possibilities between major computational resources and their users in research laboratories. In order to be part of this network, a device had to be connected to an ARPAnet IMP (Interface Message Processor). In 1969, the ARPAnet was only composed of 4 nodes, but grew to a whopping 15 by 1972. [Potaroo; Kurose *et al.* 2010]

During the early-to-mid 1970s, several other networks with proprietary infrastructure and protocols were created, and by 1975 it seemed like a good idea to develop a standard architecture that would enable the interconnection of these different networks. Work under the sponsorship of DARPA (Defense Advanced Research Projects Agency) by Vinton Cerf and Robert Kahn led to an early version of **TCP** (Transmission Control Protocol). Evolutions of that protocol led to **UDP** (User Datagram Protocol), **IP** (Internet Protocol) and modern TCP. These protocols are still the

core of today's networks. On January 1, 1983, the ARPAnet protocols were replaced by the TCP/IP protocol suite. It had approximately 200 hosts connected. [Kurose *et al.* 2010]

In the late 1970s, the National Science Foundation (NSF) created **CSNET** (Computer Science Network) that provided network services to computer sciences departments at US academic institutions that did not participate in ARPAnet. CSNET was also responsible for the first gateway between the United States and Europe. In 1986, the NSF created **NSFNET** to provide access to NSF-sponsored supercomputers. Both of these networks used the TCP/IP protocols from the beginning. Moreover, they resulted in a tremendous growth of the public Internet. [Kurose *et al.* 2010; NSF1; NSF2]

Although the term **Internet** was first mentioned as early as 1974 in [RFC675], as an abbreviation for *inter-networking*, it was not used to refer to the name of a network before the late 1980s, when ARPAnet and NSFNET were interconnected. In 1989, **BGP** (Border Gateway Protocol), which is the main focus point of this Dissertation, was introduced for the first time in [RFC1105], as means to exchange reachability information between ASes. The Internet still relies solely on BGP, albeit on its fourth version, for this task. For this reason, BGP can be referred to as the *glue* of the Internet: it allows each AS to tell its neighbouring ASes which set of IP addresses are hosted in it. Then these neighbours will relay this information to their neighbours and so on. Little by little, these IP addresses will become globally reachable, even though the traffic will have to flow through multiple ASes before reaching its intended destination. [Kurose *et al.* 2010; Tanenbaum 2002; van Beijnum 2002]

This short overview of the history of the Internet illustrates the two following points. First, as its name suggests, the Internet is a network of networks, i.e. a set of independent networks that use the same protocols to exchange information. These networks, called **ASes** (Autonomous Systems), are operated by independent parties, for example academic institutions, private corporations, Internet service providers. Because they are independent, the way they each operate and manage their own network varies. Even if they encounter the same technical challenge, they may choose to solve it differently. As a result, it is unreasonable to expect a perfect global homogeneity among these networks. In particular, each AS has its way own of configuring BGP, which implies that ASes do not necessarily behave in the same way, even when challenged by the same problem (e.g. traffic engineering). Second, the core protocols of the Internet were developed when the networks were relatively small. It can even be argued that very few people believed the Internet would reach the tremendous size it now is. When these networks were small, i.e. during the 20 first years of the global network, the users of the ancestors of the Internet were scientists who used these networks as means to share knowledge and resources. There was an implicit **mutual trust** between these users. In the world of BGP, this implicit trust is extremely important. Because the major development of BGP took place in the late 1980s and early 1990s, virtually no security mechanism were forseen in the protocol because this was considered unnecessary. It would have added a non-negligible overhead on systems that had low computing power. As a result, BGP, is simple, reliable, and mostly unsecured.

Chapter 2 will provide a detailed description of the innerworkings of BGP. For now, we give a summary description of the way it works. In BGP, the *reachability information* represents a range of IP addresses that are expressed as an IP **prefix**. For example the prefix 193.54.0.0/15 represents the 2^{17} addresses included in the range 193.54.0.0-193.55.255.255. Figure 1.1 depicts that AS2200 is the **origin AS** of the prefix 193.54.0.0/15. In other words, the computers using IP addresses included in the prefix are connected inside of AS2200. AS2200 **peers** with AS1, i.e. AS2200 and AS1 are neighbours, and **announces** to AS1 the prefix 193.54.0.0/15. BGP is a **path vector** protocol that stores the list of ASes that propagated reachability information in the **AS path**. Because AS2200 is

the origin, the only AS number that is included in the AS path when the route is originated is 2200. Everytime an AS **propagates** the route to another AS, it *prepends* its AS number to the AS path. In Figure 1.1, AS1 peers with a lot of networks and propagates the announcement to all of its peers. As a result, each announcement for the prefix contains the AS path "1 2200". In the same way, AS2 propagates the announcement to AS3 and updates the AS path accordingly. Consequently, the **AS path** is the list of distinct ASes that have to be crossed before reaching the origin of the prefix, which is located in the rightmost AS of the AS path.

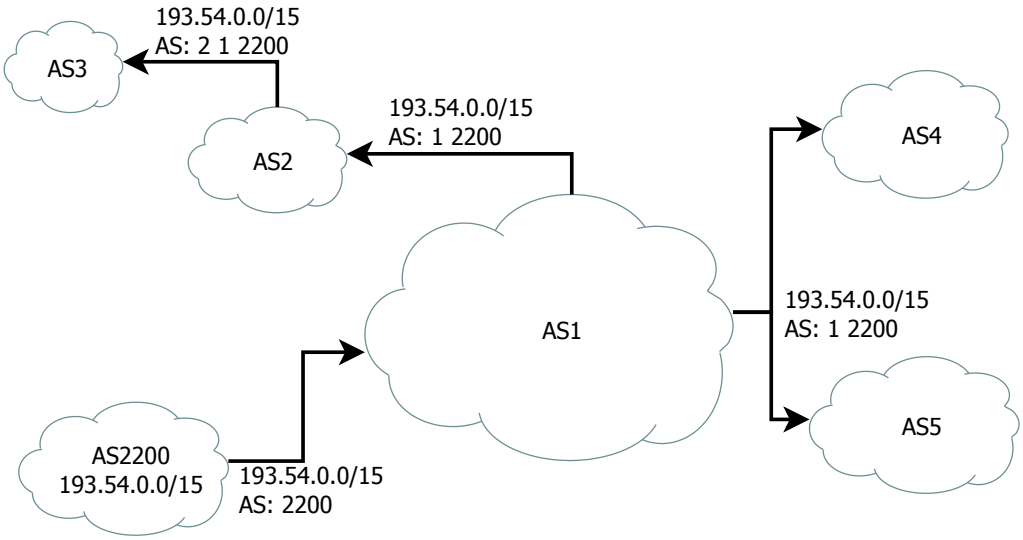


Figure 1.1: Example of BGP announcement and propagation for 193.54.0.0/15, originated by AS2200

Once the announcement is propagated, **packet forwarding** to the origin can be achieved. In BGP, packets are forwarded to the **most specific route** that matches the destination address. For example, imagine that, in Figure 1.1, an AS announces 193.54.0.0/24, which represents the IP range 193.54.0.0-193.54.0.255. This prefix is more specific than 193.54.0.0/15 because it is included in the /15. If AS4 originates the /24 prefix (Figure 1.1), traffic destined to addresses within the /24 will automatically be forwarded to AS4 and not to AS2200. Any other destination address that is outside of the /24's range and included in the /16's range will be forwarded to AS2200.

1.2 Prefix Hijacking

In the previous Section, we introduced BGP and illustrated that, due to the era in which it was designed, the main design concern was reliability. For this reason, the protocol is simple, and, as we can now judge, quite reliable. Unfortunately, it is because BGP is simple that it lacks security mechanisms, and, consequently, that it can be subject to *prefix hijacking* attacks.

Prefix hijacking happens when erroneous routing information is propagated in BGP. It leads to traffic being either routed to the wrong part of the Internet, or via the wrong part of the Internet. Some of the elements that enable prefix hijacking are the following ones [Hu *et al.* 2007]. The hijacking AS **claims ownership** of a prefix and starts announcing it. If the victim is also concurrently announcing the same prefix, a **MOAS** (Multiple-Origin AS) becomes the side effect of the attack because multiple different ASes are announcing the same prefix. In this case, the Internet is effectively divided in two zones: some routers will use the legitimate route, and some will use the hijacked

route, depending on which origin they are closer to. By announcing a **more specific prefix** than the legitimate one, the hijacked route is automatically used by BGP to forward all traffic to the hijacker. Alternatively, by announcing a **less specific prefix** (i.e. a bigger IP range), the attacker can possibly take ownership of IPs addresses that are left unannounced by the AS that they have been assigned to. **Man-in-the-middle** attacks are possible by forging the values inside the AS path, effectively redirecting the traffic to a specific part of the network. Finally, unassigned prefixes can be also abused.

As illustrative examples, we present two confirmed real-world cases of prefix hijackings that make use of these attacks. On February 24, 2008, the Pakistani government decided to block access to YouTube. Pakistan Telecom chose to do this via BGP, and hijacked YouTube's IP addresses by announcing a smaller IP range than YouTube's [RIPE NCC 2008a]. They originally intended to limit the scope of this announcement to only within Pakistan itself, but they somehow misconfigured their BGP routers and this configuration spread worldwide, effectively blackholing YouTube for over an hour. In this case, the large-scale disruption of the Internet was caused by a local misconfiguration. On the other hand, prefix hijacking can be deliberately engineered to take advantage of the trusting nature of the global YouTube infrastructure. One such case occurred with the Link Telekom case, where a spammer abused a defunct Internet service provider's IP addresses in order to send spam emails [Biersack *et al.* 2012; Vervier *et al.* 2013; Schlamp *et al.* 2013]. For the spammer, the advantage of proceeding like this is twofold. First, the identity of the spammer is hidden, thereby making identifications and possible legal procedures more tedious and cumbersome. Second, the spammer is able to change the address space of the source of spam more easily, frequently circumventing the spam blacklists that are traditionally used in order to filter out spam emails from regular emails.

As we just saw, prefix hijacking can be either the result of BGP router misconfiguration or of a deliberate attack on the way the routing infrastructure works. In any case, the erroneous routing information is spread through the whole Internet because there is no built-in security mechanism in BGP that certifies whether an AS is allowed to announce a given set of IP addresses. In other words, BGP is gullible: any BGP router can claim to be any AS and announce any IP prefixes, and its neighbours, unless explicitly configured otherwise, will believe it and repeat the information to their own neighbours. This simple mechanism is one of the features that makes BGP simple and reliable. On the other hand, trusting any incoming information is not a desirable behaviour in a context where the mutual trust between BGP neighbours is minimal, if at all existing. Unfortunately, in order to avoid prefix hijacking, BGP needs to be modified in such a way to replace mutual trust with an ad-hoc authentication mechanism that would enable BGP routers to discard fallacious announcements. Systems such as s-BGP [Kent *et al.* 2000] and soBGP [White 2003], among others, provide means to do add security based on public key cryptography and certificates. However, the computational and configurational overhead necessary with these protocols means that they have not been deployed in the real-world. Moreover, even if a (current or future) proposition becomes widely adopted, unless each and every inter-AS peering is secured through this protocol, prefix hijacking will still be possible, because a chain is only as strong as its weakest link. Considering that deploying a new protocol at the Internet-scale takes years, (e.g. the IPv6 deployment), it is unlikely that prefix hijacking disappear in the foreseeable future.

Because the inter-AS routing infrastructure cannot be altered so as to avoid prefix hijacking, the next best solution is to detect prefix hijacking. By detecting it easily, active counter-measures can be taken so as to minimize the duration of the disruption. Many techniques have been proposed for detecting prefix hijacking. They are either based on the *control plane*, i.e. the AS-level topology view inferred from the BGP routers' information, or on the *data plane*, i.e. based on the way the packets actually travel in the network between two ASes. Detection techniques from the control

plane, such as [Ballani *et al.* 2007; Lad *et al.* 2006; Gao 2001], involve the creation of a model that represents the normal, expected behaviour of a network. Whenever the current view of the network differs from the model, an alert is raised. The complexity and accuracy of the model is then the key element to a good detection scheme. Detection techniques from the data plane, such as [Hu *et al.* 2007; Zheng *et al.* 2007; Hong *et al.* 2009; Zhang *et al.* 2008], involve active probing of the network topology and/or available live hosts in the monitored network. The core idea is that when a hijacking takes place, significant topology changes should be observed and that the victim network is different from the hijacking network. The way these elements are measured, as well as the diversity of factors taken into account, ensures a good detection. More recently, [Shi *et al.* 2012] combined both approaches into a single system in order to correlate information conveyed by both views.

Unfortunately, these tools, even state-of-the-art ones, yield output cluttered with alerts corresponding to benign network events. For the owners of the concerned IP space this is not a problem because they should know the expected behaviour of their networks. Knowing the ground-truth, they can make an informed decision on the value of the alert, and take appropriate action, if needed. Moreover, since they only monitor their own networks, the number of relevant alerts is low enough so as not to be considered as over information. An external observer, however, does not have the ground-truth necessary to possibly mark alerts as a false positive. Consequently, the sheer number of different networks on the Internet, as well as the time and network resources necessary to analyze them, prohibits the systematic use of these techniques in the way they are currently implemented. Indeed, a very peculiar activity can be the result of a legitimate traffic engineering practice by the prefix owner, who is the only one with access with to the ground-truth. We will illustrate just how hard it is to differentiate between a legitimate traffic engineering practice and a prefix hijacking attack by performing a real-world situation analysis in Chapter 3.

1.3 Contributions

In Section 1.2, we illustrated why the current prefix hijacking detection and monitoring techniques are not suitable to study prefix hijacking as a phenomenon, because current state-of-the-art techniques are targeted at network operators, who only need to monitor their own networks. Because these operators know how their networks should behave, and because they only see the information related to their own resources, the solution is acceptable to them. From a third-person point of view, investigating each alert in order to either mark it as the result of a prefix hijacking attempt, or as a legitimate event that is the natural side-effect of a configurational decision or of a topological change is non trivial.

Therefore, in order to systematically and efficiently investigate the root cause behind a routing event, we need to overcome the lack of available ground-truth. When assessing the quality of a new detection technique while detailing it in an article, the adopted approach is often to obtain feedback from network owners directly. This is usually done by kindly asking to network operators to reply to a survey, either by sending them an email directly, or by sending the email on a popular medium, such as the NANOG mailing list. Another possibility is to cash in favours from different contacts who have access to the right kind of data, or who are able to get in touch with the people-in-the-know. In both cases, this is a one-off, non-automated solution that depends on the good will of involved people, the time they have aside their usual workload, and how much information they are willing to share with other parties. In the end, this solution is therefore not sustainable in the long term.

Consequently, in order to be able to classify a suspicious event as either being legitimate or as a prefix hijacking attack, ground-truth should be obtained through other means. The approach

followed in this Dissertation is to analyse a variety of datasets to observe a given event from as many facets as possible. This auxiliary data should reveal who the involved parties are, and what they were doing at that time. If possible, these identities and behaviours should also be compared to what they were doing before so as to see if the behaviour was changed. Consequently, investigating a single routing event to seek its root cause beyond any reasonable doubts can take several days, and even longer. This leaves us with two main problems. First, the relevant auxiliary datasets must be identified, and an access to their content should be secured. Second, the number of cases to investigate should be reduced to a manageable size.

In order to obtain ground-truth, we setup a collaboration, which we detail in Section 1.4. With the help of our partners, we have access to security datasets, in particular related to spam, scam, and malware activities. We also have access to NetFlows data which help identifying the applications that are used by the network involved in the investigated case. By combining all this information, we can see what kind of network activity a specified network was generating at a given point in time. We also access the Internet Routing Registries (IRRs) which contain, among other things, information about IP ranges and AS registrations. We use this information to know who the involved parties are, and how they are relate to one another.

In order to reduce the number of cases to investigate, we dedicate a sizable part of this Dissertation to remove false positive alerts. Starting from a list of suspicious BGP events, similar to what state-of-the-art tools would output, we manually investigate cases, and analyze the involved networks. By looking at a rather large number of distinct false positives, we are able to find global constructs, i.e. patterns and trends that reflect standard, but varied, routing practices. As a result, we can cluster all BGP events into groups of similar events. Then, by considering each group on its own, we estimate the security impact in order to know if this construct represents a global threat or not. If it does not, we dismiss the group as being a false positive.

In this Dissertation, we address three of the possible ways to for carry out prefix hijacking attacks. Namely, we investigate

1. concurrent announcements where both the attacker and the legitimate owner announce the same prefix from two distinct ASes (MOAS prefixes);
2. sub-prefix announcements where the attacker announces a prefix more specific than the legitimate owner (overlapping and sub-MOAS prefixes);
3. the IP blackspace where an attacker hijacks IP addresses that have not been allocated to anybody.

For MOAS prefixes, we provide an in-depth analysis of benign routing practices that lead to MOASes. We present a taxonomy of MOASes, detail which standard routing practices lead to them, show their prevalence, and how they evolve through time. Then we define a set of filters that we use to remove the false positive alerts. These filters enable us to remove over 70% of all alerts generated by state-of-the-art tools. We then correlate the remaining alerts with security-related datasets, and provide a detailed case study of a suspicious routing event. We show how we use the datasets we have available, what they bring to the analysis. We also illustrate how strong correlation should be before concluding that an event is the result of a malicious prefix hijacking attack. This kind of strong evidence has unfortunately not been used before, which implies that the conclusions of previous reported hijacks may be biased towards one verdict or another.

For overlapping prefixes, we present a classification of the prefixes inside the BGP routing table into families of prefixes that belong to the same organization. We show how much overlap exists

inside families, and show that family behaviours are clearly the result of configurational choices made by network operators. We illustrate how much information we really infer about the corporations behind the network resources by showing that distinct corporations in a similar market tend to adopt similar networking choices. We then present a prototype that aims to validate the announcements of overlapping prefixes. This prototype is currently able to discard more than 50% of false positives, even though it is not yet fully functional integrated.

For the IP blackspace, we first present a methodology that uncovers the IP space that has not been allocated for use. We then look at blackspace networks that are globally announced in BGP. By looking at the applications hosted in this IP space, we are able to uncover a few legitimate use cases of such IP addresses. Then we uncover and provide a detailed analysis of spamming and scamming campaigns that were carried out from the blackspace. To the best of our knowledge, we are the first ones to report with strong evidence such a malicious activity from within the blackspace.

1.4 Collaboration

In Section 1.2, we introduced prefix hijacking, the current detection techniques, and explained that they were targeted at reporting anomalous routing behaviour to prefix owners, such as Internet service providers. This, combined with the lack of available ground-truth makes them unsuitable for studying prefix hijacking as a phenomenon on the global Internet. As a result, in Section 1.3, we stated that we will use a diversified set of auxiliary datasets in order to obtain the elusive ground-truth and find the root cause of suspicious BGP routing events. Access to diversified data is possible thanks to a collaborative effort that was setup with our two partners: Symantec Research Labs and Technische Universität München. Consequently, some of the work presented in this Dissertation is the result of collaborating with these partners. Whenever it is the case, external contributions will be explicitly credited as such. Meanwhile, this parallel work also led to two other theses that discuss other aspects of BGP hijacking attacks. We introduce them here.

The partnership with Symantec Research Labs originally started as an attempt to focus on malicious prefix hijackings. In particular Vervier's work [Vervier 2014] focuses on a phenomenon called fly-by spamming, first reported by [Ramachandran *et al.* 2006]. These spammers appear to hijack a range of IP addresses in order to send spam, and then immediately release the IP range. Vervier's main goal is to assess the severity this phenomenon and to characterise the modus operandi of these spammers. To this end, Vervier proposes SpamTracer, which collects BGP control plane data, as well as live traceroute measurements about networks that have been involved in spamming activity. Using this tool over a period of two years, Vervier was able to uncover over 2,000 malicious prefix hijacks, some of which have been confirmed by the involved parties; consequently showing that a small but agile number of spammers are successfully able to stealthily, routinely, and persistently hijack a large number of IP addresses to send spam and host scam websites. These spammers specifically target the dormant IP space, i.e. IP space that was allocated but not globally announced before the uncovered events.

The partnership with the Technische Universität München started as an attempt to enrich the BGP routing data with application level data, such as gathered by NetFlows collected on Munich's scientific network (Münchner Wissenschaftsnetz). In [Schlamp 2015], two main points are addressed. First, Schlamp proposes a way to mark some more specific prefixes announcements as legitimate, an attack that state-of-the-art tools do not handle well. The approach uses ground-truth extracted from the IRRs (Internet Routing Registries), a dataset of benign topology relations (built from the

methods developped in Chapter 4), and public cryptographic key beacons. The current prototype of this system is overviewed in Chapter 4 and is already able to validate around half of the global events, even though it does not currently use the full set of data from the IRRs. Second, Schlamp focuses on the expiration of the domain name associated to network resources such as IP ranges and AS numbers. Upon expiry of a domain, an attacker can re-register the domain, thereby gaining access to the network resources by faking the legitimate owner's identity. This is the most likely explanation for the Link Telecom hijack presented in Section 1.2. With his experiments, Schlamp found over 70 networks and 7 ASes vulnerable only in the European zone alone.

1.5 Roadmap

In this Chapter, we introduced BGP, as well as the context and purpose for which it was engineered. We explained that, due to design choices that were made, the protocol is vulnerable to a number of attacks. One of such attack is prefix hijacking, which is possible because there is no mechanism to verify the validity of the information propagated through BGP. We explained why it is not currently possible to avoid prefix hijacking, and that, consequently, the next best thing is to be able to detect it, in order to properly deal with the situation. We then introduced the general methodology that we will follow in the remainder of this Dissertation, which is organised as follows.

Chapter 2 formally introduces BGP and its (inner) details. We look at important hijacks that occurred in the past, thereby illustrating the effect prefix hijacking can have on the global network. We discuss several propositions that have been made in order to secure BGP, and show why they cannot be implemented as of today. Then, we provide a review of the relevant related work to prefix hijacking, and detail the most important detection techniques that have been proposed so far. Finally, we discuss the limitations associated with the datasets that we are using to carry out our analyses.

Chapter 3 focuses on MOASes (Multiple-Origin AS) and provides a detailed analysis of the standard network practices that lead to this phenomenon. Then we define a set of filters that we use to remove false positives from state-of-the-art techniques, and we provide a detailed analysis of a real-world suspicious MOAS case. With this case, we illustrate the power behind our analysis, and underline the shortcomings of previous works that have been carried out in this area, namely in terms of hijack verification.

Chapter 4 focuses on overlapping prefixes. We start by classifying the prefixes inside the BGP routing table as families of prefixes that have been assigned to distinct entities. We then provide a detailed analysis on how these families behave globally, and show that distinct corporations in the same business segment appear to adopt a similar network policy. Finally, we focus on the sub-MOAS phenomenon and present a prototype that we use to validate the announcements of more specific prefixes.

Chapter 5 focuses on the abuse of the IP blackspace, which is composed of the IP addresses that have not been assigned for use to anybody yet. We first present a methodology to compute this space, and then we study the characteristics of the networks in this address space that are globally reachable. Then, we focus on the security threat associated with the announced blackspace networks and uncover a large amount of spam and scam activities, including a confirmed case of fraudulent IP address abuse.

Finally, Chapter 6 summarizes the contributions of this Dissertation, and how this work improves our understanding of prefix hijacking as a phenomenon. We also present future work, as well as the challenges associated with it.

BGP and Prefix Hijacking

2

In this Chapter, we formalize the notions introduced in Chapter 1. First, in Section 2.1, we explain the inner workings of BGP, which we use to illustrate the technical reasons behind *mutual trust*. Second, in Section 2.2, we introduce a *taxonomy of prefix hijacking* and detail the various ways through which such an attack can be carried out. We also provide a number of past incidents that had large repercussions to show the devastating effects of either *accidental* or *intentional* prefix hijacking. Third, in Section 2.3, we take a look at the various proposals that were made in order to render prefix hijacking impossible. Fourth, in Section 2.4, we introduce a number of techniques that have been proposed over the last 15 years in order to detect prefix hijacking. We provide a detailed overview of their methodology and limitations. Fifth, we discuss the various ways in which BGP data can be accessed in Section 2.5, and discuss about the limitations inherent to the collection infrastructure. Finally, Section 2.6 introduces the Internet Routing Registries and the Routing Policy Specification Language.

2.1 The Border Gateway Protocol

In Chapter 1, we saw that the Internet is composed of a set of independent networks, operated by organizations such as private corporations, Internet Service Providers (ISPs), research labs, universities, etc. Each of these networks forms an **Autonomous System** (AS). Originally, an AS was defined as a set of routers that are under a single administration. They use a common interior gateway protocol, such as OSPF, and have a common way of selecting the routes to destinations outside of their own network [RFC4271]. Nowadays, this definition is a bit simplistic: different parts of a single AS may behave differently because they may be administered by different people [Bush *et al.* 2009]. However, the routers are still administered by the same company, and, from the outside, an AS *appears* to have a coherent behaviour [RFC4271].

In this Section, we formally introduce the internals of BGP, the **Border Gateway Protocol**. BGP is quite complex, and a lot of issues are not well understood [Kurose *et al.* 2010]. It was first codified in June 1989 in [RFC1105]. One year later, BGP-2 was introduced in [RFC1163]. Version

3 was introduced in October 1991 in [RFC1267]. The current version of BGP, BGP-4, was first introduced in [RFC1654] in 1994, then revised in [RFC1771] a couple of months later. The details we present here are from the latest revision of BGP-4 which is described in [RFC4271].

2.1.1 Introduction

The primary goal of BGP is to **exchange network reachability information** between two BGP-enabled routers. If those two routers are located in the *same AS*, the routers are *internal peers* and have established an *internal BGP session*. This kind of session is used to disseminate information learnt from different neighbours inside the network. Internal BGP routers must be connected as a fully-meshed network, which leads to difficulties in large-scale networks. In the rest of this Dissertation, we will focus on **external BGP sessions**, i.e. the situation in which the two BGP peers are located in a **distinct AS**. As a result, when [RFC4271] defines diverging behaviour depending on the session type, we will only present the behaviour related to external BGP sessions.

BGP is the *de-facto* inter-AS routing protocol. As a result, it is often described as the *glue* of the Internet, because it allows the independent AS to exchange reachability information. BGP is a **policy-based routing protocol** because it allows each AS to enforce a strict routing policy. This routing policy most often reflects economic considerations, such as bandwidth price, and depends on the business relationship (and thus the peering agreement) between two neighbouring ASes [Gao 2001]. BGP is also a **path-vector** protocol. In other words, it guarantees loop-free routing by keeping a list of all the AS hops that the announcement went through. The *attribute* in which this information is stored is called the **AS path**. These notions are formalized in Section 2.1.5 and Section 2.1.6.

Even though BGP's primary use is to exchange reachable IP addresses among ASes, BGP relies on TCP (Transmission Control Protocol) to establish a session between two peering routers. The official port assigned for BGP is port #179 [PNR]. However, this use of TCP implies that a BGP-enabled router must be able to reach its peers without BGP-learnt information, which is possible through router pre-configuration, for example via a static route.

2.1.2 IP Prefixes

Since version 4, BGP propagates reachability information using CIDR (Classless Inter-Domain Routing) IP prefixes [RFC4632].

In CIDR notation, the range of IP addresses available to a network are expressed as an **IP prefix**. An IP prefix is composed of two parts separated by a slash. For example, the IPv4 prefix

192.168.0.0/16

indicates that the *network address* is 192.168.0.0 and that the 16 most significant bits of the *network mask* are true. In other words, this IP prefix spans 2^{16} IP addresses: from 192.168.0.0 to 192.168.255.255. Using the standard notation, it is equivalent to 192.168.0.0/255.255.0.0, with 255.255.0.0 as the network mask.

Because prefixes are sets of IPs, we can define relations between them. It is possible for an IP prefix to be totally included in another one. For example, 192.168.128.0/24 is completely included

in 192.168.0.0/16. The former is said to be **more specific**, and the latter **less specific**. In order to be more specific, a prefix must have the same network IP address and a longer mask. Moreover, if A is more specific than B, B is less specific than A.

IP addresses assignation is an administrative process that originally starts with the IANA (Internet Assigned Numbers Authority), a department of ICANN (Internet Corporation for Assigned Names and Numbers). The IANA originally holds the whole pool of IP addresses and distributes them to RIRs (Regional Internet Registries) according to their need. There are 5 RIRs worldwide, each responsible for a different geographical location:

- AFRINIC (African Network Information Center), responsible for the continent of Africa, headquartered in Ebene, Mauritius;
- APNIC (Asia Pacific Network Information Center), responsible for the Asia/Pacific region, headquartered in Brisbane, Australia;
- ARIN (American Registry for Internet Numbers), responsible for the USA, Canada, and portions of the Caribbean, headquartered in Chantilly, VA, USA;
- LACNIC (Latin America and Caribbean Network Information Center), responsible for South and Central America, and the remainder of the Caribbean, headquartered in Montevideo, Uruguay;
- RIPE NCC (Réseaux IP Européens Network Coordination Center), responsible for Europe and the Middle East, headquartered in Amsterdam, The Netherlands.

RIRs are responsible for assigning IP addresses to end users. The IPv4 pool has originally been attributed as chunks of /8 prefixes to RIRs by the IANA. The IPv6 pool has been originally assigned as /23 chunks, but more recently larger chunks, sometimes as large as /12s have been allocated to RIRs.

2.1.3 AS Numbers

Much like networks are assigned a globally unique IP range to use, each AS is assigned an **AS Number** (ASN) which uniquely identifies an AS on the network.

Originally, AS numbers were 2 bytes, with the following uses [RFC6996]:

- from 1 to 64,511: public AS numbers, assigned by RIRs to BGP-needing entities.
- from 64,512 to 65,535: private AS numbers. These ASNs can be used to exchange routes between two routers, but the private ASN should be replaced by a public ASN before being propagated globally. This is useful, for example, when a local ISP needs to provide connectivity to a customer. Instead of registering an ASN, they can use a private ASN for their local peering, and the provider propagates the route using its own, globally unique ASN. This situation can eventually lead to MOAS (Multiple-Origin AS) prefixes, a situation in which a prefix appears to be originated by multiple ASes. MOASes are studied in depth in Chapter 3.

Because 2 bytes ASNs were all used up, [RFC6793] proposed to switch to 4 bytes ASN. In that transition, [RFC5396] proposed to adopt a new notation for ASN representation. The first, called AS-PLAIN, is the one that had been in use since the beginning: an AS number is represented by its number, in base 10, preceded by the letters "AS". For example:

AS12345

The second, AS-DOT+ aims to represent the 32 bits in the following manner:

```
high-order 16-bit value in base 10 . low-order 16-bit value in base 10
```

In other words, in order to get the equivalent integer value of AS$N.X$ for AS-PLAIN, the formula is

$$\text{AS-PLAIN}_{\text{value}} = N \cdot 65536 + X$$

The last method, AS-DOT, simply mixes the use of AS-DOT+ and AS-PLAIN. Old 2 bytes ASNs are written with AS-PLAIN, while new 4 bytes ASNs are written with AS-DOT+.

Much like IP prefixes, AS numbers are assigned by RIRs to end users. As of January 2009, RIRs primarily issue extended 4 bytes AS numbers [RIPE NCC 2008b; ARIN 2008; APNIC 2008]. In order to preserve compatibility with older hardware, AS23456 is reserved and referred to as `AS_TRANS`. Thus, two AS paths fields exist: one, regular with 2 bytes only ASNs, and one, optional, containing 4 bytes ASNs [Toonk 2008].

2.1.4 Routing Information Base and Routing Table

In BGP, **routes** are pairs composed of a set of IP prefixes and a number of attributes associated with the destination(s). These routes are stored in the **RIB** (Routing Information Base), which is composed of three distinct elements that contain routing information:

- The **Adj-RIB-In**, i.e. the incoming adjacent RIB. There is one Adj-RIB-In per BGP peer. It contains the unprocessed routes received from that peer, i.e. using that peer as the next-hop router.
- The **Adj-RIB-Out**, i.e. the outgoing adjacent RIB. There is one Adj-RIB-Out per BGP peer. It contains the routes that the BGP router is willing to announce to its peer, according to its internal policy.
- The **Loc-RIB**, i.e. the local RIB. There is one unique Loc-RIB in the router. This table contains the routes selected by applying the route selection process to all of the Adj-RIB-In.

The **routing table**, or, equivalently, the *forwarding table*, contains the information necessary for the BGP router to forward packets. It is not part of the RIB, but is built upon the data contained in the RIB. More precisely, it is a combination of static routes configured on the router, routes learnt via an interior gateway protocol (i.e. from within the AS), and routes learnt from BGP. The way these elements are combined into the routing table is also part of the internal policy of the BGP router.

2.1.5 BGP Messages

Four types of messages are defined by [RFC4271] to carry out its operations. We detail them here.

The **open message** is used in attempt to establish a BGP session between two peering BGP routers. It contains a protocol version number which can be used for version negotiation, the ASN of the router, the **BGP identifier**, and various parameters such as the time-out delay. The BGP identifier is an IP address that is used by the router. However, it must the same for regardless of the interface used by the router to establish a BGP session, and should be globally unique so as to uniquely identify the BGP router.

The **keep-alive message** is used to supplant the TCP-based keep-alive mechanism. BGP keep-alive messages are exchanged to make sure the BGP session does not time out. They are also used as the confirmation that a BGP session has been successfully established.

The **notification message** is used to convey an error. The BGP session is closed immediately after it has been sent.

The **update message** is exchanged within a BGP session in order to *announce* and *withdraw* routes. The message is divided in two parts:

1. withdraws: a list of one or more prefixes whose routes are now infeasible (i.e. that cannot be reached via this particular next-hop),
2. updates: a list of path attributes and a list of prefixes to which they apply.

The **path attributes** are classified in 4 distinct categories:

- well-known mandatory,
- well-known discretionary,
- optional transitive,
- optional non-transitive.

Well-known attributes are attributes that must be recognized by all BGP implementations. If the well-known attribute is *mandatory*, it must be included in every update message, if it is *discretionary*, this is not necessary. A *transitive* attribute specifies that an attribute should be propagated as part of the route, if the router decides to propagate the route. The opposite, *non-transitive* attributes should not be propagated further than the local router, i.e. these attributes are shared between a router and its direct neighbour. All well-known attributes are transitive. Finally, an *optional* can be unknown to the implementation, in which case, if it is transitive, it should be accepted and passed on to neighbours by setting the *partial* flag; if it is non-transitive, it should be quietly ignored. The path attributes characterize the route and the main ones are introduced below:

- the **origin** is a well-known mandatory attribute that is generated by the origin of the route information. It is either set to IGP if the route information was learnt from an interior gateway protocol such as OSPF; or it is set to "incomplete" if the information was learnt through some other means. The last possible value is EGP, and indicates that the route was learnt with EGP as defined in [RFC904], the protocol that carried out a similar task in the NSFNET days as BGP does today in the Internet. EGP is now a defunct protocol, and this value should not be currently in use.

- the **AS path** is a well-known mandatory attribute that is composed of a list of AS path *segments*. An **AS path segment** is either a *set* or a *sequence* of AS numbers that indicate the ASes that have propagated this route. If a segment is a sequence, the *order of the ASes* indicates the order of the ASes that have gradually propagated this announcement. In a set, the order is random.

  The primary function of the AS path is to be used as a loop avoidance mechanism. A router must not select any incoming route that contains its AS own number anywhere in the AS path, effectively because it means that its neighbours use the router's own AS as a way to forward the traffic to its destination.

  Consequently, if a BGP router chooses to propagate a route (i.e. if it decides to put it in one of its Adj-RIB-Out), it must modify the route's AS path so as to introduce its own AS number in it. If the first segment in the AS path is a sequence, it **prepends** its own AS number to the AS path. If the first segment in the AS path is a set, it should prepend a new sequence segment that contains its own AS number. If the AS path is empty, it should create a sequence segment and add its own AS number in it. In this last case, the router is the **origin AS** (alternatively *origin*) of the associated prefix.

- the **next hop** is a well-known mandatory attribute that contains the IP address of the router that should be used to forward the traffic to IP addresses contained in the associated prefix. In general, when propagating a route, a router will set this attribute to the IP address of the interface it uses to peer with its BGP neighbour.

- the **multi-exit discriminator** is an optional non-transitive attribute that can be used in order to indicate a preference over the link that should be used in case multiple connections are available between two ASes. The route with the lower value will always be preferred.

- the **aggregator** is an optional transitive attribute which may be added by a router that chooses to perform a route aggregation. Route aggregation can be performed by a router in order to reduce the number of existing routes, effectively replacing multiple similar routes by a single one. This is where an AS path containing an AS set segment may be generated in order to keep the routing loop free.

- the **atomic aggregate** is a well-known discretionary attribute that should indicate if the route has been *aggregated*. However, this is not a requirement. The atomic aggregate is really a flag that indicates if the AS path contains a segment of type "set". Because of the way the AS path is built, if the atomic aggregate attribute is not set, it means that the AS path is composed of a single sequence. In that case, the *origin AS* of the route is the rightmost ASN in the AS path, i.e. the last one in the list.

As we can see, the update message actually carries the routing information, and is the message exchanged between BGP routers in order to announce and withdraw routes. Upon receipt by a BGP router, an update message triggers a route selection process that we detail in the next Section.

2.1.6 Decision Process and Route Selection

This Section details the processes that update the RIB and the routing table when an update message is received.

First, all withdrawn routes are processed by removing the route from the peer's Adj-RIB-In. Then, the updated routes are processed. The updated part of the update message either contain a new route, or an existing route with a new set of attributes. If the route is new, it is inserted in the peer's Adj-RIB-In. If the route already exists in the peer's Adj-RIB-In, it is replaced by the incoming route. This effectively enables BGP routers to change route attributes without having to explicitly withdraw them first.

Once the Adj-RIB-In has been modified, the **decision process** is started. Its goal is to **select the best route** available for each destination, while, at the same time, enforcing the router's local policy. The decision process happens in three phases.

Phase 1 **computes the degree of preference** for each route in the Adj-RIB-In and its feasibility. [RFC4271] does not define the exact process that should be used to compute the preference of a route, but defines the characteristics of an abstract preference function that is able to enforce a local policy. Router vendors, on the other hand, implemented it as a set of procedures which take into account multiple variables [van Beijnum 2002]. The local policy can be defined by changing the values of these variables and the way they relate to one another. Since this implementation is vendor-specific, it is hard to describe, but it generally comes down to preferring **routes with a shorter AS path** and a **lower multi-exit discriminator**. In order to illustrate a vendor-specific variable, Cisco IOS attributes a weight to a route, preferring routes with a heavier weight.

Phase 2 **selects the preferred route** across all existing Adj-RIB-Ins. In other words, it compares all the routes learnt from all peers in order to select the one that will be inserted in the Loc-RIB. For each destination, it selects a *feasible* route with the highest degree of preference. A route is not feasible if the address of the next hop router becomes unreachable if the route is installed, or if the interface used to forward traffic to the route is down (or that traffic forwarding is disabled on that interface). Similarly, a route with the router's own AS number included in the AS path is not feasible because it would effectively lead to a routing loop. (If the router's own AS is in the AS path, it means that the router's own AS has already propagated this announcement, and therefore, is on the path between the origin AS and the rest of the network). If, across all Adj-RIB-Ins, there is only one route to that destination, it is, by default, preferred. If there are multiple routes to the same destination with the same degree of preference, tie-breaking rules are applied. These tie-breaking rules prefer routes with a lower origin value, rules with a lower multi-exist discriminator value, routes with the lowest BGP identifier value, and routes received by the lowest IP address for the peer. Once the favourite route to a destination has been found, it should be inserted in the Loc-RIB, possibly replacing another route already existing in the Loc-RIB. Once the Loc-RIB has been updated, the routing table is updated by removing unfeasible routes, and by inserting or updating existing routes. As mentioned earlier, the routing table may also contain information external to BGP, such as static routes, or routes learnt from an internal routing protocol such as OSPF. The way all these diverse routes are combined into a single routing table is also a matter of local policy.

Phase 3 **disseminates the routes** installed in the Loc-RIB to each peer, while enforcing the local policy. This policy may selectively prevent a route from being propagated. The policy also enforces if the router should aggregate routes where possible.

2.1.7 Traffic Forwarding

Packet forwarding in a BGP router is done solely based on the information stored inside the routing table. The chosen route for forwarding is done with the **longest prefix match rule**. This rule

specifies that the most specific route (i.e. the longest prefix) that matches with the destination IP address of the IP packet to be forwarded should be used. As an example, if the two prefixes 10.0.0.0/8 and 10.0.10.0/24 are included in the routing table, a packet destined to IP address 10.0.10.26 will be forwarded on the /24 route.

2.1.8 Mutual Trust

In this Section, we formalize the notion of **mutual trust** that we introduced in Chapter 1.

In Section 2.1.5, we saw that BGP routers exchange reachability information, i.e. prefixes with update messages. In Section 2.1.6, we detailed the way BGP routers select a route to each destination that has been announced. Specifically, none of the three phases of the route selection process provide any check on any routes: incoming routes are added to the Adj-RIB-In, and a preference is computed on the route. This preference enforces the routing policy of the router's own AS, which is mostly based on economic considerations [Gao 2001]. Then all the Adj-RIB-Ins are compared in order to select the overall favourite route for a destination. Apart from making sure that the route is feasible, which basically amounts to making sure that the next hop can be reached, this mechanism does not perform any check on the route itself. In other words, unless configured explicitly not to accept any incoming routes, BGP routers consider for selection any incoming route. This implies that any router in any AS can claim to be any (other) AS, and announce any prefix, including prefixes they do not own. Moreover, considering the route attributes, there is no way in which a BGP router can verify the route before adopting it. More specifically, a router cannot verify that an AS is authorized to announce a given prefix, and a router cannot verify the identity of a distant AS, i.e. that it really is the AS it claims to be.

In Section 2.2, we formally introduce prefix hijacking and the mechanisms that abuse mutual trust in order to propagate fallacious routing information throughout the network.

2.2 Prefix Hijacking

Prefix hijacking (also known as **BGP hijacking** or **IP hijacking**) is the act of absorbing (a part of) the traffic destined to another AS through the propagation of erroneous BGP routes, which is possible because of the implied mutual trust among BGP routers. It can be the result of routers misconfigurations [Mahajan *et al.* 2002] or malicious intents [Ballani *et al.* 2007; Hu *et al.* 2007; Qiu *et al.* 2007; Tahara *et al.* 2008; Butler *et al.* 2010].

Regardless of the intentions of the issuer of the incorrect routes, we will refer to them as the **hijacking AS**. In the same fashion, the route propagated by the hijacking AS is the *hijacked route*. The network whose route has been hijacked will be referred to as the **victim AS**. The correct route to the victim AS is referred to as the *legitimate route* (or the **original route**).

By hijacking a prefix, an attacker may [Zheng *et al.* 2007]

- create a black hole in the network (by simply dropping the packets);
- steal the victim AS's identity by imitating the victim's services (e.g. duplicate website);
- intercept the traffic to eavesdrop (or record) the exchanged data and then forward the data back to the victim AS;

- send spam, and/or carry out other malicious activity.

In this Section, we first present of taxonomy of prefix hijacking which defines the possible vectors to carry out such an attack. We then provide an overview of some large-scale hijacks.

2.2.1 Taxonomy of Prefix Hijacking

A prefix hijacking attack can be carried out in different ways. In this Section, we present the vectors of attack through which prefix hijacking can occur. A similar taxonomy was presented by [Lad *et al.* 2006] and [Hu *et al.* 2007], but we have updated ours with respect to more recent works.

2.2.1.1 Concurrent Ownership

A concurrent **ownership** hijacking attack happens when the hijacking AS claims to be the owner of the prefix. In other words, the attacker announces the prefix with its own AS number as the origin of the prefix. This fallacious announcement happens concurrently to the legitimate announcement. Consequently, the prefix appears to be originated by two distinct ASes, a situation known as a **MOAS** (Multiple Origin AS). Since the hijacker is advertising itself as the origin AS, the AS path can appear shorter to neighbouring ASes than the one of a legitimate route. The hijacked route is then selected – if only by peers of the hijacking network – as the way to the victim network. Chapter 3 is dedicated to the study of MOAS prefixes, both in legitimate use, and in a hijacking scenario.

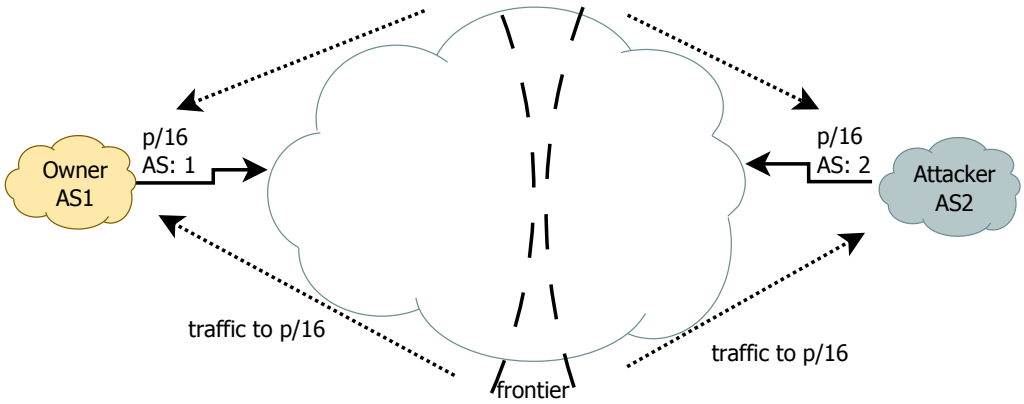


Figure 2.1: A concurrent ownership attack

Figure 2.1 illustrates a situation in which the prefix p is announced by both, the legitimate owner (AS1), and a hijacker (AS2). An approximate frontier is also depicted to illustrate that the networks that are topologically closer (in terms of the length of the AS path) to AS1 will more likely favour AS1's announcement of p. Conversely, the networks that are closer to AS2 will favour AS2's announcement of p. As a result, the Internet is divided in two zones that route p to two distinct ASes.

2.2.1.2 Subnets

In a **subnet attack**, the hijacking AS announces routes to a more specific prefix address than the victim prefix. Because of the longest prefix match rule, any router that receives the incoming route without filtering it will automatically forward any traffic destined to that subprefix to the hijacking AS. The victim only has two ways of dealing with this attack:

1. inform the NOC (Network Operation Center) of the hijacking AS that they are misbehaving. Since it is unlikely they will cooperate if the attack is not the result of a misconfiguration, the victim will have to resort to getting the cooperation of an upstream provider of the hijacking AS. This may not be simple.

2. announce an even more specific prefix than the hijacking subnets. However, in order to keep the size of the routing table as small as possible, most ASes reject too specific incoming routes. So, when both, the attacker and the victim, announce a /24 address, it is likely that there is not much more that can be done on the victim's side, apart from getting cooperation from other networks (and maybe moving online services servers to a back-up location with a different network address until the attack stops).

The hijack of YouTube, detailed in Section 2.2.2 is an occurrence of a subnet attack.

If the origin AS of the more specific route is different from the one of the original route, the route is referred to as a **sub-MOAS**, meaning that the more specific route is concurrently announced by another AS than the less specific one. Chapter 4 focuses on the study of the legitimate use of more specific prefixes, and then focuses on the detection of malicious sub-MOAS occurrences.

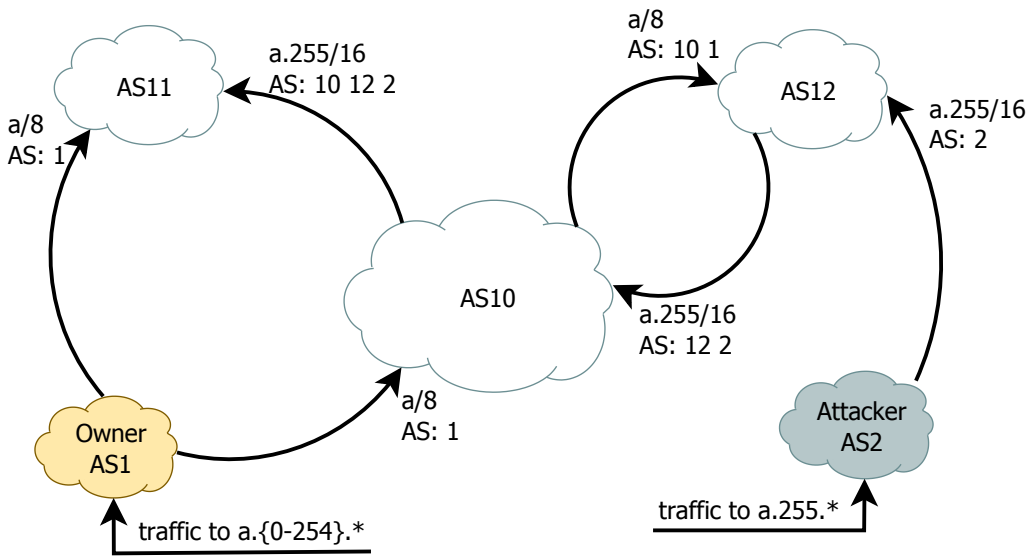


Figure 2.2: A subnet attack

Figure 2.2 illustrates a subnet attack. AS1, the owner of the prefix $a/8$ announces the prefix to the Internet. Neighbour AS10 accepts this announcement and forwards it to AS12. At this point, traffic destined to $a/8$ is forwarded to AS1. When AS2 hijacks a subnet of $a/8$, namely $a.255/16$, it announces it to AS12. AS12 accepts this announcement and propagates it to AS10, which, in turn,

forwards it to AS11. Because BGP uses the longest prefix to forward traffic, any packet destined to $a.255/16$ will be forwarded to AS2. However, all traffic to the rest of $a/8$ will still be forwarded to AS1. If the hijacker choses well the subnet to hijack, they can target a specific service from the owner. For example, in the case of the YouTube hijack, the more specific prefix announced by Pakistan Telecom was the subprefix containing YouTube's DNS and web servers.

2.2.1.3 Forged AS Links

The attacking AS can also decide to tamper with the information embedded in the AS path. Conceptually, the list of ASNs in the AS path represent the path of ASes between the router and the origin of the prefix. By tampering with the list, it effectively creates a **forged AS link** between two ASes, i.e. it appears that two ASes are neighbours when they actually are not.

With a forged AS link attack, the AS path will be longer than it would have been with an ownership attack; but the attack is harder to detect because there is no occurrence of a MOAS. For maximum effect, the attacking AS should claim to be second AS hop, since being any further down in the path would significantly decrease the amount of hijacked traffic [Ballani *et al.* 2007]. Interestingly, [McArthur *et al.* 2009] studied the impact of the hijacked route depending on the position of the hijacking AS in the AS path by measuring the amount of traffic that can be stolen. Unsurprisingly, the further from the origin the the hijacking AS appears to be, the smaller the percentage of packets that will be router to the hijacking AS. Consequently, a hijacker could decide to engineer a stealthier attack by opting to suck in a small percentage of the victim's traffic instead of the whole volume. From the owner's perspective, this would be very hard to detect because there would not be any significant loss of traffic volume. Moreover, it is likely that the owner does not know if two distant ASes are connected together or not.

It is also possible for the hijacker to steal not only the prefix, but also the AS number of the legitimate owner of the prefix, thereby announcing themselves as the legitimate AS. In other words, the hijacker announces the legitimate owner's ASN with the legitimate prefix to the its upstream. Consequently, the fake link is located between the origin and the upstream, i.e. the legitimate AS does not really peer with the hijacker's AS. [Schlamp *et al.* 2013] refer to this situation as an **AS hijacking**.

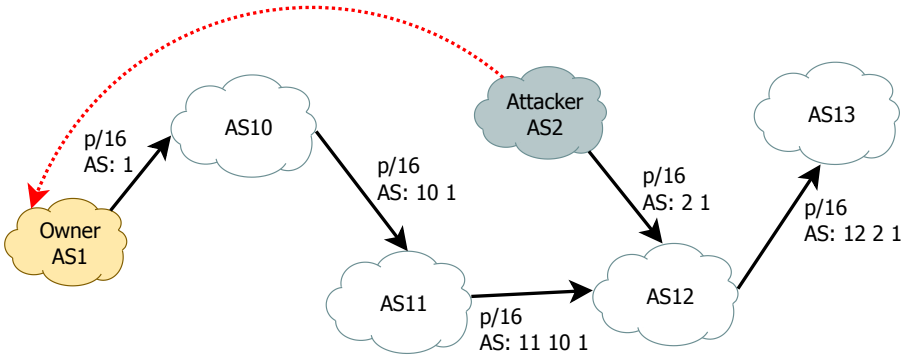


Figure 2.3: A forged AS link attack

Figure 2.3 illustrates a forged AS link attack on AS1's prefix $p/16$. AS1 announces its prefix to its peer AS10, who propagates it further in the network. Somewhere along the path, a hijacker (AS2) announces the same prefix, but inserts AS1 in the AS path. When announcing the prefix,

AS2 effectively advertises itself as a neighbour of AS1. However the topology of the network of Figure 2.3 shows that AS2 and AS1 are not neighbours. The fake link is highlighted by the red, dashed line. However, because AS12 is directly peering with AS2, it will more likely prefer AS2's announcement because the length of the as path (2) is shorter on the hijacked route than on the legitimate one propagated via AS11 (whose AS path length is 3). Consequently, AS12 will propagate the hijacked route to its neighbour AS13. (Normally, AS12 would also announce the hijacked route to AS11, and then AS11 would have to choose between the legitimate or the hijacked route. The choice will depend on other factors than the AS path, and can be down to the routing policy or the BGP tie-breaking rules, as described in Section 2.1.6.)

2.2.1.4 Man in the Middle

In order to not only receive the traffic destined to a prefix but also to be able to forward the traffic back to the legitimate network, the the BGP-scale **man in the middle** attack presented by [Pilosov *et al.* 2008] combines a forged AS link attack with a subnet attack. First, a usable AS path between the hijacking AS and the victim AS should be identified. This path will be used to forward the traffic back to the owner, which we call the *return route*. Once such a path is identified, the hijacking AS announces a covering set of more specific prefixes than the victim, and includes in the AS path the victim's AS, and the ASes of the return route. Such an announcement effectively redirects all traffic destined to the victim to the hijacking AS. Because the new announcements are more specific, they will be automatically chosen by BGP to forward traffic (like a subnet attack). Because the new announcement contains the ASNs of the ASes in the return route, the ASes on the return route will not select it because it is not feasible. As a result, the hijacker will be able to forward all traffic on this link but will still receive all the traffic destined to the victim, apart from the traffic between one of the ASes inside the return route and the victim. Consequently, this attack is very effective, but also very stealthy because the legitimate owner will not be seeing any red flag.

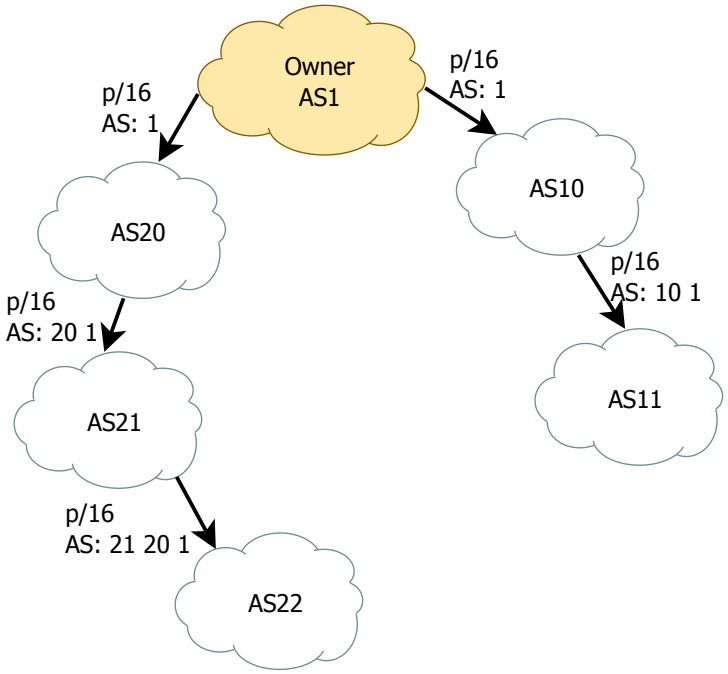


Figure 2.4: Man in the middle attack: original routes to the victim

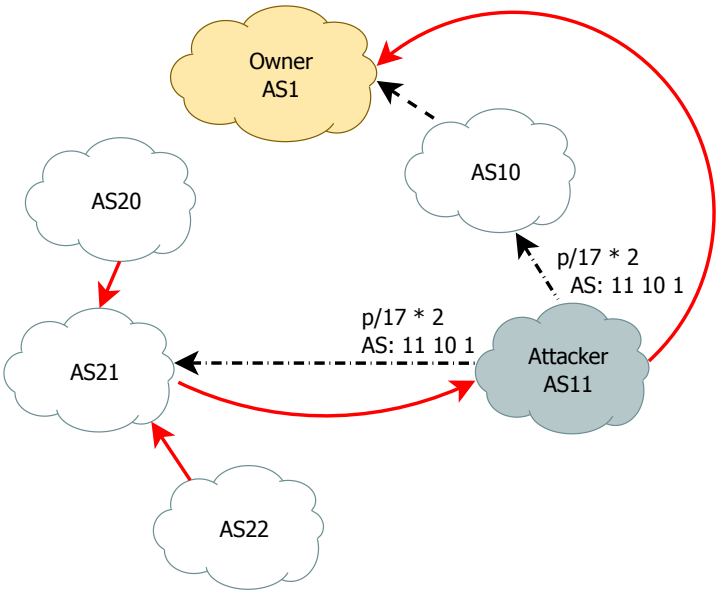


Figure 2.5: Man in the middle attack: hijacked routes

A man in the middle attack is depicted in Figure 2.4 and Figure 2.5. Figure 2.4 shows the legitimate route to p's origin: AS1. If AS11 decides to do a man in the middle attack on p, it needs to be able to redirect the traffic to AS1. In order to do so, it identifies that the legitimate route between AS11 and AS1 is through AS10. In Figure 2.5, AS11 has propagated the two /17 prefixes covering p/16. It has added AS1 and AS10 in the AS path. Because of this reason, AS10 will not consider the /17 route. But every other AS on the network will, such as AS21, which receives the route, and forwards it to its neighbours. In the end, the red plain lines show that all traffic destined to p/16 is first rerouted through AS11. AS11 is then able to send the traffic back to AS1 by using AS10, who remains unaffected.

2.2.1.5 Dormant Space

An interesting phenomenon, observed by [Vervier *et al.* 2015], happens with the **dormant space**, which is composed of the allocated, but unannounced IP space. One such example is IBM's IP space. IBM was allocated a /8 prefix, but only announces a handful of prefixes in BGP, the largest of which is a /16. By announcing a prefix contained in IBM's range, but that does not collide with IBM's own announcements, a hijacker can use IBM's IP addresses (and thus masquerade as IBM) in order to carry out malicious activity on the network.

2.2.1.6 Unallocated Prefixes

Out of the whole IP space, a number has been assigned for use, and the rest, called the **blackspace**, still remains to be affected to an end user. By announcing a prefix that is part of the blackspace, a hijacker can inject traffic into the network. Moreover, since the prefix does not belong to anyone, it is unlikely to be actively monitored and, thus, the attacker is more likely to remain undetected. Chapter 5 is dedicated to the study of this phenomenon.

2.2.2 Occurrences of Prefix Hijacking

The events detailed in this Section are real-world public occurrences of BGP hijackings and illustrate the destructive effects that results. Because prefix hijacking can result either from an accidental router misconfiguration or a deliberate attack on the routing infrastructure, the events are classified accordingly.

2.2.2.1 Accidental Occurrences

The AS7007 Incident
The first BGP-related incident on the Internet dates back to April 25, 1997 when AS7007 – attributed to MAI Network Services (MAI), a regional ISP in Virginia, USA – started, as the result of a misconfiguration, announcing highly specific routes to one of its providers: Sprint (a large backbone network) [Bono 1997; Freedman 1997; CNET News 1997]. Sprint didn't filter out those announcements and propagated them. Because their network is so large, the erroneous routes completely contaminated the Internet, resulting in a (very) large-scale subnet attack coupled with an ownership attack. When the folks at MAI noticed what was happening (within 15 minutes), they disconnected themselves off the Internet. Somehow, the highly specific routes still existed, resulting in a massive blackhole of the global network, which is the worst thing that can happen [van Beijnum 2002]. The attack lasted a little less than 6 hours.

The TTNet Leak
On December 24, 2004, TTNet (the largest ISP in Turkey) started announcing over 106 000 prefixes to Telecom Italia (TIt) [Popescu *et al.* 2005]. TIt did not set a maximum prefix count on the incoming routes from TTNet, so they accepted the routes and started propagating them upwards. Fortunately, these peers had an upper limit on the number of accepted incoming routes and it was rapidly reached. Unfortunately, specific routes were still propagated, albeit in a small number, which resulted in a virus-like propagation of erroneous routes: everybody got a little bit infected. The event lasted a little less than 12 hours.

Con Edison Hijack
On January 22, 2006, Con Edison (Con-Ed) started originating routes for prefixes *mainly* owned by their customers, resulting in hijacking the prefixes of those who were not Con-Ed customers [Underwood 2006]. Routes were first propagated through UUnet and then through NTT America. In less than 3 hours, Panix (an ISP based in New-York) was completely unavailable. The event lasted more than 12 hours. It is believed that the hijacked routes were former Con-Ed customers, but no explanation has ever been given.

The YouTube Hijack
On February 24, 2008, the Pakistani government decided to forbid access to YouTube [RIPE NCC 2008a]. YouTube is announced with an aggregated /22 prefix. Pakistan Telecom decided to enforce the interdiction by BGP means and announced a /24 prefix of YouTube. That /24 network is the one that contains YouTube's DNS and web servers. Somehow, Pakistan Telecom announced that route outside of their networks, including to their provider, PCCW Global, that did not filter them and propagated the more specific /24 route to the rest of the world. In a little less than 1h30, the

whole traffic for YouTube ended up in Pakistan. Then YouTube counter attacked and announced the covering /25 subnetworks, which resulted in getting the traffic right back to them. Two hours and thirty minutes after the start of the attack, PCCW Global withdrew the routes originated by Pakistan Telecom and YouTube reaggregated its announcement to the original /22.

The I-Root Incident

China's censorship of the Internet is enforced by erroneous DNS responses for black-listed websites. Someone trying to get access to YouTube (for example) from within China will most likely get an unrouted IP address for the hostname www.youtube.com. While this is the desired behaviour inside Chinese territory, it is certainly undesirable anywhere else. The DNS root server I is run out of China. Two versions of the server exist: one for China-based request, and one for everybody else. Between March 3 and March 25, 2010, the Chinese version was somehow made visible to the global network [Zmijewski 2010]. Anyone happening to be redirected to the I-root DNS for an "undesirable" website would then suffer from DNS cache poisoning. It was very unlikely, however. First, root DNS servers use anycast to load balance requests. So, not only had the client to choose the I root server, but also the Chinese version of it. Because anycast depends on BGP's route preference, the chances for anyone not close to China to be affected by that misconfiguration were quite small.

The China Telecom Leak

On April 8, 2010, China Telecom announced 37,000 prefixes instead of the normal amount of 40, affecting networks owned by CNN, Dell, Apple, US DoD, France Telecom, Amazon Deutschland, and others for a little less than 20 minutes [Toonk 2010; Labovitz 2010a; Labovitz 2010b; McMillan 2010; Wolf 2010]. Less than 15% of the global routing table was affected [Labovitz 2010a; Labovitz 2010b; Alperovitch 2010], and impact in North-America and Europe was minimal [Labovitz 2010a; Kisteleki 2010]; although impact in Asia was certainly not negligible. The incident raised awareness about the fragile security of Internet routing in mainstream medias who were not afraid of drawing conclusions of cyber-war. The consensus, however, is that the incident was the result of a misconfiguration.

2.2.2.2 Intentional Occurrences

Fly-By Spammers

Spectrum agility was first reported by [Ramachandran *et al.* 2006] to describe so-called *fly-by spammers*, who appear to announce (and typically hijack) a black-spaced class A prefix for a short period of time in order to use the IP addresses for spamming. Later, [Hu *et al.* 2007] further correlated BGP hijack alerts with spam sources from [Ramachandran *et al.* 2006]. In theory, spammers using this technique are able to circumvent backtracking and traditional IP-based blacklisting due to the short-lived nature of the attack. Results presented in [Vervier *et al.* 2013] suggest that fly-by spamming has fallen out of use.

The Link Telekom Hijack

On August 20th, 2011, a network administrator of a Russian telecommunication company called Link Telecom (AS31733) complained that their prefix had been hijacked by a spammer [Spirin 2010a; Spirin 2010b]. While trying to announce the prefixes, he found that the announcement was rejected by their upstream because of malicious activity associated to them. At the same time, these prefixes were already announced by an American ISP called Internap (AS12182). With much effort, Link

Telecom was able to obtain help from Internap's upstream and to block the fallacious announcements [Spirin 2010c].

This occurrence has been analyzed by [Biersack *et al.* 2012; Vervier *et al.* 2013], and in great detail by [Schlamp *et al.* 2013]. The hijack occurred between April and August 2011. The hijacker successfully took control of the prefixes by registering a defunct domain name of Link Telekom, which was going bankrupt at the time. Using this domain name, the hijacker was able to fake its identity, and to appear posessing all the legitimate authorizations in order to announce the prefix through from Internap. This is the main reason why Internap was reluctant to cooperate with Link Telekom when they were contacted about the hijack.

The Spamhaus Hijack
In March 2013, [Spamhaus] was the target of one of the most massive DDoS attack ever recorded [Bright 2013]. At the same time, on March 16th, Cyberbunker (AS34109) started announcing 204.16.254.40/32 in BGP, which is the IP address for 0.ns.spamhaus.org [Shaw 2013; Snijders 2013; Hanford 2013; Toonk 2013]. This server is normally used to check if the IP address of an incoming email is known to send spam. By hijacking the address, Cyberbunker was able to setup a server that would always reply that the queried IP address was linked with spam. The hijack lasted a couple of hours, during which a lot of email was wrongly classified as spam.

Bitcoin Theft
Between February and May 2014 networks from several different ISPs were hijacked in order to steal cryptocurrency mining power [Litke *et al.* 2014]. It appears that the hijackers targeted 51 networks from 19 ISPs in order to redirect miners to their own mining pool. By redirecting the computing power to this malicious pool, the computers were still getting work to do, but not being rewarded any Bitcoins. In total, there were 22 hijacks that lasted around 30 seconds each [Greenberg 2014], and the hijacker was able to steal at least $83k worth of cryptocurrency.

Dormant Space
[Vervier *et al.* 2015] was able to uncover over 2,000 malicious prefix hijacks, some of which have been confirmed by the affected parties; consequently showing that a small but agile number of spammers are successfully able to stealthily, routinely, and persistently hijack a large number of IP addresses to send spam and host scam websites. These spammers specifically target dormant IP space, i.e. IP space that was not globally announced before the uncovered events.

Blackspace
Chapter 5 will present a number of cases where the hijackers took advantage of unallocated IPv4 address space in order to host spam and scam infrastructures.

2.3 Securing BGP

In Section 2.2, we presented prefix hijacking and the various vectors to carry out such an attack against the BGP infrastructure of the Internet. Then we presented a series of hijacking occurrences, resulting from both, accidental and intentional operations. These occurrences illustrate that the

inter-domain infrastructure is quite fragile and that the effects of prefix hijacking are large-scale and not only theoretical. In this Section, we look at some of the various propositions that have been made in order to allow BGP to work in a secured environment.

An elegant threat model for the BGP protocol is presented by [Huston *et al.* 2011]. It illustrates just how many distinct elements need to be secured. First, the communication channel between two peering routers can be attacked, meaning that the BGP session itself can be attacked, but also that the BGP messages can be removed, altered, or replayed. Second, the identity of the peering router can be faked, meaning that a remote attacker could masquerade as a legitimate peer. Third, the authenticity, completeness, and validity of routing information received from a peer should be verified. Because of mutual trust (see Section 2.1.8), routers will accept any incoming route with any associated path attributes. Section 2.2 presented numerous ways in which these can be faked.

On the other hand, the inter-domain routing infrastructure is more than just the BGP protocol. Actually, the most straightforward way to attack the routing infrastructure is to target the router which runs BGP, not necessarily the BGP protocol itself. Specifically, routers running outdated versions of their operating systems could be attacked with low-level techniques such as buffer overflow. Credentials to access the routers are not always configured appropriately (e.g. default passwords) [Huston *et al.* 2011]. Once the router is compromised, the attacker can use it as a stepping stone for another attack, such as prefix hijacking.

In the remainder of this Section, we look at a few proposals that have been made in order to secure each element of the routing infrastructure. First, we consider some methods to secure the routing infrastructure itself. This discussion will be brief as it is not the main topic discussed in this Dissertation. Then, we present a series of techniques that would provide BGP with the necessary tools to render prefix hijacking attacks impossible.

2.3.1 Securing BGP Sessions

As mentioned earlier, BGP uses TCP in order to establish a BGP session between two peering routers, thereby relying on TCP for reliable transmission (acknowledgements/retransmissions, sequencing), and fragmentation. This reliance on TCP also implies that BGP is subject to TCP-based attacks, like any protocol depending on TCP [Butler *et al.* 2010]. For example, eavesdropping on BGP sessions is possible because the communication channel is not encrypted. However, the information shared during a BGP session is not sensitive. Attacks against message integrity are possible, in particular man-in-the-middle attacks, which could allow an attacker to tamper with BGP update messages by inserting, modifying, or deleting routes, or by breaking the BGP session by tampering with the keep alives, or notification messages. Finally, SYN flooding, FIN or RST spoofing are also possible, which may lead to a denial of service of the BGP router.

A number of proposals have been made in order to protect BGP sessions. [RFC2385] proposes to use a *MAC* (Message Authentication Code), by hashing the content of the TCP messages and of a shared secret between two peering routers with the MD5 hash function. [RFC5925] makes this proposal more flexible by introducing TCP-AO (TCP Authentication Option), where the hash function can be configured, and the shared key can be updated automatically, and without having to tear down the TCP connection. These two schemes protect the BGP sessions against integrity and replay attacks. The TCP connection between peering routers can also be protected with IPsec, which, depending on the way the IPsec tunnel has been configured, can even provide confidentiality.

2.3.2 BGP Ingress Filtering

In Section 2.1.8, we explained how implicit mutual trust is the result of the BGP route selection process. There is no step between the moment when a route is inserted in an Adj-RIB-In and the moment when it enters the Loc-RIB that verifies that the route is not fallacious. However, Section 2.1.6 illustrates that the router still enforces a local policy. This local policy can be used to create an ingress filter on routes received from peers. This filter is able to tell the selection process that a particular route should never be considered as the favourite one. In other words, it has the power to tell the BGP process that a route should *always* be rejected. As a result, a number of good practices are usually associated with configuring a BGP router. These standard guidelines are as follows:

- routes learnt from external peers to any of the router's own AS prefixes should be rejected [van Beijnum 2002],
- routes to very specific IPv4 prefixes – i.e. anything that is smaller than a /24 prefix – should be rejected[1] [Bellovin *et al.* 2001],
- routes to very large IPv4 prefixes – i.e. anything that is larger than a /8 prefix – should be rejected[2] [Bellovin *et al.* 2001],
- discard any announcement containing a private ASN in the AS path [Butler *et al.* 2010],
- specify a maximal number of routes learnt through a given peer [van Beijnum 2002],
- ingress filter routes learnt from a peer [Butler *et al.* 2010].

While some of these guidelines are easy to implement, they are in no way mandatory. Whether or not they are followed depends on the choice made by the administration of an autonomous system, even though some peering agreements explicitly enforce policies. However, populating and maintaining a list of routes that peers are allowed to be announce takes a lot of (mostly manual) work for small to medium networks, and is close to impossible for large networks that may announce thousands of distinct prefixes. It is consequently unrealistic to expect a large network, such as a tier-1 ISP, to maintain such filters on its peers.

2.3.3 Securing the Routing Information

In Section 2.3.2, we looked at the only built-in mechanism that BGP has to defend against prefix hijacking: building a set of filters that aim at preventing bad behaviour. We also saw that it is unrealistic to expect large networks to use such filters because they would be impossible to maintain. As a result, several proposals have been made in order to provide the BGP protocol with the tools it needs to validate the routing information by itself. In general, the desired features for securing BGP paths are: [Butler *et al.* 2010]

- **origin authentication**, which certifies that the originating AS is the righteous owner of the prefix,
- **path authentication**, which certifies that an AS is authorized to announce the route for the prefix,

[1] Even though this good practice was introduced in order to reduce the global routing table size, it can be useful in order to limit the scope of a prefix hijacking attack.

[2] See footnote 1.

- **topology authentication**, which certifies that peering relations announced in the AS path are topologically real (i.e. no router has tampered with the AS path).

We will see that, even though this description sounds quite simple, the steps needed to enforce these three authentications are quite complex.

2.3.3.1 Secure BGP

Secure BGP (S-BGP), presented in [Kent *et al.* 2000], is the most comprehensive proposal to secure BGP [Huston *et al.* 2011]. It provides *origin authentication*, *path authentication*, and *topology authentication* [Butler *et al.* 2010]. It is based on the X.509 certificates and *PKI* (Public Key Infrastructure), *attestations*, and *IPsec*.

S-BGP uses **PKI** to verify an entity as the righteous owner of one or more prefixes. Moreover, **X.509 certificates** are used to bind AS numbers to organizations, and also those organization to routers. The top-level certificate would be the IANA's, delegating powers to RIRs. For example, an ISP can certify that a router is acting on its behalf by issuing a PKI-signed certificate to that router.

The second element is *attestations*. There are two kinds of attestations: **address attestations** and **route attestations**. Address attestations are signed by the owner of a prefix and authorize a set of ASes to originate a route to the prefix. Route attestations are signed by an S-BGP enabled router and targeted towards the set of ASes to which the BGP update message will be sent. In other words, a route attestation is signed by each AS on the path, and thus verifies the AS path.

Lastly, **IPsec/ESP+IKE** is used for securing data exchange between two peering routers, providing authentication, integrity, and anti-replay.

In order to validate a route received by BGP, a router have the following elements:

- an address attestation for each owner of a prefix in the message,
- a certified public key for each one of these entities,
- a route attestation for each AS in the AS path,
- a certified public key for each S-BGP router that signed the route attestation.

In order for S-BGP to work, each owning entity needs to have a certified public key. Every S-BGP enabled router must have one too. This leads to a large amount of keys, which doesn't speed up the validation process, which can lead to increasing the convergence time by a factor of two. Moreover, concerns have been raised over the substantial storage requirements for route attestations, and on the performance of existing routers to support such operations. All in all, a large-scale deployment of S-BGP in the real-world appears unlikely.

2.3.3.2 Secure Origin BGP

Secure Origin BGP (soBGP), defined in [White 2003], is more flexible than S-BGP and aims to reduce the overhead needed to process incoming update messages, particularly in relation with the validation of route attestations [Huston *et al.* 2011]. It provides *origin authentication* and *topology authentication* [Butler *et al.* 2010].

soBGP uses three different types of certificates. The first certifies a soBGP-enabled router's public key. The second certifies, among other things, local network topology (i.e. who are the real peers). The third certifies address ownership.

Using the local network topology certificate, a global database with the corresponding topology graph can be constituted, and each soBGP router can thus have a consistent view of the network. That database is used to verify BGP announcements. Unfortunately, this infrastructure is fundamentally static, only changing when a new certificate is issued to reflect a topology change (as opposed to S-BGP's route attestations which provide a dynamic view, since certificates are joined with the update messages). Thus, routers may discard legitimate announcement due to lags in topology update. Additional infrastructure is required to synchronize topology changes among all ASes. Moreover, forged paths are still possible if they are consistent with the soBGP topology, but not representing actual physical routes.

Computational overhead for validating signatures is avoided, in soBGP, by authenticating long-term routing elements before joining a BGP session. This authenticated data is stored locally on each router. Transient elements (such as AS paths) are checked for correctness (e.g. with the topology database), instead of validated through PKI. As a result, soBGP cannot guard against modifications to these transient elements that are not reflected in the static databases.

Finally, soBGP provides several deployment options, leaving much of the behaviour of soBGP-enabled routers to the administrator's tastes. While these options give better deployment opportunities to soBGP, they could introduce interoperability issues.

2.3.3.3 Pretty Secure BGP

Much like soBGP, **Pretty Secure BGP** (psBGP), proposed in [van Oorschot *et al.* 2007] aims to define a model that is more flexible than S-BGP in order to find a trade off between security computation overhead and available computational power [Huston *et al.* 2011]. psBGP provides *path authentication*, weak *origin authentication*, and weak *topology authentication* [Butler *et al.* 2010].

A key design aspect of psBGP is that the certificates related to ASes can be handled by the PKI, while prefix certificates cannot. The authors argue that this due to the lower number of ASes, and their relatively static allocation, which is the total opposite of prefixes. Instead, they rely on a concept similar to the *web of trust* from PGP [Zimmermann 1995], where a reputation score is computed based on how each AS rates each other AS based on how much trust they put in them. As a result, origin authentication is a decentralized process. Each AS creates a **PAL** (**Prefix Assertion List**) in which it certifies its own addresses, and also the addresses of its peers. Consistency for an AS can be verified by checking the peers' PALs. Path authentication is performed much like in S-BGP, by getting each router on the path to sign the AS path. In this regard, it is stricter than soBGP.

In the end, the principles behind psBGP are a bit contradictory. On the one hand, psBGP requires a standard, structured PKI in order to validate ASes, but rejects it for prefixes validation, instead relying on a side channel. Moreover, the large-scale deployment of a hierarchical RPKI (see Section 2.3.3.5) by RIRs proves that prefix validation can be carried out through this medium. Relying on a web of trust means that some ASes need to trust some other ASes that they are not necessarily connected to, which is the basic design behind BGP, even though psBGP attaches a level

of trust to it. Additionally, the use of PALs introduces a large overhead in the validation of origin, and the validation of the AS path requires, much like with S-BGP, a large amount of computational power.

2.3.3.4 Interdomain Route Validation

The **Interdomain Route Validation** (IRV) system, presented in [Goodell *et al.* 2003], defines both an architecture and a query-response information dissemination protocol. Instead of modifying the way BGP works, it aims at proposing a side-channel through which the routing information can be checked for accuracy [Huston *et al.* 2011]. The idea is to provide a service that ASes can join without adding a lot of strain on the current infrastructure. It provides *origin authentication*, *path authentication*, and *topology authentication* [Butler *et al.* 2010].

Each IRV-enabled AS hosts an **IRV server** which provides authoritative information about the prefixes originated by that AS. Actually, the IRV server could even contain the full routing policy of the AS by using **RPSL** (Routing Policy Specification Language) [RFC2622; RFC4012]. When receiving an update message, the BGP router contacts the local IRV server. The local IRV server then contacts the IRV server of the ASes included in the update message to find out if the routing information is valid. One advantage is that an AS can decide how much checking it wants done. For example, they can specify a weak verification of the AS path, or only opt for checking a certain percentage of all incoming messages, etc. This enables the AS to balance the security gain and the additional work it needs to perform to suit its operations and infrastructure best. Moreover, IRV servers can alter their responses based on the incoming query's IP address, therefore possibly strengthening its routing policy by invalidating announcements that should have remained within the AS (i.e. leading to a route leak), or should not be forwarded by a neighbour.

There are a number of issues with IRV. Since the routes verifications are carried outside of BGP, the distant IRV server needs to be contactable. This causes problem when a new route appears, and during outages. Moreover, [Goodell *et al.* 2003] does not specify if the routes need to be IRV-validated before they are accepted. As a result, this is left to the choice of the local policy, which leads to similar inconsistencies that already exist in the network today.

2.3.3.5 Resource Public Key Infrastructure

The **Resource Public Key Infrastructure** (RPKI) is described in [RFC6480], and (currently) provides *path authentication*[3]. A design goal of RPKI is to adopt a structure similar to the existing resource allocation structure, thereby only requiring an extension to the existing methods and practices. The system is based on three elements: the infrastructure, signed routing objects, and a distributed repository system.

The infrastructure is much like every other PKI: it is hierarchical and is made up of X.509 certificates, which are called **resource certificates**. These resource certificates attest the rightful allocation of either IP addresses or AS numbers. There are two kinds of resource certificates:

[3]The set of RFCs describing RPKI refer to this feature as *origin validation*. This may be confusing because our classification, which we borrowed from [Butler *et al.* 2010], uses the term "origin *authentication*" to prove that an AS owns a prefix. A combination of certificates and ROAs in RPKI may lead to origin authentication as such, but the standard idea behind ROAs is that an AS is indeed allowed by the prefix owner to generate the prefix. This AS may belong to the owner as well.

certificate authority (CA) certificates and **end-entity** (EE) certificates. Any entity that is allowed to issue a resource is a certificate authority, and thus, possesses a CA certificate. The private keys associated to a CA certificate are used to sign other certificates; End-entities certificates are used to verify the way resources are used. Basically, they are used to sign *Route Origination Authorizations*.

Route Origination Authorizations (ROAs) are created by a resource owner and certify that an AS is allowed to originate the prefix. The validity of an ROA is verified by making sure that the entity who signed the ROA is the owner of the prefix. ROAs are not certificate, but signed information (i.e. they are not codified with the X.509 standard). They are signed with the private key related to an EE certificate. In order to map the EE certificate to an entity, a certificate binding the CA certificate and the EE public key should be issued by the entity. Consequently, the private keys behind EE certificates can be one-time-use, which simplifies key management. It also implies that entities that wish to issue ROAs need a CA certificate, even if they do not redistribute resources. Additionally, ROAs can specify the specificness of the prefix that an AS is allowed to announce. For example, an ROA linking AS1 with 10.0.0.0/16-24 implies that AS1 is allowed to announce the IP space induced by 10.0.0.0/16, but is allowed to announce more specifics within this IP space, provided that they are not longer than a /24. In order to revoke the rights of an AS to announce a prefix, standard Certificate Revocation Lists (CRLs) are maintained and should be taken into account to validate the ROA. Because an ROA is not a certificate by itself, the CRL revokes the EE certificate associated to the ROA(s).

A **distributed repository system** is used to make available all signed objects to the BGP community, and detailed in [RFC6481]. When CA certificates, ROAs, and CRLs are generated, they are uploaded to the repository from which they can be downloaded (primarily by ISPs). The repository system is *distributed* because it does a single repository does not contain every existing signed object. Every certificate authority uses an authoritative location to publish its certificates and signed objects. For this reason, every certificate contains pointers to the repository locations of:

- the repository where the signing certificate was published,
- the repository where the object was published.

Using these pointers, the certificates can be validated by traversing the hierarchy of the PKI, in a distributed manner.

RPKI is actively supported by RIRs and a large portion of the (vocal) BGP community. Each RIR has now setup an RPKI repository, through which they certify which portion of address space has been delegated to ISPs. Moreover, major router vendors have now implemented RPKI in their software [Cisco 2014; Juniper 2013], which enable routers, if configured to do so, to verify the ROAs for incoming routes. Additionally, work to extend RPKI to secure the AS path, in other words, to provide *topology authentication* is underway at the IETF [SIDR]. Consequently, RPKI appears to be the only method to secure the interdomain routing information that has currently gained momentum. However, at the time of writing (August 2015), only around 6% of the announced IPv4 address space, and around 9% of the announced IPv6 space has been secured with RPKI. Progress can be monitored using [RPKI Dashboard].

2.4 Detecting Prefix Hijacking

Section 2.3 detailed the security issues related to BGP. We mainly focused on the implicit mutual trust that binds every ASes (introduced in Section 2.1.8) and detailed a number of proposals that

have been made in order to ensure that the routing information is secured. Out of all the proposals that have been made, only RPKI (Section 2.3.3.5) appears to have been deemed feasible by the BGP community. As a result, RIRs and router vendors have taken steps to provide the necessary infrastructure to actively use RPKI in route selection, thereby making sure that the active selected route is valid. However, we also underlined that there is currently a very low number of routes that have been secured through RPKI. Moreover, RPKI currently only certifies whether an AS is authorized to announce a prefix on the owner's behalf. At the same time, Section 2.2 introduced a number of way to carry out a prefix hijacking attack, most of which do not involve announcing the prefix from an unauthorized AS. Actually, because RPKI does not currently provide topology authentication, adding the legitimate origin ASN as the last element of the AS path would be enough to validate the announcement, even if the route has been hijacked. Consequently, RPKI is mostly an architecture that prevents router misconfigurations from causing large-scale network outages, as some of the cases presented in Section 2.2.2.1, not a system that can prevent (malicious) prefix hijacking once and for all. Even though it is probably the best first step towards a fully secured interdomain routing, it is not currently sufficient. And it is unlikely to become sufficient in the close to mid-term future.

Considering these facts, it is key to be able to detect prefix hijacking. Even if prefix hijacking cannot be prevented, active counter measures can be taken in order to limit the scope of its side-effects. In this Section, we present the current state-of-the-art techniques. Prefix hijacking detection techniques are usually classified according to their detection method:

- methods based on the BGP control plane, i.e. the routing information conveyed in BGP update messages, aim to build a model of the Internet. Once an update messages suspiciously affects this model, an alert is raised. These techniques are detailed in Section 2.4.1.
- methods based on the data plane probe the way packets flow between a monitoring point and a monitored network. Once the topology has sufficiently changed, an alert is raised.
- more recently, Argus, which is discussed in Section 2.4.3.2, has merged these two approaches into a larger scheme.

2.4.1 Control Plane Techniques

In this Section, we present the main techniques that aim at detecting prefix hijacking from the control plane. The control plane is composed of the BGP information that is exchanged among BGP routers. Depending on the detection technique, the control plane is either composed of the content of BGP update messages, or the content of the BGP RIB of the *collector router*. (A collector router is a BGP router that has been configured to regularly dump a number of BGP information into archive files that are generally made public. We will detail further the BGP data collection infrastructure and its limitations in Section 2.5.)

2.4.1.1 PHAS: a Prefix Hijack Alert System

The idea behind PHAS, explained in [Lad *et al.* 2006], is to provide, as the name suggests, a prefix hijack alert service. Based on the premise that the prefix owner is the only one that can unambiguously distinguish a legitimate route change and a hijacking attack, the authors offer the possibility for network administrators to subscribe to monitoring services on a prefix p and to be notified of an origin AS change event somewhere on the Internet, in near real-time. PHAS was available for use free of charge at [PHAS] between 2006 and the autumn of 2009.

Detection method
The system builds, over time, a set $\mathcal{O}_p(t)$ containing the different origins ASNs for prefix p seen at time t on every router where PHAS is deployed. PHAS alerts the users whenever $\mathcal{O}_p(t) \neq \mathcal{O}_p(t-1)$. To avoid notifying users for repeated origin changes, the authors introduce a *time window*. The origin set is extended to $\mathcal{O}_p(t-k,t)$ that contains every origin ASN seen for prefix p during the time $[t-k,t]$, on every PHAS enabled router. The system then generates an alert when $\mathcal{O}_p(t-k-1,t-1) \neq \mathcal{O}_p(t-k,t)$. This trick avoids repeated origin events, but will still generate an alert whenever a new originating AS appear, or whenever a known origin AS disappears, notifying users only on potentially wrong origin ASes. It is important to note, however, that the addition of a time window delays notifications of loss events.

In order to reduce this notification delay for networks that do not present a lot of events, an *adaptive window size* is introduced. On top of a windowed origin set, each prefix is assigned a penalty S_p. When an update message is received for prefix p, S_p is increased by $1/2$. The size of the window for p is then $2^{\lfloor S_p \rfloor}$. S_p decays exponentially, determined by a time value.

Finally, users have the possibility to add filters to PHAS to prevent alerts. The generated alerts pass through the filter before being sent off to the user.

Extensions
The authors also provide possible extensions to PHAS to deal with other types of attacks from the origin attack. For subnet attacks, a mechanism based on watching modifications made to the set $\mathcal{SP}_p$ that contain the advertised subprefixes of p is proposed. If no subprefixes are advertised, $\mathcal{SP}_p = \{\ \}$. For last hop attacks, the suggested method is to watch the set $\mathcal{L}_A$ containing the last hops witnessed for prefixes with A as the origin AS.

Using these two additional sets in PHAS helps to further identify hijacking attempts. However, the subprefix set (resp. the last hops set) is potentially huge for a network such as 12.0.0.0/8 (resp. for a tier-1 ISP).

Accuracy
PHAS is useless against stealthy hijackings, such as forged AS links (see Section 2.2.1.3) [McArthur *et al.* 2009] and man in the middle (see Section 2.2.1.4) [Pilosov *et al.* 2008].

2.4.1.2 PGBGP: Pretty Good BGP

The goal of PGBGP is not only to detect hijacking events, but to improve overall routing quality and reliability. The core idea behind PGBGP, presented in [Karlin *et al.* 2006], is that "unfamiliar routes should be treated cautiously when forwarding data traffic".

PGBGP defines a set of *normal* data containing the prefix, its origin AS, a timestamp of the last received update. The normal data set and the router's RIB are used to create a history for known prefixes and origins. Obviously, at startup, there is no known history, and all routing updates are accepted for h days. Afterwards, incoming routes that would alter the state of the normal behaviour are quarantined for s days. The quarantined routes are considered as suspicious. After that time, they are accepted, if still in the routing table. This quarantine mechanics prevents short-term erroneous announcements from disrupting routing. Finally, PGBGP removes data from the history if it has not been announced for h days.

As any incoming route is tested against the history, hijacking attempts, arriving with a new origin AS, do not match known history for that prefix and are therefore quarantined. While considered as suspicious, the old, trusted routes are used for packet forwarding.

To avoid subnet attacks, PGBGP checks if the new incoming prefix is a subnet of a known one. If it is, and the route to the subnet does not traverse the larger prefix AS, it is suspicious. However, forwarding packets along the trusted route may be useless if routers along that route have been compromised. Therefore, PGBGP tries to avoid forwarding packets to neighbour routers that have announced the suspicious route.

Less specifics of known prefixes are always accepted by PGBGP as the authors believe that it is the result of a new network destination, not of a hijack, because traffic destined to the original network will use the legitimate, more specific route.

This technique was classified here, as part of the detection techniques, and not as part of Section 2.3 (Securing BGP) as a prevention technique, because PGBGP was used as source to the [Internet Alert Registry], a tool which was aimed at reporting hijacks on the Internet. The [Internet Alert Registry] has been inactive for at least 5 years now.

2.4.1.3 Directed AS Topology

The idea behind this method is to build a directed graph of the network topology and to compare the AS paths in update messages against it. It is presented in [Qiu *et al.* 2007].

AS relationships
Because the method is heavily dependent on one of the author's previous works, [Gao 2001], the main ideas of that article are summarized here.

The BGP policies are heavily influenced by commercial contractual relationships between organizations. Typically, these commercial agreements can be classified into different categories:

- *customer* $\leftrightarrow$ *provider*: a customer pays a provider for Internet access. Therefore, a provider becomes the transit network for its customers' traffic. The opposite is not true: a customer does not transit traffic between its different providers.
- *peering*: peering networks agree to exchange traffic among themselves, free of charge.
- *mutual transit*: these networks agree to provide each other with connectivity to the Internet.
- *mutual backup agreement*: these networks agree to provide each other with connectivity to the Internet, in the event that one of the networks' providers fails.

The two last categories are referred to as *sibling* relations.

Based on these categories, the BGP policies can be translated into the following rules:

- Exporting to provider: own routes and routes to customers are exported to a provider. Usually other providers' routes are not exported.
- Exporting to customer: own routes, providers' routes, as well as other customers' routes and peers' routes are exported to a customer.

- Exporting to a peer: own routes and customers' routes are exported to a peer. Usually, providers' routes and other peers' routes are not exported.
- Exporting to a sibling: own routes and customers' routes are exported, as well as providers' and peers' routes.

Finally, the valley-free property illustrates how network agreements shape the traffic on the Internet. The **valley-free property** states that "after traversing a provider-to-customer or peer-to-peer edge, the AS path cannot traverse a customer-to-customer or peer-to-peer edge" [Gao 2001]. In other words, a provider-to-customer edge cannot be followed, at some point, by a customer-to-provider edge, but only by provider-to-customer or sibling-to-sibling edges. Otherwise, a network is a transit for traffic between two of its providers. In the same fashion, a peer-to-peer edge can only be followed by a provider-to-customer or sibling-to-sibling edge. If that were not the case, a network would effectively be the provider for one of its peers.

Furthermore, a *downhill path* is defined as a sequence of provider-to-customer or sibling-to-sibling edges. An *uphill path* is a sequence of customer-to-provider or sibling-to-sibling edges. Using these definitions, it is easy to visualize that the shape of traffic as viewed form an AS level, when export policies are correctly defined, is in the shape of a mountain (without a valley).

Detection method

The authors first observe that the majority of BGP routes are stable and legitimate. Thus, these routes can be learnt over time. Let's consider a prefix p. An observer receives a legitimate update message for p, containing the AS path $a_k, \ldots, a_0$. In other words, ASes a_i and a_{i-1} $(i = 1, \ldots, k)$ are neighbours. A *directed* AS link is a link such as $a_i \rightarrow a_{i-1}$ $(i = 1, \ldots, k)$. Moreover, a_i (resp. a_{i-1}) is upstream (resp. downstream) of a_{i-1} (resp. a_i). The directed links also indicate the import/export policies of the involved ASes. A downstream (resp. upstream) AS allows route to be exported (resp. imported) to an upstream (resp. downstream) AS [Gao 2001].

Let's consider, at time t, the sets $\mathcal{A}(t-k, t)$ and $\mathcal{L}(t-k, t)$ containing the associations between a prefix and an origin AS number and the directed AS links, respectively, seen between time $t-k$ and t (i.e. in a time window of size k).

Whenever an update message reaches the observer, the system verifies that the AS links deduced from the AS path of the message are valid (i.e. are they in set $\mathcal{L}$?). If the links check out, the system verifies the association between the prefix and the origin AS (i.e. is it in set $\mathcal{A}$?).

If an extracted $a_i \rightarrow a_j$ association from the AS path does not belong to $\mathcal{L}$ but $a_j \rightarrow a_i$ does, there is a policy violation, and the link is a *redistribution link*, which violates the valley-free property. If $a_j \rightarrow a_i \notin \mathcal{L}$, the path is a *fake link*: the announced neighbouring ASes are not really neighbours. It is highly likely that someone tampered with the AS path. Also, if $(p, a_0) \notin \mathcal{A}$, there is prefix hijacking. Furthermore, if $(p, x) \in \mathcal{A}$ for $x \neq a_0$, it is an ownership attack. If $(q, x) \in \mathcal{A}$ with $q \subset p$ (i.e. q is more specific than p), it is a subnet attack. Finally, if $(q, x) \in \mathcal{A}$ with $q \supset p$ (i.e. q is less specific than p), it is a supernet attack, i.e. possibly an attack on dormant space.

Of course the scheme will only work if the built model of the network (sets $\mathcal{A}$ and $\mathcal{L}$) are good enough. Therefore the initialisation phase is very important. The authors propose heuristics to remove alerts generated by transient routes, path extensions (which are the result of address suballocation), usual BGP misconfigurations, (de)aggregations, sibling ASes links, address sharing peers and backbone links (seeing a backbone tier-1 ISP in an AS-path is normal).

Accuracy
The authors announce a false positive rate as low as $0,02\%$ and an average of 20 raised alerts a day. They have a nearly 100% accurate detection on documented incidents.

However, and it is very important, the quality of the calibration data must be stellar. Moreover, the AS relationships on which this method is based, [Gao 2001], is a model of a perfect Internet, and thus not entirely accurate. Finally, the authors do not provide any accurate method to set the threshold values in the different heuristics used throughout the paper.

2.4.1.4 BGPmon.net

BGPmon.net [BGPmon.net] is a feature-rich BGP alert system that was developed by Andree Toonk, who was, at the time, a network operator at BCNET. Originally, [BGPmon.net] provided services free of charge. Back then, on top of alerting misconfigurations, it was able to detect prefix hijacking, even the man-in-the-middle attack presented in [Pilosov *et al.* 2008]. A network administrator was able add monitoring for their network via classical origin AS watch and regular expression match on the AS path. The system could also interact with Internet Routing Registries (see Section 2.6). The alerts were classified as 3 distinct types of events: own network configuration error, stability issues, and hijacks.

Own network configuration errors are the result of a route leakage because of a router misconfiguration. It is usually characterized by a subprefix being announced from the same location. Stability issues are characterized by a large number of withdrawals for a prefix (a threshold can be set up). [Toonk 2009] presents the study, through the [BGPmon.net] system, of two suspicious routing events: one leading to a MOAS, and one suspected man-in-the-middle attack. Man-in-the-middle attacks were detected with the following: the system looks for a more specific route with a new AS path, and cross-checks with routing registries. If there is no route object with the valid owner as the maintainer and the originating AS, then it is likely to be an attack.

BGPmon.net was originally targeted as network operators, who could configure which alarms they wished to receive via e-mail. Any alarm (notified or not) would still be available from the web interface for later checks or for history purposes.

In 2012, [BGPmon.net] was revamped and setup as a commercial service. In 2015, [BGPmon.net] was acquired by OpenDNS. The system appears to have been upgraded to take into account RPKI ROAs, and take into account outages. Toonk maintains a blog where he provides an analysis of a number of BGP events. Moreover, he is quite active in the BGP operator scene, often presenting BGP hijack trends at NANOG meetings.

2.4.2 Data Plane Techniques

In this Section, we present the main techniques that aim at detecting prefix hijacking from the data plane. The data plane is basically the forwarding path, i.e. the way IP packets flow in the network due to the forwarding action of BGP (and non-BGP) routers.

2.4.2.1 Hop Count to a Reference Point

This technique, presented in [Zheng *et al.* 2007], relies only on the data plane to detect hijacking events. Namely, it uses the distance (expressed in hop count) between a set of N well placed monitors M and the watched network, based on observation that distance measurements to a destination network is relatively stable over time (also confirmed by [Lad *et al.* 2006]).

On top of the N monitors, one (or more) *reference points* per monitor is needed. A reference point is a router topologically close to the network under surveillance, but outside that network's address.

Detection method
First, periodically, a monitor measures its distance d_t (at time t) from the monitored network. It keeps in memory a moving average window of size k, $A(t-k,t)$, that contains, at time t the average value of the measured distances between $t-k$ and t. Because a prefix hijack is likely to have serious consequences on the topological location of the victim network, whenever an attack occurs, d_t will significantly differ from A_t, thus raising a red flag[4]. This step is known as the *network location monitoring*.

Second, when a red flag is raised, the *path disagreement detection* is called. Its goal is to compute the path similarity between the (supposedly affected) AS path to the network and the (normally unaffected) AS path to the reference point. Because the authors rely only on the data plane, they chose to use [iPlane] to map the hop IP addresses to their (supposedly correct) AS number. Once the similarity s_t between these paths has been computed, its value is compared with s_h, the similarity path value that had been computed prior to the hijacking alert. If $s_t/s_h > T$ for a threshold T (i.e. the similarity has decreased dangerously), an alert is raised by the monitor.

Obviously, if multiple monitors raise an alert, chances of being under attack increase.

Limitations
The detection is highly dependent on the choice of the monitors. To ensure a good quality of detection, monitors have to be topologically sparse and use different routes to the network. It may not be easy to locate such positions.

The whole method relies solely on the data plane. An attacker using a tool like Fakeroute [McArthur *et al.* 2009] will render the detection system blind. Moreover, a man-in-the-middle-attack such as [Pilosov *et al.* 2008] also renders the scheme useless.

The path disagreement detection might not be accurate because of the policy of one AS along the way between the monitor and the network/the reference point. An AS radically changing its policy could even trigger an alarm.

2.4.2.2 Fingerprinting the Network

The fingerprinting technique, presented in [Hu *et al.* 2007], is based on the postulate that the hijacking network is different from the legitimate one. Consequently, it is possible to compare the

[4]To be complete, the authors use another window to smooth the instantaneous measurement d_t, as transient problems leads to noise.

fingerprint properties of the hosts on these networks to infer if they are identical or not. Because networks topologically close to the hijacked networks are unlikely to pick an erroneous route instead of a legitimate one, the condition can always be tested.

Multiple fingerprinting techniques are used, both network based and end-host based. Network based fingerprints include firewall policies, bandwidth information, characteristics of routers, ... End-host fingerprints include OS, IP identifier probing, TCP/ICMP timestamp probing, uptime... It is essential to select multiple discriminative properties to ensure that the hijacker cannot fake the responses.

Detection method

To detect ownership attacks, the system looks for MOASes. For each prefix in a MOAS conflict, the method then builds an AS path tree rooted at the prefix. Then it tries to find a live host to use as probing target. From multiple probing locations (selected so that packets traverse every possible AS to the destination), fingerprints are performed. Then the results are analysed.

To detect intermediate hop attacks, the authors use an AS-level traceroute to detect fake edges in the path. They limit the amount of false positives using a couple of heuristics: popularity constraints (if a link between two ASes is only used by a few prefixes, it is more suspicious than a route used by a lot of prefixes), geographic constraints (a network edge corresponding to two geographically distant points is suspicious), and relationships constraints (based on AS relationships, partially based on [Gao 2001]).

When a potential subnet attack is detected (i.e. incoming update message announcing a subnet), the networks with a provider-customer relationship (based on [Gao 2001]) are removed. This is based on the supposition that a provider has no reason to hijack the traffic of one of its customers, and that a customer cannot steal the traffic of its provider. For the remaining routes, a *reflect-scan* is used for fingerprinting. The reflect-scan is similar to the TCP idlescan technique. An additional step to perform the reflect scan is to find a live host that is not inside the attacked subnet (but in the non-hijacked part of the announcement) to perform the test with.

Limitations

The result of fingerprints are very dependent on the installed OS on the machine (including TCP/IP stack responses). Also, it is not always possible to find a live host to perform those tests, let alone two hosts (in the case of reflect-scans). Moreover, devices on the way (e.g. firewalls) can render the quest for probe-able hosts long and fruitless.

The AS-level traceroute can be blinded via a tool such as Fakeroute [McArthur *et al.* 2009].

IP anycast (extensively employed by root DNS servers) is recognized as a hijack.

2.4.2.3 Using Idle Scan

Detecting BGP hijacking attacks through idle scan is presented in [Hong *et al.* 2009]. This technique aims to avoid relying on multiple vantage points, which adds complexity to properly detect an attack since a single vantage point cannot use multiple routes to target in order to probe the end networks.

Detection method
The system watches BGP update messages and, anytime it detects a MOAS conflict, starts an idle scan to find out if the MOAS is legitimate or the result of an attack.

The probing technique is similar to fingerprinting's reflect-scan (Section 2.4.2.2). But instead of using a machine outside of the hijacked subnet (but still inside the original network) to do the test, it makes use of a host part of the legitimate last-hop network.

Limitations
The limitations are the same as the ones presented for the fingerprinting technique (Section 2.4.2.2).

2.4.2.4 Using PING Tests

The method is presented in [Tahara *et al.* 2008] and focuses on the sole use of PING tests to differentiate a legitimate MOAS and an ownership attack.

Detection method
When a monitor receives a suspicious update containing a prefix to be observed, that monitor executes PING tests for every host address of that prefix. At the same time, it notifies another monitor that has not received the update yet to perform the same test on the original route. The two PING results are compared. If the results are similar enough, there is no hijacking.

Limitations
A preliminary experiment has shown good results, but a large-scale test remains to be done. However, PINGing a whole network range may result in substantial network load, although the authors suggest that for larger networks, only a set of distributed subnets need to be checked.

2.4.2.5 iSPY

[Zhang *et al.* 2008] presents a method for detecting prefix hijacking by analyzing at the result of what amounts to an AS-level traceroute. The method relies on the ability of the victim AS to find its vPath. A *vPath* is the set of AS-level forward paths from the network to the other ASes on the Internet[5]. It can easily be obtained from tools such as traceroute.

Detection method
Considering two forward paths P and P' to destination d, obtained at time t and t' ($t < t'$), if $P = P'$, then everything is fine. If $P' \neq P$ but P' is *complete* (i.e. traceroute receives every response to destination), the route change was legitimate. If P' is incomplete and $P' \subset P$ (i.e. every AS number in P' is in P, up until P' receives no more data), then a *cut* exists between the last router of P' and the next one in P. Finally, P' is incomplete and $P' \not\subset P$, there is a *cut* between the last hop in P' and the (unknown) next one.

Defining Ω as the set of all existing cuts, the cardinal $|\Omega|$ of Ω is the detection signature: if it is bigger than a threshold value, there is hijacking.

[5] Actually, only to the transit ASes of the Internet (i.e. without stub-ASes).

Limitation
First and most importantly, the detection scheme works only if the hijacker blackholes the traffic.

iSPY is likely to confuse a stealthy attack as a legitimate cut link, and can be fooled by a tool such as Fakeroute [McArthur *et al.* 2009].

2.4.3 Combining the Control Plane and the Data Plane

In this Section, we present the main techniques that aim at detecting prefix hijacking by combining routing information extracted from the control plane, and forwarding information extracted from the data plane.

2.4.3.1 The Next Hop Anomaly

This method is presented in [Ballani *et al.* 2007] and was designed under the assumption of a concurrent ownership or a forged AS link attack where the hijacker is located at the first hop after the legitimate AS. (The method would work for any intermediate path level attack, but was limited to this case to reduce the problem to a manageable size.)

Detection method
Let p be a prefix originated by AS O. A router belonging to AS S receives in its update an AS path field containing $N_1, \ldots, N_j, O$. Based on this AS path, any packet to p should be directly forwarded to O once it reaches N_j. The authors define a **next hop anomaly** as a data-plane trace where AS N_j forwards packets for p to some AS I ($I \neq N_j$): it suggests that N_j and O are not interconnected. The next-hop anomaly is used as the detection signature.

Limitations
As is, the signature generates a lot of false positive that the authors attribute to errors in IP to AS mappings, including IXPs (Internet eXchange Points) routers (usually having an IP belonging to the IXP's address space) not included in the AS path, sibling ASes that share address space and have routing agreements, and provider address spaces where a customer uses a small part of its ISP's address as their own. Another source of false positive is traffic engineering.

Even after having removed events attributed to those reasons, the authors are unable to decide whether the remaining cases are the result of prefix hijacking or traffic engineering agreements. Basically, "there is no way to verify the data-plane adjacency of two ASes as claimed by the corresponding control-plane advertisements" [Ballani *et al.* 2007].

2.4.3.2 Argus

Argus is the latest, and therefore arguably the most advanced, prefix hijacking detection system. It combines several of the techniques reviewed previously. The architecture and goals of the system were presented in [Xiang *et al.* 2011], then the architecture was refined and an in-depth analysis of the system capabilities was given at [Shi *et al.* 2012]. Argus was designed as a service free of use, available at [Argus].

Detection method

The fundamental idea behind Argus is that when a hijacking occurs, a part of the Internet is able to reach the legitimate network (either because the whole network has not yet converged, or because the hijack is local), and that the hijacking network and the legitimate network are different. The architecture of Argus, which is divided in three separate modules, can be summarized as follows. The **Anomaly Monitoring Module** (AMM) monitors BGP update messages. When a suspicious BGP message has been detected, the **Hijack Identification Module** (HIM) is notified. By using information from the **Live-IP Retrieving Module** (LRM), Argus *probes* the suspect network using PING to find if the suspicious BGP update is due to a hijack or to a benign routing event.

The AMM checks for three kinds of anomalies. The first one is an *origin anomaly*, i.e. Argus checks if prefix p has a new origin AS. The second one is the *adjacency anomaly*. Argus maintains a database which contains all existing pairs and triples extracted from all AS paths. An adjacency anomaly is raised when an AS pair or triple appears for the first time, which indicates that the topology of the network has considerably changed. The last one is the *policy anomaly*, which is Argus's implementation of [Gao 2001]'s valley-free property (detailed in Section 2.4.1.3).

The goal of the LRM is to collect a number of live IP addresses that can be probed by the HIM once a suspicious routing event for prefix p is detected. It does this by selecting a live IP address that is included in p, *and* not covered by any sub-prefix of p. Live IP addresses are collected from [CAIDA Ark] and [iPlane] traceroute measurements.

When an anomaly is detected, the HIM is activated. The HIM actively probes the network, by using a large set (which the authors define as over 40) public route-servers and looking glasses. By connecting to these vantage points (VP), the HIM actively probes the target network using these two actions during 120 consecutive seconds:

1. check if the router is affected by the anomaly by looking at the BGP routing table,
2. send a PING probe to an IP address retrieved from the LRM.

The results of this measurement are arranged as two vectors C and D, whose elements can be defined as follows:

$$C_{ij} = \begin{cases} 0 & \text{if VP}_j \text{ is affected by the anomaly at time } t = i \\ 1 & \text{otherwise} \end{cases}$$

$$D_{ij} = \begin{cases} 0 & \text{if probing is successful from VP}_j \text{ at time } t = i \\ 1 & \text{otherwise} \end{cases}$$

Then, the *fingerprint* F_t at time t of the anomaly is computed as the correlation coefficient between $C(t)$ and $D(t)$, which is formalized as follows:

$$F_t = \frac{\sum_{j=1}^{m} (c_{tj} - E(C_t))\ (d_{tj} - E(D_t))}{\sqrt{\sum_{j=1}^{m} (c_{tj} - E(C_t))^2 \times \sum_{j=1}^{m} (d_{tj} - E(D_t))^2}}$$

where $E(\cdot)$ returns the average of the vector, i.e.

$$E(C_t) = \sum_{j=1}^{m} \frac{c_{tj}}{m}$$

and

$$E(D_t) = \sum_{j=1}^{m} \frac{d_{tj}}{m}.$$

Argus's classification of an event depends on the final value of the fingerprint F_t. Figure 2.6 illustrates the verdict attached to the anomaly depending on F_t. If there is a strong positive correlation between the vectors, it means that the polluted vantage points cannot PING the IP address while the unpolluted ones can, the anomaly is considered a hijack. If there is a strong negative correlation between the vectors, it means that the polluted vantage points can PING the IP address while the unpolluted onces cannot, the anomaly is a route change. If there is a weak correlation, with strong reachability (i.e. most of the PINGs were successful), the anomaly is assumed to be due to multi-homing, anycast, traffic engineering, i.e. a benign BGP engineering practice. If there is a weak correlation, with a weak reachability (i.e. most of the PINGs were not successful, the anomaly is mostly due to a link failure. A correlation is strong if $|F_t| \geq 0.6$, which has been empirically defined in [Shi *et al.* 2012].

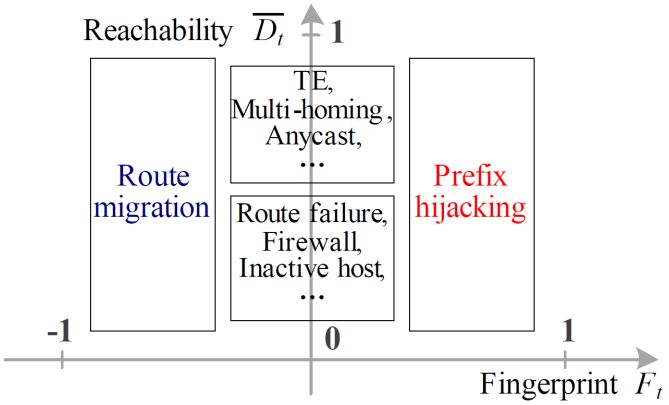


Figure 2.6: Attribution of the cause of an anomaly according to Argus's fingerprint (source [Shi *et al.* 2012])

Limitations

The signatures used by the AMM cover MOASes (called origin anomalies in Argus) extensively. But they do not cover more specific prefixes, the dormant space, or the blackspace. Moreover, even though the use of pairs and triples to check for adjacency sound like a decent idea, no study was done on this topic to verify its accuracy. Also, it is unclear how Argus deals with adjacency in the case of set segments included in the AS path. The policy anomaly clearly indicates that Argus is more targeted at network operators who need to monitor their own network, as policy anomalies do not result in outages.

The HIM uses PING tests in order to probe the legitimate network and the hijacking network. Even though the LRM appears to be designed to work hard to find live IP addresses, there is no indication on how long an IP stays live. Moreover, Argus somehow reaches a conclusion if it does not obtain any PING reply from either network (a link failure, according to Figure 2.6). All in all, the choice of targeted PING tests appears to be a weak choice, probably motivated by leveraging the existing public infrastructure in order to carry out the measurements.

2.4.4 Discussion

The methods described in this Section have all been proposed in order to detect prefix hijacking. They attempt to detect prefix hijacking based on the information extracted from the BGP control plane, i.e. the routing information exchanged between BGP routers, or based on the data plane, which is the way the IP packets are forwarded to destination. The last-generation prefix hijacking detection tool, called Argus (Section 2.4.3.2), which uses both the control plane and the data plane in order to detect suspicious routing events.

These techniques all share the fact that they have been designed as tools to alert network owners that something peculiar is happening with their networks. Generally, the techniques that are based on the control plane, presented in Section 2.4.1, provide a (free or commercial) subscription-based service. Once an account has been created in the system, the monitoring scheme sends alerts, most often via email, to the network operator to inform them that something unexpected is currently going on. Depending on how sophisticated the system is, the network owner can provide various information about his network (e.g. upstream ASes), and customize the alerts that they are interested in, so as to limit the number of alerts that are raised by the system. Techniques that are based on the data plane, presented in Section 2.4.2 generally need to be deployed (and sometimes implemented) by the network owners themselves. Because they are setup to specifically monitor a target network, they are not suitable for a large-scale monitoring of the whole routable address space. A similar argument can be carried out with Argus. The [Argus] service provides an extensive API that network owners can use to take advantage of the detection mechanisms. Moreover, the inclusion of a technique in order to detect violations of the valley-free property by networks is a clear indicator that Argus wishes to interest network operators in monitoring their BGP announcements, not in monitoring prefix hijacking in itself.

Consequently, these tools *report* routing changes, which may be resulting from hijacks, but, are more likely not. For example, the authors of [Xiang *et al.* 2011; Shi *et al.* 2012] fail to build a convincing case for any of the supposed hijack cases that they have analyzed with Argus. If we take into consideration the fact that these tools are designed in order to report suspicious routing changes to network operators, this behaviour is not a problem. Network operators are notified of alerts only if it is related to their own network, which implies that the number of alerts is already quite low. If the alert is a false positive, they can quickly dismiss it, and maybe update the profile they created when they subscribed to the monitoring service so that the alert is not repeated. As a result, these tools are suitable to provide targeted reports on routing changes (which may be the result of a prefix hijacking attack) to network owners, but they are unsuitable to study prefix hijacking.

In Chapter 1, we mentioned that the goal pursued in this Dissertation is to study prefix hijacking as a phenomenon. At the same time, we have established that the current state-of-the-art techniques are unable to provide us with a clear view of this phenomenon because they were not designed to highlight hijacking events, but to report any sort of suspicious event. In other words, the current detection techniques do not attempt to look beyond the signature that they used to detect an event. Basically, if an event matches a signature, it must be suspect. With this behaviour, the global list of generated alerts is very large, and populated mostly with benign events that result from a standard use of the routing practices. Consequently, these techniques are unfit for our study. In Section 1.3, we detailed how we planned on proceeding in order to distinguish between the many false positives, and the suspicious events that need to be investigated.

2.5 Accessing BGP Data

In Section 2.4, we introduced a number of techniques that monitor the BGP control plane, particularly in Section 2.4.1, in order to locate suspicious routing events. These techniques are not meant to be implemented and executed on a router. However, they need access to BGP data in order to work. Over the years, a variety of tools permitting the observation of BGP routing tables have been developed, originally in order to assess the visibility of a prefix from diverse locations on the Internet. In this Section, we introduce the popular choices.

2.5.1 Looking Glasses

A **looking glass** is a network, somewhere on the Internet, that is "kind enough to show you their BGP routing table" [van Beijnum 2002]. A number of ASes provide access to their routers, either through a direct (limited) account accessible via telnet (e.g. [RouteViews]), or through a web interface that enables a visitor to execute a limited set of commands on the router[6] (e.g. [PCH LG]). These public looking glasses are part of the network of vantage point that Argus uses to send PING probes (see Section 2.4.3.2).

2.5.2 Raw Data

Over the years, a number of projects have setup BGP **collector routers**, i.e. BGP routers that dump and archive the BGP data. The first one was the RouteViews project [RouteViews], which run by the University of Oregon. It is a composed of a set of 15 routers on AS6447 placed at several different locations and peering with different backbones. Every two hours, the content of the each collector's RIB is dumped into a file and made available from RouteViews' website. Additionally, every 15 minutes, the collector router generates an archive of all BGP messages exchanged with its peers.

Two other major route collector projects were carried out. One by RIPE NCC, called [RIPE RIS], composed of 13 routers on AS12654, which dump the entire BGP RIB every 8 hours, and provide an archive of exchanged BGP messages every 5 minutes. Another similar project is run by Packet Clearing House (PCH) [PCH].

Data published by route collectors is public and free of use. Archived data is available since 1997 for [RouteViews], since 1999 for [RIPE RIS]. PCH setup its raw data project in 2010, but does not appear to (publicly) provide older archives. While the collector router infrastructure is quite comprehensive and has generated a huge amount of data, access to new data is not possible real-time, and usually suffers from a delay of one to two hours before it is made available. Most of the techniques presented in Section 2.4.1 rely on such data to implement and evaluate their systems.

2.5.2.1 Comparison and Incompleteness

An interesting comparison of the data collected by these three BGP collector infrastructure is available in [Gregori *et al.* 2012]. The authors used the data collected in February 2012 by 10 RouteViews

[6] A discussion on the security weaknesses of looking glasses web software is available in [Bruno *et al.* 2014]. Although the authors are too alarmist, the study certainly provides quality food-for-thought when considering the security of interdomain routing in its globality.

collectors, 13 RIPE RIS collectors, and 51 PCH collectors. First, they classify neighbours of collector routers, which they call *BGP feeders*, into three distinct categories, according to the amount of IPv4 space they announce:

- *minor* feeders, which announce less than the equivalent of a /8 prefix,
- *full* feeders, which announce almost the full assigned IPv4 space,
- *partial* feeders, which are in between.

Table 2.1 shows that for RouteViews and RIPE RIS, around one quarter of neighbours feed a full BGP table; and that it is very low of PCH. (In the case of PCH, only one single peer over the whole route collection infrastructure was feeding one collector with a full BGP table). Full feeders are transit ISPs (mostly tier-1 ISPs), and have a large number of BGP connections.

	Full feeders	**Partial feeders**	**Minor feeders**
RouteViews	25.71%	3.39%	70.90%
RIPE RIS	24.52%	11.72%	63.76%
PCH	0.03%	5.84%	94.12%

Table 2.1: Classification of BGP feeders from each route collector project

Since the majority of the routes learnt by the route collectors are from full feeders, the Internet, as seen from the collectors is biased towards the core of the Internet. In other words, it is equivalent to the Internet as seen by the largest networks in the world. Consequently, by using the data generated by route collectors, a large number of AS-level connections cannot be uncovered. In Section 2.4.1.3, we saw that routes exchanged on peer-to-peer links are not forwarded to upstream providers. Therefore, peer-to-peer connections that are located close to the edge of the network are unlikely to be visible from a route collector's point of view. Even though the values included in Table 2.1 suggest that PCH's collector's feeders might be located closer to the edge (since most of their feeders are minor feeders), in reality, the number of additional AS links that PCH collectors uncover compared to RouteView's and RIPE RIS' are on the order of 2%, which is extremely low. This is because PCH establishes peer-to-peer connections with its feeders, as opposed to RouteViews and RIPE RIS who ask them to give the full routing information. Therefore, the AS connections included in PCH's raw data is also mostly composed of provider-to-customer and customer-to-provider AS links.

Finally, [Gregori *et al.* 2012] considered the geographical location of collector routers, and found that 91% of full feeders are located in Europe or in North America. Africa does not host a single full feeder, even though one RouteViews collector, and four PCH collectors are located in that continent. This implies that the view on the African Internet is obtained with views located in other regions, and therefore that the information is missing some important characteristics. A similar problem is observed in Latin America, where the number of full feeders is very low.

In order to overcome the incompleteness posed by collector routers, [Chen *et al.* 2009] propose to supplement raw data with traceroute measurements. By adding an opt-in module to a popular BitTorrent client, they send traceroute probes towards peers that the client connects to during its regular operations. Because BitTorrent client are located at the edge of the Internet, and because they are so numerous, the topology view from these traceroutes is very different than the one of collector routers. By leveraging this method on over 600k distinct machines in 3,700 distinct ASes,

[Chen *et al.* 2009] are able to identify over 40% of missing peering AS relationships, and around 13% of missing customer-to-provider relationships. However, traceroute measurements provide an IP view of the Internet. Converting the IP topology to an AS representation is a difficult work. While [Chen *et al.* 2009] take extreme caution when converting the data, IP-to-ASN mappings are known to be unreliable. Nevertheless, the authors provide an interesting analysis of the reason why the links they uncover are missing from route collector data. The main reasons are: route aggregation, default routes, and the over reliance on the valley-free property, as detailed in the next Section.

2.5.2.2 Limitations

[Roughan *et al.* 2011] describe several weaknesses in BGP data that are inherent to the way route collectors work. This section summarizes these comments, focusing on those that are relevant to prefix hijacking.

In Section 2.1, we saw that BGP is an information-hiding protocol per design: networks exchange routing information without revealing anything about their own internal structure. As a result, information such as the size of the network (e.g. the number of border points with neighbouring ASes) is unknown. The peering policy with these neighbours also remains hidden. This leads to the abstraction that an AS is an atomic node, which over simplifies the notion of ASes: it lacks policy diversity and multi-connectivity between multiple ASes. This multi-connectivity is important because large ASes – spanning countries, sometimes even continents – may not have the same view of the Internet depending on the location of a machine within that network. In general, the belief that the Internet can be efficiently and correctly modeled into a digraph leads to a global over abstraction of the network.

Moreover, in order to remain scalable over the whole Internet, only the best selected path is propagated. As a result, the data lacks route diversity. Since BGP is a policy-based routing protocol, the forwarded selected path is not necessarily the best one in terms of topology metric (such as IP hops distance). Moreover, many paths appear not to be forwarded far from the network edges. Back up links, for example, seem to appear in the wild only when there is a major issue at the network edge. As a result, normality models often used by prefix hijacking detection techniques are unable to react to this kind of event in an appropriate manner.

Route collectors peer with a large number of distinct peers, and are geographically diverse. However, as previously mentioned, a large AS may not have the same view of the Internet depending on its peering point. As a result, the reported forwarded routes by this kind of peer is really dependant on the collector location. Route collectors are geographically diverse in order to ensure better reach within the network. However, they show a heavy bias toward core networks because they are often located within IPXs and/or peering with backbone networks.

Finally, routing security researchers often simplify the model of BGP business relationships by classifying peering agreements into either provider-customer, peer-to-peer, siblings, coupling these peering agreements with an expectation of valley-free paths [Gao 2001]. However, valleys appear to be more of a rule than an exception in BGP routes because the peering agreements cannot be as neatly classified as the previous three relationships. As a result, when checking against a peering policy-based model, either a wrong policy is inferred by the algorithm, or an alert is raised due to a policy violation. In both cases, it is the result of an over-simplification of the reality which may lead to serious error of judgements when reviewing candidate hijack events.

2.5.3 BGPmon: BGP Monitoring System

BGPmon, not to be confused with the hijacking detection service [BGPmon.net] which was introduced in Section 2.4.1.4, aims to give real-time access to BGP data, avoiding update lags inherent to collectors-based systems [BGPmon]. Unlike collectors, BGPmon doesn't implement a full-blown BGP client, but only the requested functions: receive and log routes. As a result, BGPmon is lightweight enough to peer with more neighbours [Yan *et al.* 2009].

The architecture used by BGPmon is the publish/subscribe one. A set of brokers form an overlay network that peer with BGP routers and exchange information among themselves. They manage the final stream and compute the best route from the publisher to the subscriber. Subscribers (applications) can personalize the information they want to receive in their stream (including open, close, update, notification BGP messages, state changes in BGP, break up and tear down of peering connections, . . .).

Currently, the BGPmon application incorporates the three facets of the system: broker, publish, and subscribe. It is divided in three levels. The first one peers with a BGP enabled device and places BGP messages in a queue, creating a stream of events. The second one labels events from that queue that identifies announcements, withdrawals, updates, duplicates, . . . As this second stage can be quite costly memory-wise, it can be disabled. Disabling it, however, results in losing the ability to simulate a route refresh without the help of remote sources. The final stage adds status information and injects route table snapshots into the stream.

A planned companion to BGPmon is BGPbroker that would enable the separation of the peering servers (that connect to neighbour routers) from the client servers (that send event streams to clients).

The quality of data available from BGPmon is quite similar to the one available from collector routers. Actually, BGPmon peers with a lot of these collectors. Its main advantage is to provide real-time access to data.

2.6 The Internet Routing Registries

The idea behind **Routing Registries** is to constitute a database containing a high-level description of routing policies of Autonomous Systems. The idea dates back to NSFNET (see Chapter 1) in which submitting the information to the *Policy Routing Database* was a requirement [Blunk 2011]. When RIPE NCC was formed, in 1989, it created a registry for allocation of IP addresses and ASNs. Little by little, the registry was expanded, first by RIPE, and then by the IETF; and, eventually, the concept was standardized in [RFC2622], which defines the **Routing Policy Language Specification** (RPSL). The RPSL specifies a number of objects that can be used to detail

- which prefixes an AS announces,
- who the neighbouring ASes are,
- which prefixes are announced to which neighbouring AS,
- which prefixes are allowed to be received from which neighbouring AS.

To illustrate the basics of RPSL, let's use the RIPE database as an example. [RIPEDB Docs] details all available objects in the RIPE database as well as their usage, attributes, and possible relations. A subset of these objects are:

- The `mntner` objects models access rights and authorize creation, deletion, or modification of other objects. Objects that are linked to a `mntner` can only be modified by the owner of the authentication credentials behind the corresponding maintainer account. This link has now been made mandatory for all objects.
- The `organisation` objects provide information about companies, non-profit groups, or individuals who hold an Internet resource. They mainly contain business contact information such as post address, emails, phone numbers, etc. They are maintained directly by the RIPE NCC, and linking another object with an `organisation` required an explicit authorization from this organization.
- The `inetnum` objects represent assigned IPv4 space in the RIPE region. These objects are either maintained directly by RIPE, or by another maintainer, depending on who assigned the prefix. For example, the `inetnum` corresponding to a prefix assigned to an ISP by RIPE will be maintained by RIPE. A sub-allocation made by this ISP to one of its customer will have an `inetnum` whose maintainer is the ISP.
- The `aut-num` object has a dual purpose. The first one is to detail an ASN allocation in the RIPE region; just like `inetnum`, it is maintained by RIPE. The second purpose of the `aut-num` object is to specify routing policies. By using the attributes `export` and `import`, an AS can specify which (set of) prefixes is to be either announced, or accepted as announcements to/from the specified AS.
- The `route` object is used to link an `aut-num` object with a `inetnum` object, thereby binding an ASN with a prefix. For such a bind to exist, valid credentials for the `mntner` object of both the IP block and the ASN are required.

Several tools have been built to automatically generate filters for BGP routers based on information included in the IRR [Blunk 2011].

Even though publishing routing policies in the **Internet Routing Registries** (IRRs) has been considered as good practice for a long time [van Beijnum 2002; Butler *et al.* 2010], the accuracy of IRRs is widely debated among the community. For example, [Siganos *et al.* 2004] underlines the inconsistencies among the different versions of the database, as well as the varying level of accuracy depending on the considered database and object. However, a bit later, the same authors developed a system that uses information included in the IRRs to validate the origin AS of IP prefixes in [Siganos *et al.* 2007]. More recently, [Khan *et al.* 2013] carried out a similar experiment and compared the origin AS inside the IRR with the one in BGP for the whole routing table. They found that 82% of prefixes had a match in the IRR, and that for only 4% of them, the origin AS in the IRR and in BGP was different. Moreover, 78% of ASes appear to register all of their BGP announcements in the IRR, 11% register a subset of their announcements, and the rest does not use the IRR at all.

A possible explanation for the apparently increasing quality of data included in the IRRs is that some RIRs now make publishing correct information in the IRRs mandatory. For example, the RIPE Standard Service Agreement states that an ISP should actively maintain its sub-allocations in the IRR. Failure to do so may lead to having its rights to sub-allocate Internet resources revoked. Moreover, RIPE itself creates the `inetnum` objects relevant to each allocation it makes. The `mntner` linked to these objects are owned by RIPE themselves, making it impossible for ISPs to remove such

an object from the database. In Chapter 4, we will show that a lot of people actively maintain IRRs, and argue why the included information cannot be entirely dismissed.

2.7 Summary and Conclusion

This Chapter formalized a lot of material. First we introduced the internal specifications of the BGP protocol, the de-facto standard used for interdomain routing. We detailed the way autonomous systems exchange IP prefixes in order to advertise themselves to the world. We also detailed the way BGP chooses its favourite route, and explained that each AS can individually decide on a routing policy that influences the way in which the favourite route is selected. Then, we showed that BGP, by design, believes everything that it is being told by its neighbours, a concept known as mutual trust.

Then, we showed how this mutual trust can be abused and lead to prefix hijacking attacks. We introduced a number of ways in which prefix hijacking can be done by providing a taxonomy of prefix hijackings and illustrating six distinct vectors of attack: concurrent ownership (otherwise known as MOAS attacks), subnet attacks, forged AS links attacks, man in the middle attacks, and abuses of the dormant space or blackspace. We also detailed a number of real-world cases to illustrate that these events do happen both, accidentally and intentionally, and have disastrous consequences for the victim networks.

We then presented a number of proposals to secure BGP and render prefix hijacking impossible. These techniques can be divided into two distinct groups. The first one are techniques that modify the BGP protocol in order to secure the exchanged routing information. These technique make use of public key infrastructures in order to sign and verify the routing information. We showed that, while the first proposal solved every problem in theory, it is impractical to deploy for two main reasons: (i) in order to secure the whole routing infrastructure, every router must switch to the new protocol; (ii) the induced overhead on the current routing infrastructure is too high. As a result, later proposals aimed at finding a trade off between additional security and additional work on routers. The second group of proposals to secure BGP relies on additional protocols which can be introduced alongside BGP as a side-channel to enable the validation of routing information. We detailed RPKI, which is the architecture currently actively supported by RIRs and router vendors. However, we also showed that, currently, the IP space secured by RPKI is well below 10%.

Even if RPKI were completely deployed and enforced in the close-to-midterm future, it is not able to prevent the more sneaky kinds of attacks that were presented in Section 2.2. Consequently, reliable prefix hijacking detection tools should be used in order to highlight prefix hijacking occurrences. By knowing when and where a hijack occurs, a mitigation process can be started promptly and effectively. We detailed a number of techniques that are either based on the BGP control plane, i.e. the information contained in a BGP router's routing information base, or on the data plane, i.e. the way IP packets are forwarded throughout the Internet. We explained why these tools are, in a way, advanced routing changes notification mechanisms; and are inapt to be used as a starting point to study prefix hijacking as a phenomenon, which is a goal we set in Chapter 1.

Finally, we introduced the BGP data collection infrastructure, and compared the three main route collector projects. We also introduced the Internet Routing Registries and discussed about the quality of the included information.

Multiple Origin AS Prefixes

3

In the previous chapters, we saw that BGP does not come with any defense mechanism against prefix hijacking attacks, and that, in the short term, there is no possibility that BGP will be updated to do so. If prefix hijacking cannot be avoided by design, it should at least be detectable.

Chapter 2 presented a series of techniques that try to detect prefix hijacking. However, these tools, even last-generation, yield outputs cluttered with alerts corresponding to benign network events. For the targeted audience of such tools (ISPs, prefix owners, ...), this is not a problem since they know the expected behaviour of their prefix. With this ground-truth, they can make an informed decision on the value of the alert, and take appropriate action, if needed. Moreover, since they only monitor their own prefixes, the number of alerts they receive is low enough so as not to be considered as over information.

In this Chapter, we focus on Multiple Origin AS prefixes, referred to as MOAS prefixes, which are IP prefixes originated from multiple autonomous systems. We provide a detailed analysis of the regular network practices that lead to such a topology, and we provide a technique for removing a significant fraction of these false positives from prefix hijacking alerts. We build a system to investigate MOAS cases, and provide a real-world case study of a suspicious MOAS event.

3.1 Introduction

In Chapter 2, we detailed how the Internet is composed of a set of interconnected independent **ASes** (Autonomous Systems), that are glued together using **BGP** to exchange reachable IP prefixes. Sometimes, an IP prefix p is simultaneously announced from multiple ASes, resulting in a so-called *Multiple-Origin AS prefix*. In other words, a MOAS event, MOAS prefix, or simply **MOAS** (Multiple-Origin AS) is the result of an IP prefix p being announced simultaneously from multiple endpoints. Even though [RFC1930] discourages MOAS situations, [Zhao *et al.* 2001] presented a series of legitimate network engineering practices that lead to the announcement of MOAS prefixes. Examples include prefix multihoming (i.e. a prefix using multiple upstreams), and the use of anycast. These use cases are expected to create **long-lived** MOAS events. However, [Zhao *et al.* 2001] also uncovered

a number of **short-lived** MOAS events whose root causes are unclear, and that were, by default, attributed to router misconfigurations. MOAS events can also be the result of a concurrent ownership prefix hijacking attack, where the attacker claims to be the rightful origin of the prefix alongside the regular prefix owner.

In the first part of this Chapter, we provide an in-depth longitudinal study of MOASes with which we clarify and quantify the root causes behind long-lived *and* behind short-lived MOAS events. Our approach is the following one. First, we consider MOAS events individually, and we revisit existing previous work by focusing on MOAS durations: [Zhao *et al.* 2001; Chin 2007]. Then, we provide the first study of MOAS events as groups of events related to their prefixes, not as completely independent events. With this study, we show that short-lived MOASes are less numerous than previously reported because many short-lived MOAS events are actually repeated events related to a small set of prefixes and due to instability or route flapping.

Second, we introduce a taxonomy of MOASes as distinct MOAS patterns – **peering**, **classical**, and **me-too** MOAS – and study their prevalence and temporal characteristics. With these patterns, we show that the majority of MOASes are fake MOASes that are the result of loosely-defined, or outdated routing policies. The traditional MOAS shape only amounts to 30% of all cases.

Third, we also look at the evolution of these findings by relying on the analysis of two full years of measurement collected 10 years apart, in 2002 and 2012 respectively, in order to underline discrepancies that may arise due to changes in standard practices, or due to the global evolution of the Internet.

In the second part of this Chapter, we introduce a dataset of suspicious MOAS by deriving an algorithm from the network knowledge gained out of the longitudinal study of MOAS prefixes. This dataset contains MOASes that are more likely to be the result of a prefix hijacking concurrent ownership attack. We then present a system that we use to correlate these MOASes with various network and security data sources in order to find out if the MOAS was the result of a hijacking attack or not. With a thorough analysis of a real-world case of a suspicious MOAS, which we called *the Bulgarian case*, we illustrate how this kind of correlation needs to be carried out with much care, and how the lack of available ground-truth makes observing routing events from the third-party point of view difficult.

3.2 Related Work

[Zhao *et al.* 2001] pioneered the analysis of MOAS events and analysed BGP data between late 1997 and mid 2001. This analysis concluded that 36% of the MOAS events were one-time events and lasted less than a day, 30% of which were attributed to a single misconfiguration. Excluding those, the average MOAS duration was 30.9 days. For MOASes that lasted over 9 days, the mean duration was 107.5d. These figures are computed using the MOAS duration per event[1].

The authors then discuss a number of reasons for which a prefix would be originated from multiple ASes: prefixes associated with an Internet exchange point (IXP) may be advertised by all the ASes within the IXP, since they are reachable through all of them. Multihoming without BGP (i.e. via static links or some IGP protocol) also leads to MOAS, since the prefixes are then announced by the upstream providers. Multihoming with BGP, but with a private ASN yields the same result.

[1]The formal definition of this metric is available in Section 3.3.1.

Anycasting can also lead to MOAS prefixes. Finally, since prefix aggregation in BGP transforms the AS path into an AS set (in which the order of ASNs is random), some artificial MOAS prefixes can be observed.

[Chin 2007] revisited the work of [Zhao *et al.* 2001] by studying three weeks of data in January 2007, and found an average lifetime of MOAS events to be 13.25 hours. Chin then proposed new reasons behind MOAS prefixes: multinational companies may advertise prefixes from various branches in different countries, and such organizations possibly own multiple AS numbers. Companies may also host their servers in data centers, announcing the prefixes both, from the data center and from their offices. Some countries that use satellite links and simultaneously use different providers have their prefixes announced by these providers, resulting in a MOAS conflict.

Of course, MOAS prefixes can also be the result of a malicious attack against the routing infrastructure, i.e. a concurrent ownership prefix hijacking, as detailed in Chapter 2. One of the first systems devoted to the detection and notification of MOAS was PHAS [Lad *et al.* 2006], which monitored the origin ASes associated to a prefix by building a set of current origins, and raised an alert whenever an AS started announcing the prefix, or whenever an AS withdrew its announcement. Other schemes, such as [Hu *et al.* 2007; Hong *et al.* 2009; Shi *et al.* 2012] try to compare the fingerprints of the networks within each originating AS. If the announcement is legitimate, the networks are the same, and so will the fingerprints. Any discrepancy indicates that the end networks are different, and, consequently, likely the result of a hijacking attack. The fingerprint signature is computed either from actively probing live hosts in the target networks, or by using network diagnostic tools, such as `ping`. In order to fingerprint the two announcements, they either use time series, i.e. comparing the regular network to the new MOAS network; or they try to find a measurement point that has not been affected by the BGP route change yet. Nevertheless, these papers focus on the threat posed by MOASes, and not on their inner characteristics or classification.

3.3 Methodology and Dataset

In this Section, we present the methodology and the dataset that have been used to perform an in-depth, longitudinal study of MOAS prefixes. Section 3.4 presents the result of the application of this methodology to two years of data: 2002 and 2012. In Section 3.5, we will build a dataset of suspicious MOAS cases from the network knowledge we gained with this analysis. This dataset is then used in Section 3.7 to detect and analyze a highly suspicious real-world MOAS case.

3.3.1 Definitions

Formally, a **MOAS prefix** (Multiple-Origin AS) is the result of a prefix p being simultaneously originated from multiple ASes. In other words, at a given point in time, the AS paths for p end by a set $\mathcal{O}(p)$ of multiple origin ASNs, so that $\mathcal{O}(p) = \{a_1, \ldots, a_n\}$. For example, using Figure 3.1, $\mathcal{O}_{]t_0,t_1[}(p) = \{1\}$, $\mathcal{O}_{]t_1,t_2[}(p) = \{1, 2\}$, $\mathcal{O}_{]t_2,t_3[}(p) = \{2\}$, $\mathcal{O}_{]t_3,t_4[}(p) = \varnothing$, and so on. If a prefix p is MOAS prefix, we alternatively say that p is a MOAS. It is important to stress that MOASes only occur for the same prefix p. In particular, any prefix q more specific than p with a different origin than that of p is not defined as a MOAS prefix, but as a MOAS *sub*prefix (alternatively *sub*-MOAS), which will be one of the focus points of Chapter 4.

The literature defines **MOAS event duration** as the duration of a single MOAS event. In Figure 3.1, the durations of the three MOAS events are $t_2 - t_1$, $t_6 - t_5$, and $t_8 - t_7$. MOASes

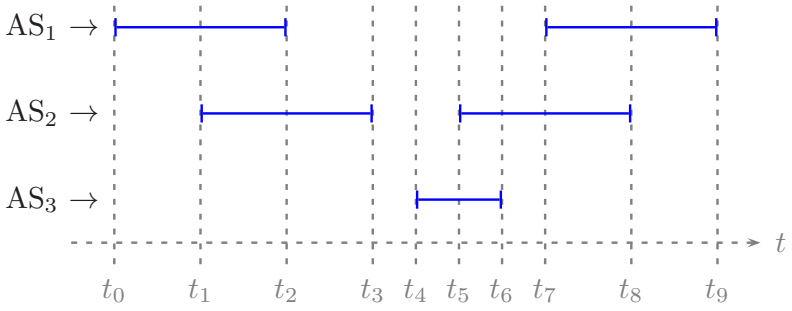


Figure 3.1: Example of announcements for a prefix p

are usually classified according to this metric with the following terminology [Zhao *et al.* 2001]: **short-lived MOAS events** last less than 1 day, while **long-lived MOAS events** last more than 1 day.

If p is not a MOAS, but p is still present in the routing tables, p is a **SOAS** (Single-Origin AS), meaning that p is originated by a single AS. In Figure 3.1, this happens during $]t_0, t_1[$, for example. If p is not included in the routing tables, we will say that p is **down** (Figure 3.1 during $]t_3, t_4[$). This does not imply that traffic destined to p cannot be routed, because a set of covering prefixes could be used to forward the traffic. By contrast, a prefix is **up** whenever it is a SOAS or a MOAS.

We define the **lifetime** of a prefix p as the difference between the timestamp at which the prefix was last withdrawn (that is, the timestamp at which the prefix goes down for the last time) and the timestamp at which the prefix was first announced. The lifetime of p in Figure 3.1 is simply $t_9 - t_0$. On the other hand, the **uptime** of p is defined as the total duration during which the prefix was advertised. In Figure 3.1, the uptime of p is $t_3 - t_0 + t_9 - t_4$.

In Section 3.4.2, we introduce a new metric that we call the **MOAS duration per prefix** which is defined as the sum of the durations of the individual MOAS associated with this prefix. Using Figure 3.1, the MOAS duration for prefix p is $t_2 - t_1 + t_6 - t_5 + t_8 - t_7$.

3.3.2 BGP Dataset

In order to study MOASes, we use data from the [RIPE RIS] route collector located in Amsterdam (`rrc00`), which has above 40 geographically diverse peers. We retrieve the update messages and simulate BGP operations according to [RFC4271], which we detailed in Chapter 2. More precisely, we maintain a routing table for each peer – similar to BGP's Adj-RIB-In – the *adjacent routing table*. Each route announced by a peer is added to that peer's adjacent routing table. Whenever a withdrawal is received for a prefix, every route to that prefix is removed from the peer's adjacent routing table. Since we are not interested in routing traffic, we do not try to select preferred routes. We are, however, interested in knowing if a prefix p is up, i.e. if p is present in any of the adjacent routing tables.

The set of origins $\mathcal{O}(p)$ associated with prefix p is composed of the union of all the origins included in all of the AS paths of each adjacent routing table. If the cardinality of $\mathcal{O}(p)$ is larger than 1, p is a MOAS. For example, in Figure 3.1, during $]t_0, t_1[$, $\mathcal{O}_{]t_0,t_1[} = \{1\}$ whose cardinality is 1, and the prefix is a SOAS. During $]t_1, t_2[$, $\mathcal{O}_{]t_1,t_2[}(p) = \{1, 2\}$ whose cardinality is 2, and the prefix is a MOAS. Finally, during $]t_3, t_4[$, $\mathcal{O}_{]t_3,t_4[}(p) = \varnothing$ whose cardinality is 0, and the prefix is down.

3.4 A Longitudinal Study of MOAS Prefixes

Our study starts by revisiting the previous works [Zhao *et al.* 2001; Chin 2007]: we compare the uptimes of MOAS and SOAS prefixes, and put into perspective the average uptime of MOAS prefixes with the average duration of MOAS events. By doing this, we show that our results are similar to what had been observed by [Zhao *et al.* 2001; Chin 2007], thus ensuring that our further results can be put into perspective with the results provided by both of these studies.

We then consider that MOASes are not only a set of independent events, but they are related to a prefix. By grouping MOAS events per prefix, we are able to uncover inner relationships between multiple successive events. Most notably, we show that a large fraction of short-lived MOAS events are not the result of misconfigurations, which contradicts [Zhao *et al.* 2001].

We look at topology graphs of MOAS prefixes from which we extract MOAS patterns that we use to classify and quantify MOASes. We study the temporal evolution of the topology graphs related to MOAS prefixes by comparing them months before and after MOAS events. This evolution enables us to understand the root causes behind the MOAS patterns.

In Section 3.5, we will build a dataset of suspicious MOAS cases from the network knowledge we gained with these results. This dataset is then used to detect and analyze a highly suspicious real-world MOAS case, in Section 3.7.

3.4.1 General Results

During the year of 2002, almost 310k different prefixes were announced, less than 9% of which presented (at least) one MOAS event. In 2012, there were almost 765k distinct announced prefixes, less than 6% of which were in a MOAS state at some point during the year. These figures suggest that, while both, the number of global prefixes and the number of MOAS prefixes increased in 10 years, their proportion has decreased. In both cases, less than 5% of MOAS prefixes were the result of route aggregations. We removed these prefixes from our MOAS cases before further analysis.

		uptime			lifetime		
		μ	CoV	q_{50}	μ	CoV	q_{50}
MOAS	**2002**	328d	0.25	363d	334d	0.23	364d
	2012	308d	0.34	364d	317d	0.31	364d
SOAS	**2002**	146d	1.11	37d	172d	0.89	146d
	2012	223d	0.72	348d	239d	0.65	360d

Table 3.1: General statistics on BGP data for 2002 and 2012

Table 3.1 shows the mean (μ), coefficient of variation[2] (CoV), and median (q_{50}) durations for the uptime and the lifetime of both MOAS and SOAS prefixes during 2002 and 2012. Mean values for MOAS prefixes in both, 2002 and 2012 are significantly higher than the values for SOAS prefix in terms of uptime and lifetime. This suggests the use of MOAS to improve the connectivity of a prefix. In particular, median uptime and lifetime of MOAS prefixes are both close to 1 year, meaning

[2]The coefficient of variation is equal to the standard deviation divided by the mean, i.e. CoV $= \sigma/\mu$. If its value is lower than 1, the variable is considered to have a *low* variance.

that 50% of those prefixes were seen over the entire observation period. The mean and median value for SOAS prefixes in 2012 – both close to 1 year – are also much higher than those in 2002, where the median uptime of 37d is very low compared to the observation period of 1 year, and to a median lifetime of 146d. These figures for 2002 are in line with the ones presented in [Siganos *et al.* 2002], even though their analysis does not focus on MOASes, which strengthens our confidence in the accuracy of our method. While we only detail 2002 and 2012, we looked at the data of years in between and found similar conclusions.

3.4.1.1 MOAS Events

In this section, we consider MOAS events as a set of distinct events, independent of the prefix with which they are associated. For example, we consider independently the 3 MOAS events depicted in Figure 3.1 during $]t_1, t_2[$, $]t_5, t_6[$, and $]t_7, t_8[$. The **MOAS duration** (per event) is the duration of a single event. In Figure 3.1, the durations of the three MOASes are $t_2 - t_1$, $t_6 - t_5$, and $t_8 - t_7$.

		μ	CoV	q_{50}
All MOAS events	**2002**	33d	2.23	22h
	2012	48d	1.88	26h
Short-lived MOAS events	**2002**	133mn	2.26	9.3mn
	2012	101mn	2.60	3.13mn

Table 3.2: Duration of MOAS events

Figure 3.2 depicts the duration of MOAS events in 2002 and in 2012. MOAS duration information for 2002 and 2012 are available in Table 3.2. The large difference between the mean and the median shows how prevalent short-duration events are. The consequence of the comparison between these values and those presented in Table 3.1 is that MOAS prefixes do not spend their whole life in a MOAS state.

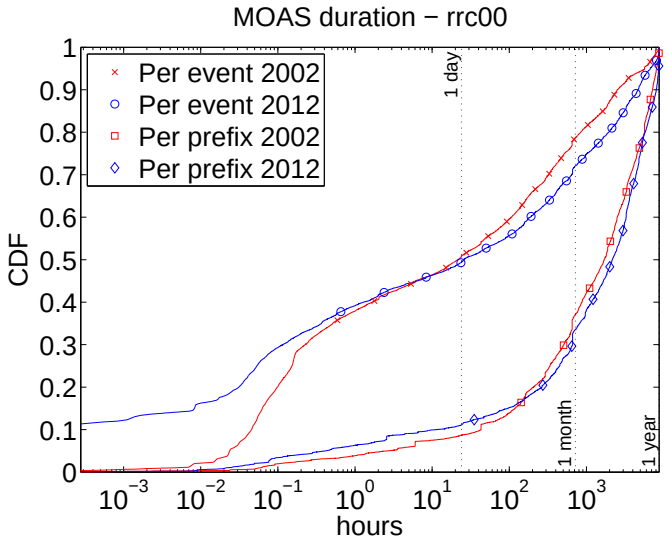


Figure 3.2: MOAS events duration

3.4.1.2 MOAS Prefixes

In this section, we consider MOAS events grouped by the prefix for which they appeared. Distinct MOAS events may appear over the course of the observation period for a single prefix p. We say two MOAS events associated with a prefix p are **distinct** if the origin sets $\mathcal{O}(p)$ are different for the two events. For example, in Figure 3.1, prefix p has 3 MOAS events: during $]t_1, t_2[$, $]t_5, t_6[$, and $]t_7, t_8[$. Moreover, $\mathcal{O}_{]t_1,t_2[}(p) = \mathcal{O}_{]t_7,t_8[}(p) \neq \mathcal{O}_{]t_5,t_6[}(p)$. So, even though Figure 3.1 depicts 3 MOAS events, only 2 of them are distinct in the sense that they involve different ASes. Furthermore, the **duration** of **MOAS events per prefix** is the sum of the durations of the individual MOASes associated with this prefix. Using Figure 3.1, the MOAS duration for prefix p is $t_2 - t_1 + t_6 - t_5 + t_8 - t_7$. In the remainder of this section, unless explicitly stated, duration means the duration of the MOAS events *per prefix*.

Figure 3.2 plots the duration of MOAS events per prefix. Only around 10% of the MOASes are short-lived, which heavily contrasts with the 50% obtained when considering each MOAS event on its own. This implies that certain prefixes must have many MOAS events. This is confirmed by Figure 3.3, where the number of MOAS events and the number of distinct MOAS events per prefix are plotted. The prefixes are sorted by decreasing number of MOAS events. For the first 1000 prefixes with the most MOAS events, the mean and median duration of single MOAS events is very small (in the order of a few minutes or less).

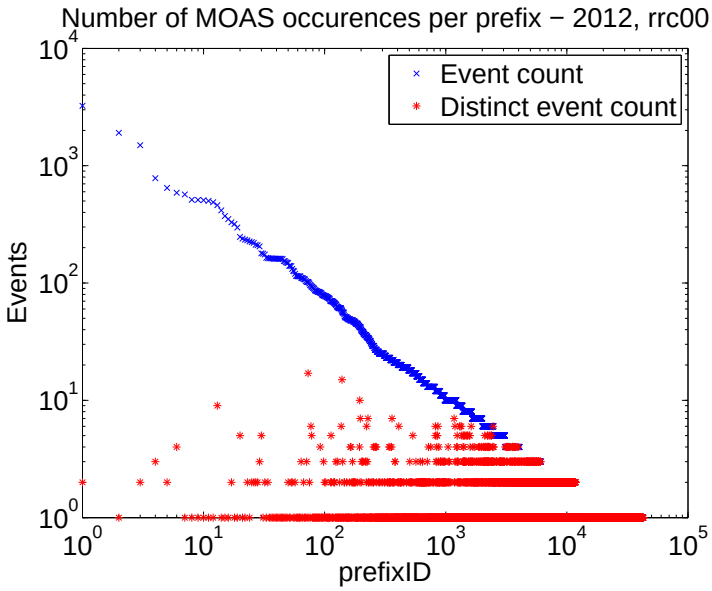


Figure 3.3: Number of MOAS events per prefix in 2012

Figure 3.3 shows that, for approximately 1000 prefixes out of the 43k MOAS prefixes, the number of *distinct* MOASes is significantly lower than the number of MOAS events. Some of these prefixes only have 1 distinct MOAS, but hundreds of MOAS events. In these cases, there was a continuous flipping between SOAS and MOAS announcements. This kind of behaviour can be explained by an instability between the prefix owner and one of its upstreams. However, by looking at the AS paths in the duplicate BGP update messages related to these events, we saw that only one sub-path actually caused this flipping phenomenon to the route collector. For this reason, we suspect that this flipping was not caused by an instability in the prefix owner's connections, but by some router located in an AS between the collector and the prefix origin. The equivalent figure for 2002 looks very much alike, and is depicted in Figure 3.4.

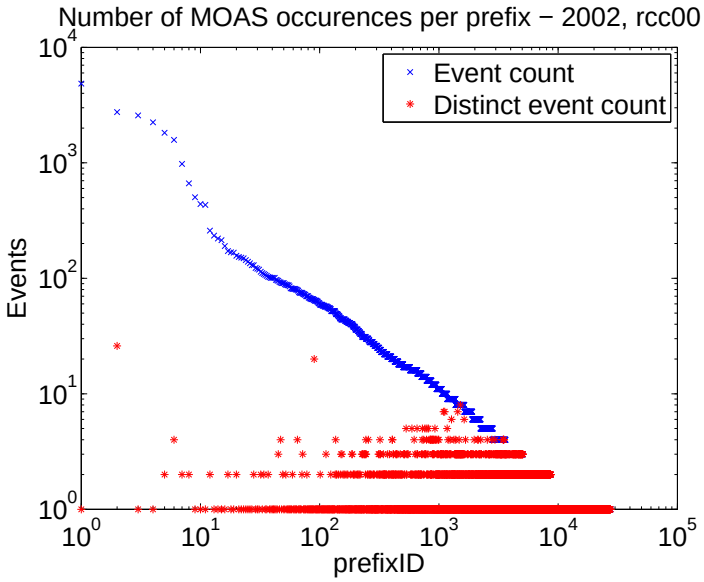


Figure 3.4: Number of MOAS events per prefix 2002

Figure 3.2 also implies that **the bulk of short-lived MOAS events cannot be attributed to misconfigurations**. If, as supposed by [Zhao *et al.* 2001; Chin 2007], most short-lived MOAS events are due to a misconfiguration, there should not be that many recurrent MOAS events for the same prefix. Indeed, only the sum of numerous short-lived events for the same prefixes (Figure 3.3) can result in raising the MOAS duration per prefix as much, compared to the MOAS duration per event. Since misconfigurations are usually sorted out promptly [Mahajan *et al.* 2002], many short-lived events affecting many distinct prefixes would not shift the CDF plot to the right. As a result, the curves in Figure 3.2 would not show such a drastic difference between the "per event" and the "per prefix" computations.

The mean and median MOAS durations per prefix in 2002 and 2012 are detailed in Table 3.3. We clearly see that both the mean and median values for the MOASes per prefix are a lot larger than individual MOAS event durations. This is, once again, the result of the combination of the many short-lived events per prefix.

We also considered the fraction of MOAS uptime for a prefix over its total uptime. One might expect MOAS prefixes to remain in MOAS state during most of their uptime in order to maximize the benefits behind their chosen MOAS configuration. However, as Figure 3.5 shows, the distribution of the fraction of time in MOAS state distribution is uniform and contradicts this expectation. We explain this phenomenon by the use of *transient* MOAS configurations. A temporal analysis of the topological evolution of MOAS networks uncovered multiple cases of stub networks switching between upstream AS providers. This operation can be summarized as follows. Originally, prefix p is announced by ISP A. At some point, the owners of p find it more advantageous to use ISP B. In order to avoid any service disruption, p remains connected to (and announced by) A while things are being set up with B (i.e. connecting p to B), and then also starting announcing it from B. This results in a MOAS. After some time (weeks), p is disconnected from A, and remains exclusively announced and reachable via B. We also uncovered situations where a set of prefixes p_i belonging to different entities all evolved in the same manner, with the same set of transient MOASes. The reason was a topology change at the ISP used by the prefix owners. In this situation, the prefixes were connected via multiple origins ASes for a couple of weeks, and then a part of the topology graph was pruned, leaving the prefixes effectively reachable via a single AS. The difference between

these two situations is that, in the second case, the prefix owners were not the ones who decided to change the way they are connected to the Internet.

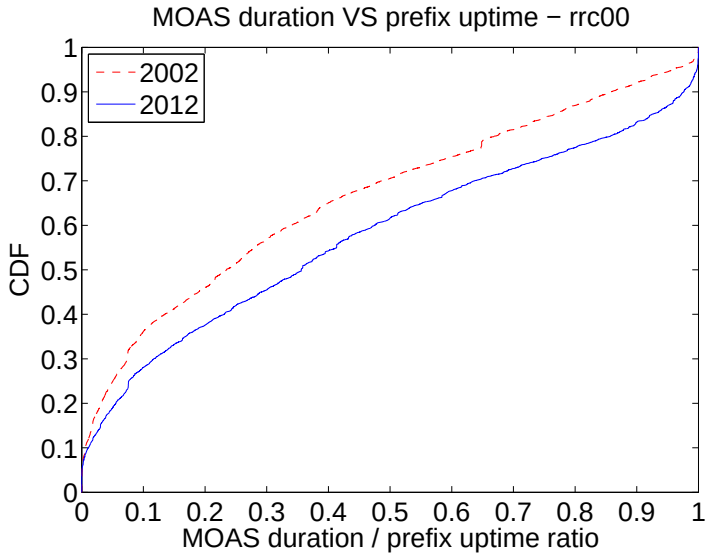


Figure 3.5: Total MOAS duration for the prefix VS prefix uptime

3.4.2 MOAS Patterns

We analyzed the AS-level graph of MOAS prefixes and were able to extract a set of patterns that result from MOAS announcements. This led us to a taxonomy of MOASes that we present now. In order to understand the reasons behind these MOAS events, we looked at the evolution of the AS topology of MOAS networks during 6 months surrounding the MOAS occurrence.

The first pattern, depicted in Figure 3.6, shows a situation where both, the prefix owner and its upstream are announcing the prefix. We call this situation a **Peering MOAS**. Even though Figure 3.6 only depicts one upstream, we saw cases where upstreams of the upstream were also announcing that prefix. The mean and median durations for peering MOASes are presented in

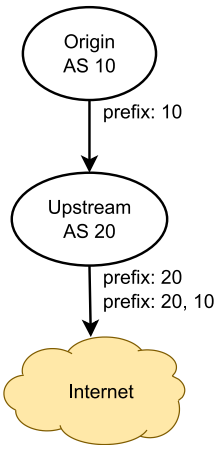


Figure 3.6: Graph of first MOAS pattern: Peering MOAS

Table 3.3. Figure 3.9 shows the distribution of the durations of peering MOASes for 2002 and 2012. 60% of those events last longer than a month, and around 10% of them are short-lived.

		μ	CoV	q_{50}
All MOAS prefixes	**2002**	111d	1.01	71d
	2012	125d	0.95	90d
Peering MOAS pattern	**2002**	103d	1.03	64d
	2012	123d	0.97	88d
Classical MOAS pattern	**2002**	80d	1.24	32d
	2012	101d	1.12	43d
Me-Too MOAS pattern	**2002**	203d	0.64	241d
	2012	181d	0.60	197d

Table 3.3: Duration of MOAS prefixes

We often saw this pattern appearing in the following setting. The prefix is first announced by the upstream, but assigned to the customer (e.g. Figure 3.1, during $]t_0, t_1[$). At some point, the customer decides to handle routing on its own, and acquires its own AS number and starts BGP peering with the upstream. At this point, there is a MOAS (Figure 3.1, during $]t_1, t_2[$). Eventually, the upstream withdraws its announcement of the prefix, leaving only the owner's announcement in the routing tables (Figure 3.1, during $]t_2, t_3[$). In this case, the MOAS was the side-effect of a real topology change. Even though we did not explicitly witness any situation in which this description is not accurate, this pattern is not necessarily the result of a provider-customer relationship. It is conceivable for this pattern to be the result of any direct peering relationship, such as peer-to-peer or siblings networks, as defined by [Gao 2001].

In other peering MOAS cases, the owner has several upstream providers, some of which re-announce the prefix. Effectively, the prefix owner is multihomed; but a subset of its upstreams originate the prefix. We believe that in this situation, the network was originally connected to a single upstream that handled BGP operations on its behalf, but then decided to multihome in order to benefit from increased connectivity. However, the original ISP's configuration remains unchanged and carries on announcing the prefix.

We consider this pattern to create **fake** MOASes because, in both of these situations, there is no gain for the owner from its upstream's announcement. Indeed, if the upstream stopped originating the prefix, the situation would remain unchanged: the prefix would still be reachable via all of its upstreams without loss of connectivity. Table 3.4 shows that peering MOASes amount to around 70% of all MOAS events. We believe this class of MOAS is caused by loosely-defined (or outdated) routing policy. Another cause would be prefixes associated with IXPs, as described by [Zhao *et al.* 2001; Chin 2007], which can be announced by the IXP's AS on top of the owner's AS.

The second pattern, depicted in Figure 3.7, is the expected AS pattern when talking about MOASes. For this reason, we call it the **classical MOAS** pattern. There are multiple distinct AS paths leading to the prefix. The mean duration of these MOASes are shown in the penultimate row of Table 3.3. These values suggest that classical MOASes are longer-lived in 2012 than in 2002. This is confirmed by Figure 3.9 which plots the durations of these events. In 2002, around 50% of them were short-lived, which then decreased to around 35% for 2012.

We found the main reasons behind this pattern to be in accordance with engineering practices described in [Zhao *et al.* 2001; Chin 2007]. In order to verify this, we used WHOIS data for the

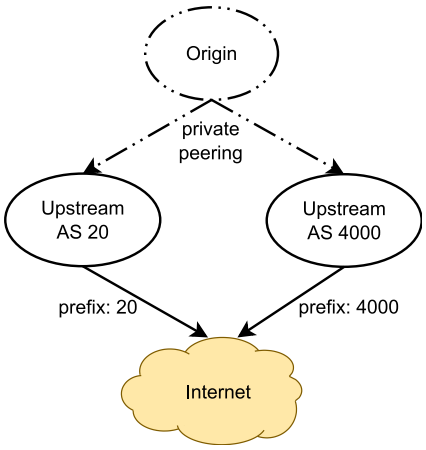


Figure 3.7: Graph of second MOAS pattern: Classical MOAS

prefix and origin ASes. The ASes most often belonged to well-established ISPs, and the prefixes were registered to another entity. We also saw cases where multinational companies were the owner of each of the origin ASes.

Table 3.4 shows the proportion of classical MOASes among all MOASes, which is around 25%. Consequently, the pattern that is traditionally believed to be *the* MOAS configuration only amounts to a quarter of MOAS prefixes.

Any loss of origin in a classical MOAS means a loss of connectivity between the prefix and its upstream. If an origin AS stops announcing the prefix, it will not receive traffic for it. It will therefore not provide any connectivity to the Internet for the owner. This contrasts with the situation of multihoming with a fake/peering MOAS, where the loss of an upstream origin does not affect the connectivity of the network, since the upstream AS remains in the AS path to the origin.

The last pattern, depicted in Figure 3.8, is named **Me-Too MOAS** to underline its "being over-announced" property. It is composed of both of the previous patterns at a single time: the left-hand side of Figure 3.8 shows a peering MOAS, while the first-level AS peers are arranged in a classical MOAS manner. The mean and median durations of this pattern is shown in the last row of Table 3.3. These values suggest that me-too MOASes are stable. Figure 3.9 confirms that few of these events are short-lived (less than 5% in both cases), and over 80% of them last longer than two months.

We saw this pattern appear in the two following situations. The first one was a combination of subletting of IP space. Using Figure 3.8 as illustration, the prefix block p is owned by AS20 and AS10 rents it. The WHOIS record associated with p clearly stated that prefix p was part of non-transferable IP addresses. So, because AS20 is the owner, it keeps on announcing p. However, since AS10 rents it, it also announces the prefix. This results in a peering MOAS, i.e. the left-hand side of Figure 3.8. Additionally, AS10 assigned p to one of their customer for use. At some point, this customer chooses to do multihoming and uses AS4000 for that purpose. In return, AS4000 announces p as well, i.e. the right-hand side of Figure 3.8. The second situation was when a prefix owner decided to change upstreams. Originally, the owner's prefix was announced by a tier-1 ISP which used multiple AS numbers, one for its global activities (AS20 in Figure 3.8), and one for its local activities (AS10 in Figure 3.8). However the ISP used both of those AS numbers to originate the prefix, although it needs to go through the local AS from the backbone to reach the customer

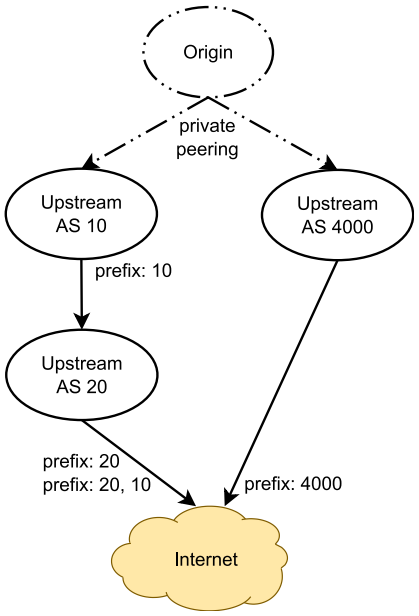


Figure 3.8: Graph of third MOAS pattern: Me-Too MOAS

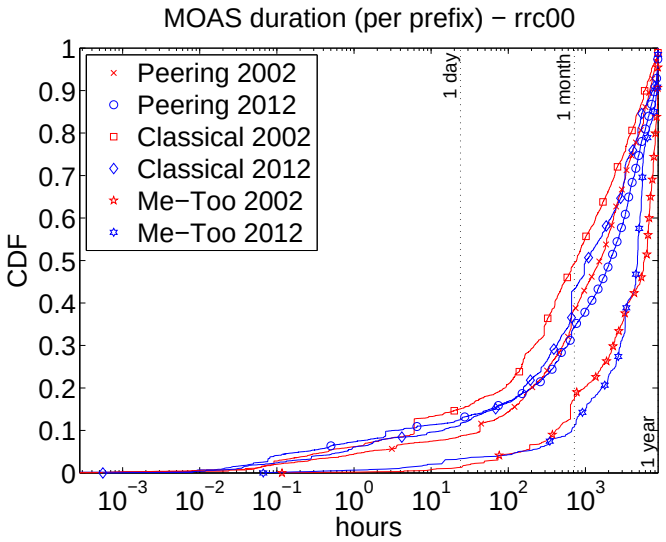


Figure 3.9: MOAS patterns duration (per prefix)

(this corresponds to a peering MOAS). Then, the user (AS at the top of Figure 3.8) decides to switch their ISP service to another tier-3 ISP (AS4000 in Figure 3.8). During the transition, which usually lasts several weeks, the prefix was announced by both the old tier-1 (AS10 and AS20 in Figure 3.8) and the new local tier-3 ISP (Figure 3.8, AS4000). This situation, then, presents a peering MOAS with a classical MOAS.

Per prefix	2002	2012
Peering MOAS (a)	72.55%	72.63%
Classical MOAS (b)	31.09%	28.6%
Me-Too MOAS (c)	5.5%	3.84%
(a) & (b)	6.37%	3.24%
(a) & (c)	1.95%	1.59%
(b) & (c)	1.37%	0.49%
(a) & (b) & (c)	0.52%	0.24%

Table 3.4: Proportion of occurrences of MOAS patterns

Table 3.4 shows that me-too MOAS events amount to 3% to 5% of MOASes. This can be explained by the fact that this configuration is unlikely to arise from erroneous situations, unlike the previous two patterns since it requires (at least) 3 origin ASes for a single prefix, *with* a peering relation among two of them.

The bottom rows of Table 3.4 show the proportion of prefixes that exhibit different types of MOAS patterns. These values suggest that MOAS prefixes only exhibit one kind of MOAS event throughout their lifetime. When we put this information in relation with the MOAS durations in Table 3.3 (125d on average) and the MOAS prefix uptime in Table 3.1 (308d on average), it is clear that MOAS prefixes do not spend their whole uptime in a MOAS state. **However, the fact that the MOAS prefixes do not switch from one MOAS class to another suggests that their configuration remains stable.** We can think of two main reasons why these MOAS announcements would be withdrawn. A reason could be that the owner of the prefix intentionally withdraws this announcement, for example due to exceeding the bandwidth allowance of one of its peer. Another reason is the data bias from our collector router, i.e. these routes are not propagated to the collector anymore because they have been filtered out.

3.4.3 Evolution Over Time

We took care of presenting the results of our analysis for both 2002 and 2012. However, there was no significant difference between the two years, either in terms of duration or proportion. This is remarkable because, over ten years, the size of the routing table and the number of distinct announced ASNs increased by 400% [Potaroo].

3.5 MOAS Filtering: Building a Suspicious MOAS Dataset

In the first part of this Chapter, we studied MOAS events in multiple ways. First, we revisited previous works by looking at MOAS events on their own. Then we considered MOAS events along with their prefix. Grouping the events in that way underlines the relationship between seemingly

independent MOAS events. Most notably, we showed how many short-lived events repeat in order to result in a long-lived MOAS prefix. This observation eliminates the possibility that these events are the result of a misconfiguration, unlike previously reported by [Zhao *et al.* 2001; Chin 2007].

We also looked at the evolution of the topology graph of MOAS prefixes and we classified MOASes into three distinct patterns. The most popular pattern, peering MOASes, is composed of long-lived MOASes where the different origin ASes are directly peering. We consider these as fake MOASes because there is no benefit from the MOAS announcement, yet they make up for just over 70% of all MOAS events. The second class of MOAS is composed of classical MOASes. This is the standard MOAS configuration. However, they only make up for between a quarter and a third of the global MOAS events and global MOAS prefixes. The last pattern, me-too MOASes, is a combination of the other patterns and was encountered as a transitional configuration when an owner was switching its upstream provider to another one. It makes up for 3% to 5% of all MOAS cases.

In the second part of this Chapter, we focus on prefix hijacking, and, more precisely, on concurrent ownership attacks, since, as described in Chapter 2, their side effect is a MOAS. Each of the MOAS patterns that we presented up to now lead to different security implications. We now detail these implications and explain how we constitute a *dataset of suspicious MOAS events* by building a set of **MOAS filters** which discard MOASes resulting from standard BGP engineering practices. This dataset proves to be between 65% and 85% more effective than current state-of-the-art MOAS detection techniques. We then use this dataset to present a system that we use to analyze these suspicious MOAS events. In particular, Section 3.7 provides a case-study of a real-world MOAS occurrence, which we refer to as the *Bulgarian case* and detail *how* and *why* this case is suspicious.

A provider P is always on its customers' ways to and from the Internet. Should P decide to eavesdrop on, or temper with the traffic of one of its customers, it can do so without BGP means. As a result, we can disregard any MOAS that is the result of a provider-customer relationship in the Internet. Methods to infer the relationship between two ASes have been presented before, most notably by [Gao 2001], and are even available as pre-computed datasets such as [IASR]. These methods inferring AS relationships usually output a directed relation (i.e. the provider is assigned as one of the AS, the customer as the other). We are just interested in the existence of such a relationship, not in its direction. By doing this, we avoid raising alerts due to the possibly random order in which we detect the announcing ASes. This reasoning can be extended to any peering relation where transit has been agreed upon by the two peers. In other words, for all BGP peerings that include a transit relationship, there is no gain from doing a concurrent ownership attack. For BGP peerings that do not include a transit relationship, i.e. a *peering relation* as defined by [Gao 2001], undertaking an ownership attack means that a peer would start redistributing to the world the route to the target (peer) network. It is hard to assess the impact of such an announcement because the peer networks are topologically very close (because they are peering). The malicious announcement's visibility is, then, dependent on the policies of the attacker's upstreams, which may or may not prefer this malicious announcement to the legitimate one. For this reason, should a direct peer decide to hijack the traffic of its peer, the best tool would be a subnet attack, (which, in this case could easily be turned into a man-in-the-middle attack). This from of attack would guarantee that 100% of the traffic is flowing through this malicious route. Consequently, it is unlikely that a peer would choose to hijack the traffic of its peer by using an ownership attack. For this reason, we mark all peering MOASes as benign.

Unfortunately, classical MOASes and Me-Too MOASes cannot be filtered out by their AS-level network topology. Both of these patterns can be the result of a legitimate announcement where the

prefix owner's network is connected to its upstreams via a private BGP peering, or some static route configuration. But they can also be the result of the injection of erroneous data in BGP. However, Figure 3.9 shows the number of short-lived MOAS (per prefix) is quite low. This suggests that the bigger bulk of MOASes is indeed composed of stable, long-lived prefix-wise MOAS conflicts. In order to filter these out, we rely on the duration of the announcement, based on [Karlin *et al.* 2006]'s suggestions: the global routing infrastructure's robustness would largely benefit from quarantining new, unknown routes during 24h before allowing them to be used to forward IP traffic. This stems from the fact that a network owner would have enough time to take appropriate actions against a long-lived erroneous announcements. For example, they can contact the attacker and inform them that they are misbehaving. If this does not lead to a withdrawal of the bad announcement, they can contact the upstream of the attacker in order to try and get them to filter out this announcement. In a similar way, we only start trusting an origin when its uptime for a given prefix is bigger than a threshold T. However, because the topology of the Internet is constantly evolving, we also need to stop trusting origins that have not announced a given prefix for a long time. As a result, we discard any origin AS for a given prefix, regardless of its uptime, if that origin has not announced the prefix in the last 30 days. Using this method, the model for a prefix – containing the trusted origin ASNs along with their uptime – is updated as time goes by, and always sticks to what is currently seen in the network.

The goal of the filter that we just described is to take advantage of the standard routing practices, which were uncovered in the first part of this Chapter with the longitudinal MOAS study, in order to filter out as many benign MOAS conflicts cases as possible. Compared to well-established methods, which are mostly derivative of the well-known PHAS algorithm [Lad *et al.* 2006], we suppress between 65% and 80% of alerted prefixes. The remaining cases, composed exclusively of Classical MOAS conflicts (or mixed Me-Too MOAS conflicts), need to be investigated through some other means. In the remainder of this Chapter, we will present a system that enables such an analysis, and we will provide a thorough analysis of a suspicious classical MOAS event.

As a summary, we give here the steps of the MOAS filtering algorithm:

1. Process BGP update messages according to [RFC4271], specifically:
    - an adjacent routing table is created for each peer of our vantage point(s);
    - each route to a prefix p announced by a peer N is added to the adjacent routing table associated to N;
    - when a prefix p is withdrawn by a peer N, the route to p in the adjacent routing table associated to N are erased.
2. When a prefix p is updated,
    - if p is a new prefix (i.e. if p's data is not currently in memory):
        - fetch the previous origins for p, that were seen during the previous 30 days, and that have an uptime longer than 48 hours;
        - if no such origin AS exist, the prefix is in *learning mode* for 24 hours, and any incoming origin for p will be accepted as ground-truth.
    - check if the incoming origin for p is different than the one(s) we already know.
        - if it is not, everything is fine.
        - if the origins are different, raise a MOAS alert. MOAS alerts will be raised for that origin until it reaches an uptime longer than 48 hours.

3. When a prefix p is withdrawn,

   - if p has been withdrawn by every peer (i.e. if p does not appear in any of the adjacent routing tables), then p is not part of BGP routing tables anymore. p is unannounced, has gone offline. Delete the model associated to p from memory, and store it on disk for 30 days.

3.6 Analyzing Suspicious MOASes

In Section 3.5, we detailed how we take advantage of the MOAS analysis provided in the first part of this Chapter in order to define filters that enable us to discard MOAS prefixes resulting from standard BGP engineering practices, hence leaving us with a narrower set of MOAS prefixes that are not, in an obvious way, the result of benign practices, and thus could be the result of a prefix hijacking attack. We called this technique MOAS filtering. In this Section, we introduce a system engineered to study, at a large scale, malicious BGP concurrent ownership hijacks. In particular, we focus on malicious hijackings that would be carried out by so-called fly-by spammers [Ramachandran *et al.* 2006]. This system combines several data sources and analysis techniques in a novel way, which is possible thanks to the collaboration we presented in Chapter 1. On top of using the output of the MOAS filtering algorithm, we utilize live spam feeds collected at spam traps, issue traceroutes towards spamming networks, look for suspicious traffic in NetFlow data collected at a scientific network, and analyze historical dumps of IRRs (Internet Routing Registries). We use this system to provide the analysis of a MOAS case study, and also to show the inner limitations associated with analyzing a routing event from a third-party point of view, and thus the limitation of previous works on fly-by spammers.

The remainder of this Section focuses on the *architecture* of the system to study MOAS cases, and includes a description of the each dataset that we use. Then, in Section 3.7, we provide a highly detailed study of a real MOAS case, with which we illustrate how correlating suspicious routing events with security-related incidents is, unfortunately, *insufficient* to identify BGP hijack attacks performed with malicious intent. We also show how easy and tempting it is to draw quick conclusions, and therefore underline how careful one must be when looking at routing events from the outside. Unfortunately, this amount of caution is not always taken when analyzing routing events, hence suggesting some established analyses should be reevaluated using a larger variety of data sources, most of which is readily available.

The overall system architecture for MOAS analysis is depicted in Figure 3.10. In a nutshell, the data sources are composed of data passively collected from spam traps, BGP feeds, NetFlow data collected at a large academic/research network, and archived copies of IRR databases, as well as measurements resulting from active traceroute probing. In order to obtain suspicious BGP events that pose a security threat we *correlate* the MOAS filtering algorithm (① in Figure 3.10) with spam data collected from spam traps ②, and with traceroute results performed to suspicious, spam-emitting networks ③. We further correlate those prefixes with network footprints using NetFlow data collected at a large academic/research network ④. Finally, we search IRR databases for evidence to (in)validate suspicious BGP announcements ⑤.

The spam dataset is provided by our partners in Symantec Research Labs. The Symantec.cloud spam traps collect data (② in Figure 3.10) and feed it to SpamTracer [Vervier *et al.* 2013] (③ in Figure 3.10), a tool built to monitor the routing behaviour of spamming networks. It performs traceroute measurements towards networks that have sent spam to Symantec.cloud spam traps.

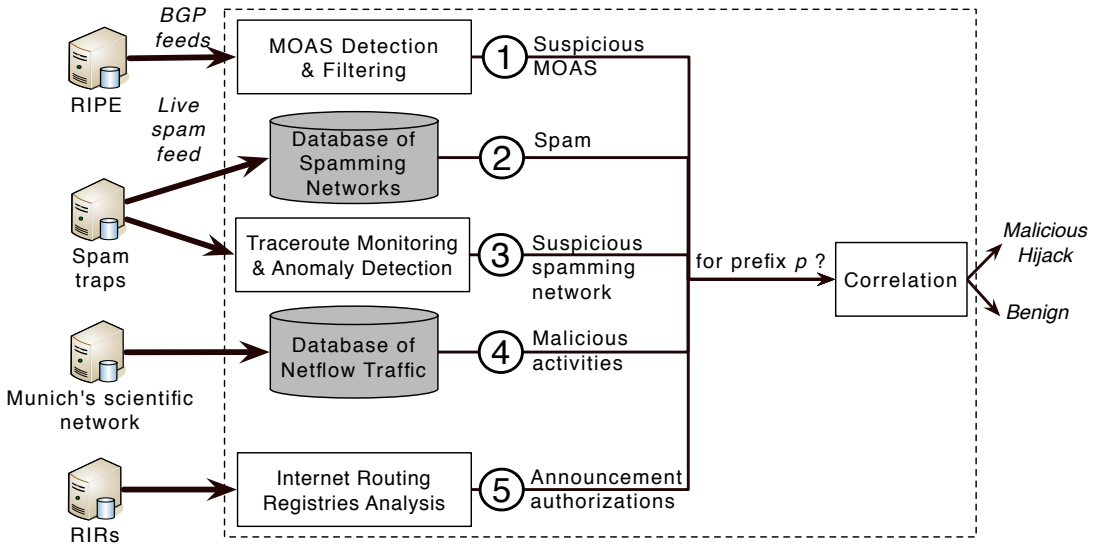


Figure 3.10: Architecture of the MOAS analysis system

These measurements are performed on a daily basis and repeatedly for a certain period of time after spam is received, especially focusing on on short-lived hijacks as observed in [Ramachandran *et al.* 2006] and attributed to by fly-by spammers. SpamTracer is currently able to monitor up to approximately 8,000 network prefixes everyday with one IP address traced per prefix. By performing measurements on consecutive days for one week, data plane paths and BGP routes towards a given network can be compared and analysed in depth to find indications for an ongoing hijack. Because we monitor networks just after spam is received, we expect to observe a routing change as soon as the hijack ends, provided the network was indeed hijacked.

The NetFlow data and its analysis is provided by our partners at the Technische Universität München (TUM). We use this data is to analyze changes in traffic patterns before, during and after a suspected hijack (④ in Figure 3.10). Such changes can range from simple outages in monitored networks, where outgoing connection attempts are unanswered, to changes in traffic volume or even to a significant amount of new connections from and to different sets of ports. The data is collected according to [RFC5103] from the Münchner Wissenschaftsnetz (MWN) (Munich's scientific network) which comprises more than 80,000 end hosts. It is used by researchers, students, and administrative personnel, who generate monthly upstream and downstream traffic volumes of more than 300 and 600 Terabyte, respectively. Consequently, the MWN is large enough to be effected by large-scale spam campaigns, and we expect to observe at least some portions of spam that originate from hijacked networks.

Finally, an administrative point of view is obtained, also courtesy of our partners at the TUM, by searching the IRR databases for conclusive information on suspected hijack events, such as relationships between the involved ASes and IP prefixes. We extract `route` objects from archived daily IRR dumps provided by the different Regional Internet Registries (RIR). These `route` objects are maintained by ISPs or end-users that are responsible for the IP space and allow them to specify which AS(es) should announce a given prefix. Although ISPs and end-users are not forced to keep those records complete and up-to-date, when available, they still provide valuable forensic information on past and present relationships between the holder of an AS and a prefix. Such an ordinary relationship can thus cast a malicious hijack event into doubt. IRR records may also contain meaningful information about prefix and AS holders, e.g. a description of the holder's business or

contact details that can further be used in the analysis of hijack events.

3.7 Case Study: The Bulgarian Case

In this Section, we first present briefly the result of the application of the system presented in Section 3.6 to the month of February 2013. Then, we provide a detailed analysis of the *Bulgarian case*. This analysis is done in two steps. First, it is carried out, like by most other work, by using networking and security information extracted from the various datasets described in Section 3.6. Then, we add administrative information (⑤ in Figure 3.10) to complete the analysis. Doing this, we illustrate the difficulty behind inferring the maliciousness of a suspicious BGP routing event, and underline how the IRRs can be used as a substitute for operator feedback, which is necessary to uncover the ground-truth.

For the month of February 2013, the MOAS filtering algorithm raised alerts on 2,331 distinct prefixes. A time window of 15 days was used to correlate these events with spam from IP addresses observed at spam traps and on blacklists to identify malicious hijacks. In the following we present an in-depth analysis for one of the matching events. Note that all results are anonymized with good cause.

Based on several alarms raised by our detection system on February 3rd, 2013, we became aware of an incident taking place in Bulgaria. Several MOAS conflicts were observed for networks that correlated with emerging spamming activities. We carried out a detailed analysis of these events, and present our results in chronological order below.

3.7.1 First Examination

In this Section, we use the traditional approach of inferring the maliciousness of a BGP routing event – i.e. answering the question "is it a hijack?" – by using the traditional approach of correlating networking data with security data. The analysis is divided in four distinct phases, corresponding to four distinct BGP announcement behaviours. These phases are illustrated at the top of Figure 3.11.

3.7.1.1 Phase 1: Normal Situation

Since 2008, the prefix `A.B.0.0/16` has been announced in BGP by *Alice*, a Tier-3 ISP. *Alice* 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 3.11).

3.7.1.2 Phase 2: Hijack and Spam

On December 4, 2012, *Mallory* started announcing a set of nine more specific (/24) prefixes of *Alice*, who carried on with the original /16 announcement (Figure 3.11). By using online *whois* queries, we learned that *Mallory*, supposedly, is a VPS service provider also located in Bulgaria. A thorough web-search however returned no result for this specific company.

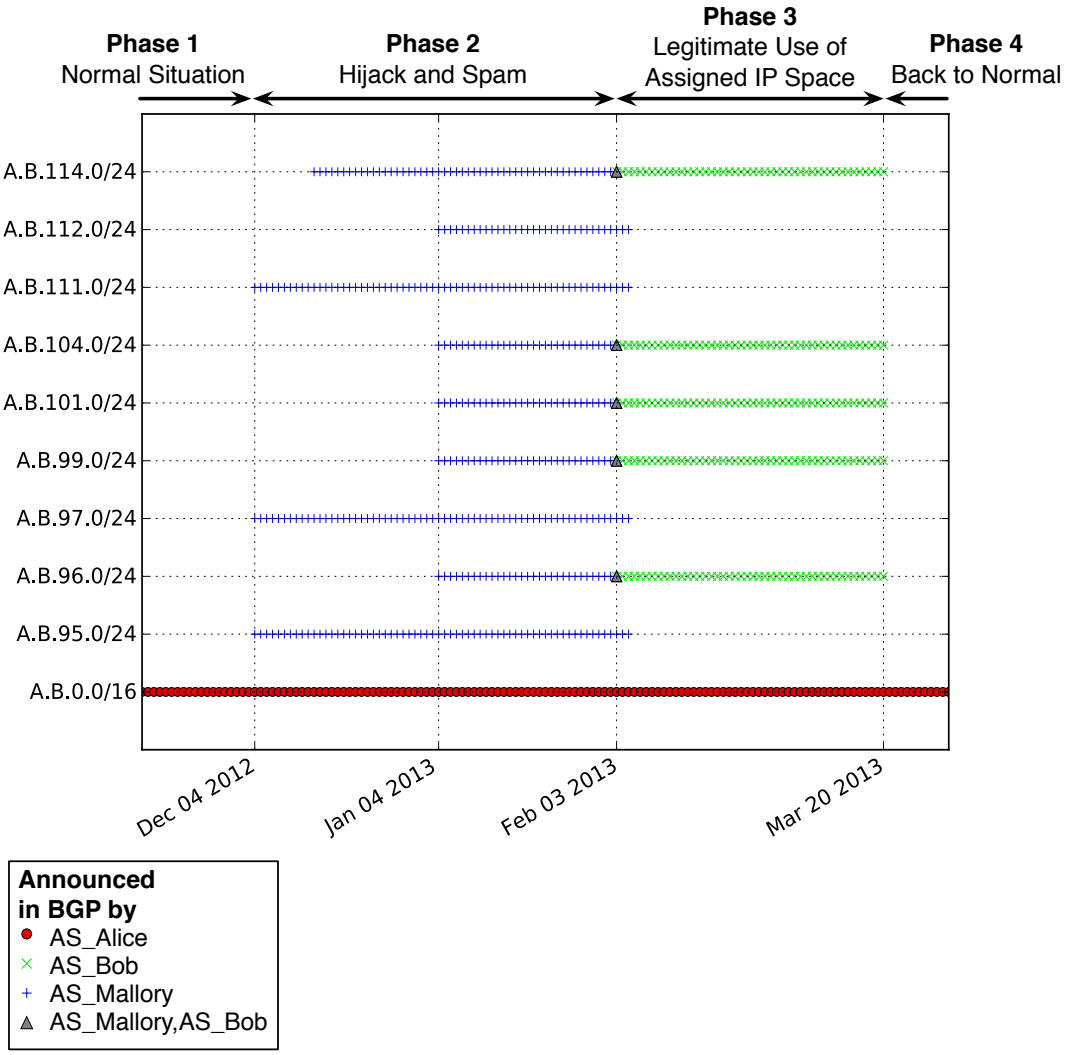


Figure 3.11: Route announcements for the *Bulgarian Case*

Spam

Figure 3.12 depicts spams received by Symantec.cloud spam traps from IP addresses belonging to the nine prefixes announced by *Mallory*. The figure also presents blacklisted IP addresses from Uceprotect Level-1 [Uceprotect] related to these prefixes. This figure shows a strong correlation between the BGP routing announcements, spam, and blacklisted IP addresses. On some days, up to 80 spam emails were sent to our spam honeypots. Many prefixes also had around 100 backlisted IP addresses for several days. On Figure 3.12 we still observe some blacklisted IP addresses after the end of phase 2 but we attribute them to the one-week expiration period of Uceprotect records. The Symantec.cloud spam dataset may provide the spam botnet name responsible for the spam based on spam bot signatures. Because spam bots are usually compromised machines, they should not be observed on hijacked IP space. And indeed no such botnet could be inferred from Symantec.cloud's reports for spam hosts in the suspicious prefixes. This indicates that those machines were likely set up by the spammers themselves.

Scam Hosting Infrastructures

We further analyzed the spam mails and were able to identify several URLs within these messages.

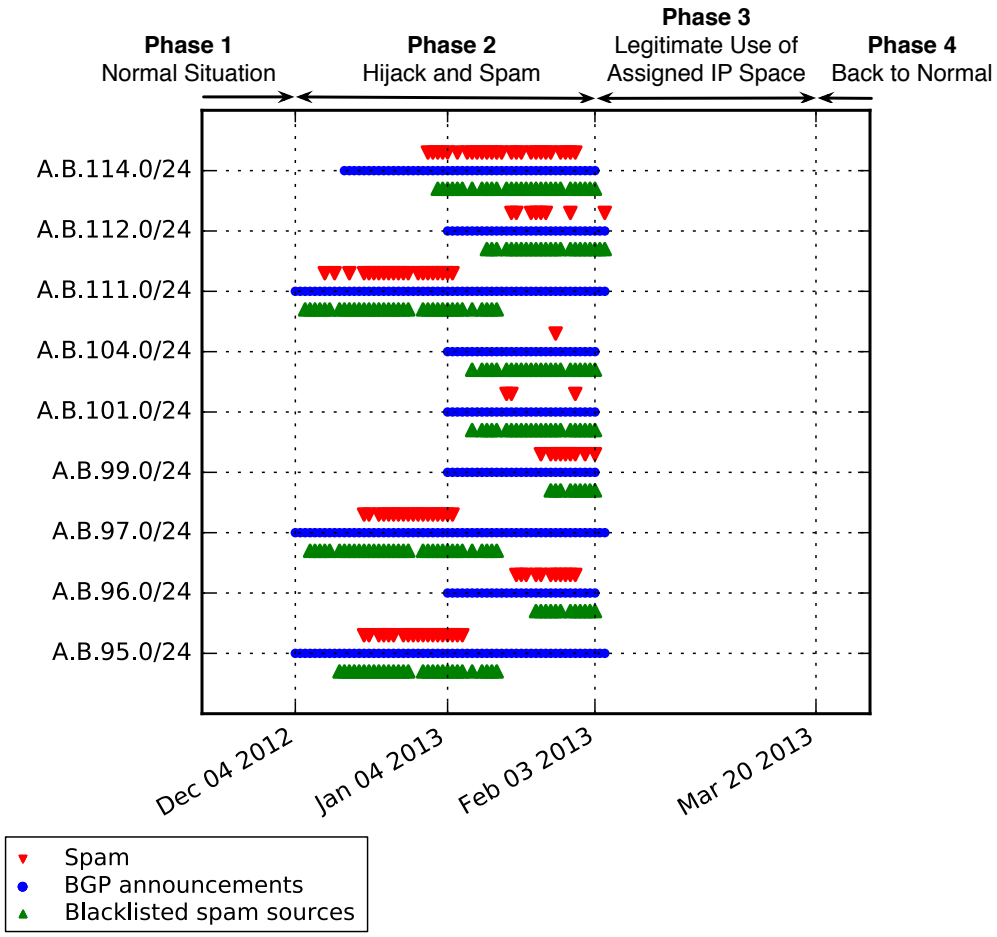


Figure 3.12: Spam received from and blacklisted IP addresses in reported prefixes

Out of 118 extracted domain names, 89 resolved to an IP address within six of the obtrusive prefixes. We conclude that the spam was also used as a platform to promote a scam infrastructure hosted within these prefixes. About 90% of all scam hosts in the nine `A.B.x.0/24` networks coincided with IP addresses of spam hosts, which indicates that the spammers took full advantage of the prefixes under their control.

It is interesting to see that almost all scam hosts' IP addresses shared the same last byte while being spread over all abused networks (e.g. A.B.{95,96,114}.**5**, A.B.{95,114}.**9**, A.B.{95,96,97,114}.**14**, etc). Similar characteristics appear 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*. All pieces of evidence suggest a single administrator behind the domains and network infrastructure.

NetFlow Traffic Analysis

We analyzed NetFlow data for the period of December 2012 to March 2013, and were able to collect 13,001 inbound flows from the suspicious prefixes. The majority of these flows accounted for SMTP requests (71.0%), DNS replies (25.2%), HTTP replies (1.6%) and SMTP replies (1.4%). The remaining 1.8% of flows indicated 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 across all networks of *Alice's* customers seems to be implausible: we thus assume that these flows attribute to IP spoofing activities unrelated to the Bulgarian case, and exclude them from our analysis.

All connection requests (incoming for SMTP and outgoing for DNS and HTTP) are depicted in Figure 3.13. We observe a strong correlation in phase 2 between the BGP announcements (Figure 3.11), the observed spam (Figure 3.12), and the blacklist records (Figure 3.12). We observed a total of 925 IP addresses for the delinquent's activities, of which 850 IP addresses were used to send spam mail. Less than 10% of these addresses were re-used for the DNS and 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 reported prefixes.

This analysis confirms that the prefixes were used in order to massively send spam from several hundred clients. Furthermore, it clearly shows that the person in charge hosted more than 100 live services (DNS and HTTP), presumably to do phishing or similar fraudulent activities.

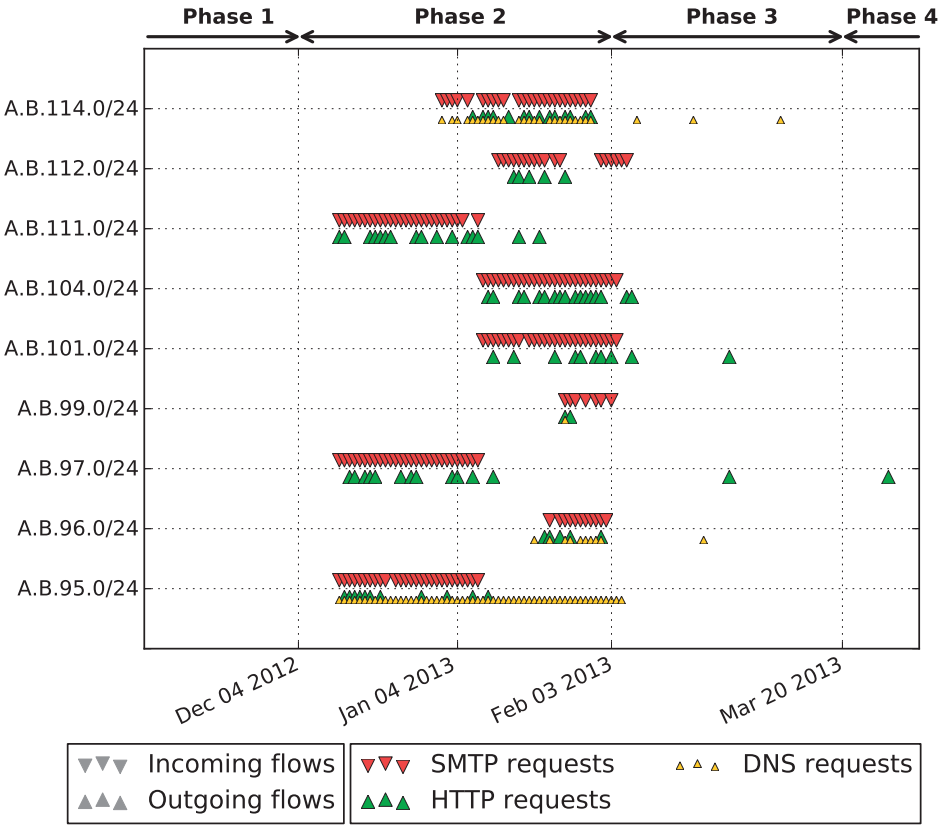


Figure 3.13: Flow data for reported prefixes

3.7.1.3 Phase 3: Legitimate Use of Assigned IP Space

On February 3, 2013, *Bob* started announcing five of the nine prefixes announced by *Mallory*, resulting in MOASes during a few hours before *Mallory* withdrew all of its announcements. *Alice*, once more, kept on announcing the original /16 prefix (Figure 3.11). Several spam hosts that used to reply to traceroute probes on consecutive days during phase 2 also suddenly became unreachable suggesting a real change in network topology.

Bob is a business-to-business IT service provider located in the same country as *Mallory* and *Alice*, according to their website. Its ASN first appeared in BGP in November 2008. All five /24 prefixes were announced via *Alice* acting as legitimate upstream provider. Figure 3.14 depicts the overall topology from a BGP's point of view.

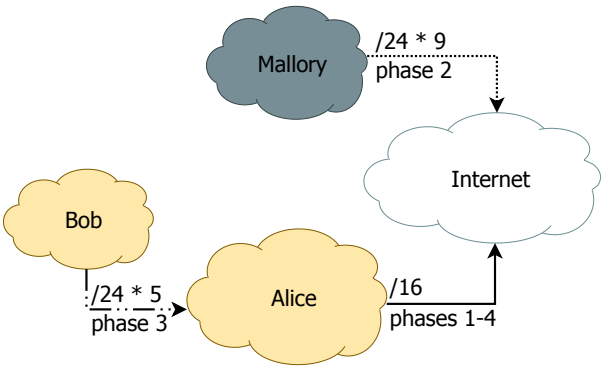


Figure 3.14: Topology derived from BGP

With 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.

3.7.1.4 Phase 4: Back to Normal

On March 20, 2013, *Bob* withdrew its announcements of *Alice's* five prefixes, resulting in the same initial situation as described for phase 1, where the whole prefix `A.B.0.0/16` was announced by *Alice* only.

Given these findings, approaches presented in [Ramachandran *et al.* 2006; Hu *et al.* 2007] would conclude the existence of a malicious BGP hijack. All evidence presented so far, especially the strong correlation for both the control plane and the data plane, lead us to the conclusion that we indeed observed a malicious hijacking event for this *Bulgarian Case*.

3.7.2 Second Examination

Despite the evidence for a malicious hijack incident described so far, we decided to further investigate the case and found significant evidence *against a hijacking event*. We analyzed more than one year of archived RIPE IRR database dumps in order to infer the legitimate owners of the suspected prefixes by searching for `route` objects and looking into the corresponding *origin (AS)* attributes. We found that *Alice* carefully maintained such `route` objects in the RIPE IRR database throughout all four phases. We obtained the first three objects related to the prefixes in question on December 4th, 2012 (Figure 3.15). Their *origin* attributes were set to *Mallory*, and the creation time corresponded to her first BGP announcements. This clearly indicates that – at least according to the RIPE IRR database – *Mallory* was authorized to use these prefixes.

Figure 3.15 gives an overview for all relevant `route` objects that we found in the RIPE IRR database. We learned that the dates of appearance fully match all BGP announcements of *Mallory*

and *Bob* (see Figure 3.11), and all objects were maintained by *Alice*. If we assume that an attacker is incapable to alter the RIPE IRR database at will (and that he had no access to *Alice's* maintainer account), we must conclude that *Alice* delegated all nine prefixes to *Mallory* by choice, and reassigned some of them around February 3rd, 2013 to *Bob*.

We further extracted the database objects' *descr* attributes, and even found some weak evidence for a relationship between *Mallory* and *Bob*. Those free text fields can be set to any value. For *Mallory*, all fields were set to `BG-XX-N`. `BG` indicates Bulgaria, whereas `N` corresponds to each of the prefixes' third byte. More importantly, `XX` represented the initial letters of *Bob*'s company name. After reassignment, the description changed to *Bob*'s full company name.

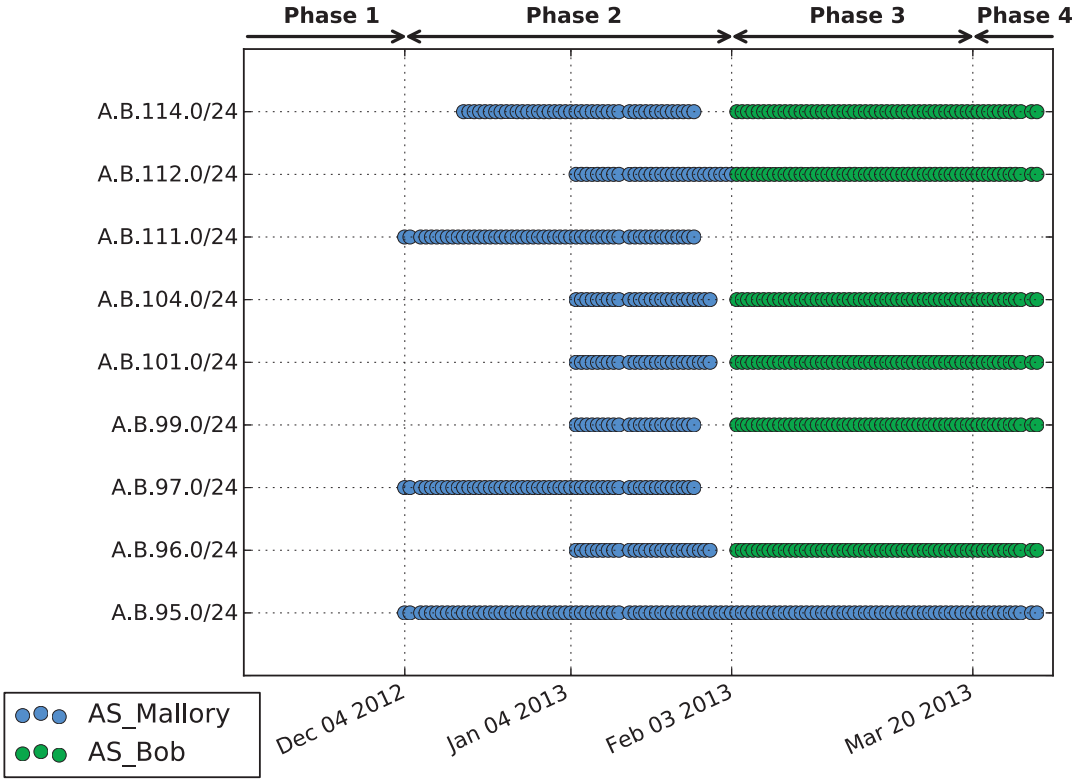


Figure 3.15: RIPE IRR `route` objects for reported prefixes

Finally, we contacted *Mallory's* upstream provider and learned that Mallory requested to announce rented prefixes. After receiving complaints, the upstream provider cancelled *Mallory's* contract.

Given all circumstances, we must conclude that *Mallory* acted maliciously by sending spam. However, we cannot decide if Mallory really hijacked prefixes, or if Mallory just rented the networks for abuse.

3.7.3 Discussion

Even though we have accumulated a series of converging indices incriminating one of the actors, namely *Mallory*, involved in performing BGP hijacks with malicious intent, we still cannot reach a decisive conclusion.

As presented in Section 3.7.1, the strong correlation between the BGP announcements of *Alice's* sub-prefixes by *Mallory*, the spam received by Symantec.cloud and the evidence of scam hosting infrastructures during phase 2 initially led us to believe that *Mallory* had indeed hijacked these prefixes to emit spam. This result is supported by the following observations:

1. The temporal correlation between the BGP announcements and the emerging spam during phase 2 strongly suggests that machines in *Mallory's* network are the spam sources.
2. *Mallory*'s first appearance in BGP as well as the registration date of the domain names advertised in the received spam mails directly coincident with phase 2 of the incident.
3. *Alice* provided upstream connectivity for *Bob*, while *Mallory* hired an independent upstream provider, although *Alice* continuously announced the full enclosing /16 prefix.
4. As soon as *Bob* started to announce his assigned prefixes in phase 3, *Mallory's* announcements and the emission of spam stopped, and no more traffic flows were observed.

Our findings in Section 3.7.2 validate prefix ownership based on the RIPE IRR database for all involved parties during all phases of the incident. However, this fact does not exclude a malicious BGP hijack: it is possible that an attacker covered up his traces by altering objects in the RIPE IRR database. According to RIPE, 86 of database maintainers were using password-only authentication in 2011 [RIPE NCC 2011]. However, password protection may not be enough since an attacker could use information leaked from the IRR database [Halse 2012] and/or phishing e-mails [Bellovin 2012] to gain privileged access to the database.

Our system to detect malicious BGP hijacks was partly designed upon findings of previous studies on the root causes of BGP hijack events, like [Ramachandran *et al.* 2006]'s study on short-lived BGP announcements, the correlation between BGP hijack alerts and spam by [Hu *et al.* 2007] and a validated hijack case performed by a spammer described in [Vervier *et al.* 2013; Schlamp *et al.* 2013]. Comparing our findings presented in Section 3.7.1 with those reported in previous work quickly led us to the conclusion that the *Bulgarian Case* was indeed a malicious BGP hijack. However, the novel forensic analysis of an IRR database described in Section 3.7.2 at least opened our mind that we possibly have not found a real hijack event, but rather a plain abuse of rented IP space. In the end, although we remain indecisive, we learned that it is crucial to consider complementary data sources, preferably as independent as possible (e.g. IRRs) and/or feedback from network owners (e.g. via mailing lists like NANOG as in [Vervier *et al.* 2013; Schlamp *et al.* 2013]) in order to 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 when responding with possibly legal actions.

3.8 Summary and Conclusion

In this Chapter, we focused on Multiple Origin AS prefixes, i.e. MOASes, which are prefixes that are simultaneously announced by distinct autonomous systems. This situation arises either due to standard network practices, such as network multihoming, or due to an attack on the routing infrastructure, e.g. a concurrent ownership prefix hijacking attack.

In the first part of this Chapter, we focused on analysing the benign network cases that result in MOASes. We first updated previous works [Zhao *et al.* 2001; Chin 2007] by studying MOASes

durations. Then, we considered MOASes as a group of events related to prefixes. Doing so, we showed how short-lived MOASes are not, unlike previously reported, the result of misconfigurations. We continued by introducing a taxonomy of MOASes as three distinct patterns. The most popular pattern, named peering MOAS, makes up for around 70% of all MOASes and is composed of long-lived MOASes where the different origin ASes are directly peering. The second pattern, named classical MOAS, is the standard and expected MOAS configuration much discussed by [Zhao *et al.* 2001; Chin 2007]. It makes up for between a quarter and a third of MOASes. The last pattern, named me-too MOAS, is a combination of the other two patterns and was encountered when as a transitional state between multiple providers. We also looked at data 10 years apart, and showed that there is almost no difference in all studied MOAS properties, even though the global network size radically grew.

In the second part of this Chapter, we presented the security implications of the three MOAS patterns by reasoning on the AS graph resulting from the MOAS patterns, as well as on the MOAS announcement duration. With these implications, we detailed the MOAS filtering algorithm, which we use to build a dataset that contains networks whose MOAS is not the result of an obvious network engineering practice. We then correlated this dataset with various network and security datasets in order to find malicious hijacks. We presented a thorough analysis of the *Bulgarian case*, a suspicious MOAS case which coincided with spam originated from the affected prefixes. We further observed a variety of scam activities hosted on these prefixes and we finally put together conclusive evidence for an ongoing hijack attack. With similar findings, previous work would have concluded the existence of a malicious hijack case. We decided to question our results and learned that the presumed delinquent might have legitimately rented IP space to carry out his malicious activities. We thus conclude that considering multiple and independent data sources, such as BGP and traceroute routing data, spam and NetFlow security data and IRR data or feedback from network owners is primordial to avoid drawing conclusions biased by a limited set of evidence possibly skewed towards one verdict or the other. We consequently suggest that previous cases should again be put to test, and conclude that state-of-the-art detection systems have still great room for improvement for the study of malicious BGP hijacks.

Publications

The material presented in this chapter led to the following publications:

- Quentin Jacquemart; Guillaume Urvoy-Keller; and Ernst Biersack.
  "A longitudinal study of BGP MOAS prefixes".
  In 6th International Workshop on Traffic Monitoring and Analysis (TMA 2014), April 2014.

- Pierre-Antoine Vervier; Quentin Jacquemart; Johann Schlamp; Olivier Thonnard; Georg Carle; Guillaume Urvoy-Keller; Ernst Biersack; and Marc Dacier.
  "Malicious BGP hijacks: appearances can be deceiving".
  In IEEE International Conference on Communications (ICC) – Communications and Information Systems Security (CISS) Symposium, June 2014.

Overlapping Prefixes

4

In Chapter 3, we looked at multiple origin AS prefixes, called MOASes, and investigated the reasons behind legitimate announcements of such prefixes. We then designed a set of filters that remove these events from the global MOAS list in order to focus and analyse those that could be the result of a BGP hijacking attack.

Now, we focus on overlapping prefixes, i.e. multiple prefixes that induce the same IP range, and follow a similar approach to the one for MOAS prefixes. Sub-prefixes are the natural side effect of prefixes sub-allocations, and can be used in order to do traffic engineering. This is possible because, as detailed in Chapter 2, traffic is always forwarded towards the most specific announced prefix, i.e. the most specific sub-prefix. But they can also be used in order to hijack the traffic destined to a less specific prefix, i.e. a subnet attack.

In this Chapter, we describe and clarify the use of overlapping prefixes. In order to do this, we use Internet Routing Registry (IRR) databases as semantic data in order to group IP prefixes into *families of prefixes* that are owned by the same organization. We introduce several metrics that enable us to study how these families behave in terms of IP space overlap and in terms of AS-level topology. We look at the AS-level topology of overlapping prefixes and show that different topologies have very distinct use cases. We look at real-world examples and show that the network behaviour of multiple companies in the same business area is similar. Finally, we present the prototype of a system that aims at validating legitimate sub-prefix announcements.

4.1 Introduction

The IP space has been divided into a set of IP prefixes that are assigned to organizations by RIRs (Regional Internet Registries). These organizations can choose to further divide the IP prefixes they were assigned in smaller IP spaces that they can use as independent networks. This is possible because packets are routed according to the longest prefix match rule, as detailed in Chapter 2. In other words, any traffic will always be forwarded to the most specific IP prefix containing the destination IP address. This can be useful in order to do traffic engineering, e.g. to make sure off-site

servers are reachable from the global Internet. At the same time, the recent hijacking attack against Spamhaus demonstrated that announcing more specific prefixes can be used to launch an effective DoS (Denial of Service) attack [Toonk 2013]. Even in the case of misconfigurations, large-scale repercussions can be disastrous [RIPE NCC 2008a].

The global BGP routing table currently contains over half a million entries. It is consequently improbable that there is no overlap among them. It is even more improbable that they all result from a prefix hijacking attack. As a result, the first part of this Chapter focuses on the essential task of describing and understanding the uses of overlapping prefixes from a BGP point of view. A way of doing this would be to create pairs of overlapping prefixes, and then compare them together. As an example, let us consider the three overlapping prefixes a/8, a.b/16, and a.b.c/24. How relevant is the study of the three pairs of these prefixes? If the organization to which the /8 has been assigned is an ISP, the /16 prefix might have been sold to one of its customer; and it is solely this customer who decided to create the /24 subnet. Hence, the comparison between the /8 and the /24 is not meaningful. Conversely, if the organization behind the /8 prefix is not an ISP, the /16 prefix is likely not a sub-allocation, but the result of network engineering.

Therefore, by trying to simply compare pairs of overlapping prefixes, we ignore assignment policies. Namely, IP blocks are assigned by RIRs to organizations. These organizations then use their IP space as they see fit. An ISP, for example, will most likely sell a part of its IP space to customers, who, in turn, will use the (sub) IP space as they see fit. As a result, simply comparing any pair composed of any overlapping prefix disregards the fact that different entities may administer the prefixes. In order to overcome this problem, we use the prefix assignment information included in IRR (Internet Routing Registry) databases in order to cluster overlapping BGP prefixes into **families of prefixes**. Prefixes inside these families are then guaranteed to be under the control of a single organization. Consequently, their comparison can be done without ambiguity. These families are composed of two types of prefixes: **children prefixes**, which are BGP announcements that are *not* included as-is in the IRR databases; and **family fathers**, which *are* included in the IRR databases. We define a large set of metrics to analyse the behaviour of these families. These metrics shed light on the amount of overlap inside families, but also in-between families. We also look at the AS-level topology of these prefixes and see how they relate to each other. We look at a few real-world examples of families, and show that the behaviour inside groups of families of tier-1 ISPs, tier-3 ISPs, and private corporations, are comparable. At the same time, we investigate the distributions of prefixes inside BGP and inside the IRR databases, and offer possible reasons for their large size difference.

In the second part of this Chapter, we focus on the abuse of this phenomenon. Namely, we focus on subnet attacks, and on **sub-MOASes**. Chapter 3 focused on regular MOAS prefixes, i.e. a single prefix announced from multiple origin ASes. Sub-MOAS are the situation in which a sub-prefix s of an announced prefix p is announced from a different origin AS than the one of p. To use the same notation as the one previously defined in Chapter 3, a sub-MOAS is a situation in which $\mathcal{O}(s) \neq \mathcal{O}(p)$. We present a prototype that *validates* sub-MOAS announcements through the combination of information extracted from three data sources. First, we apply the recommendations we set forward while providing the analysis of the Bulgarian case in Chapter 3 by looking for a relationship between prefixes s and p in the IRRs in order to rule out false positives. Second, we use the AS-level topology of both prefixes and compare them together. Third, we use of an IPv4-wide dataset of cryptographic keys that we capture following an SSL/TLS handshake with HTTPS services. Using this prototype, we are able to validate about half of sub-MOAS events in the covered IP space.

4.2 Related Work

Previous work in this area can be divided into three categories: work that analyzes the BGP routing table growth; work that aim to validate BGP routing announcements using IRR data; and work that tries to detect subnet prefix hijacking attacks.

The evolution of the BGP routing table has been studied many times, most famously by [Potaroo], which reports on the growth of the routing table size from the mid 1990's to today. The analysis also includes AS number usage, average AS path length, and other typical BGP aspects. Other papers, such as [Bu *et al.* 2004], investigate the reasons behind this growth, and classify the prefixes inside the routing table depending on the reason for which they need to be announced. The methodology used by [Cittadini *et al.* 2010] to study the evolution of aggregation practices over time may bear some similarity to ours, but differs in several key aspects. Most notably, [Cittadini *et al.* 2010] provides limited prefix grouping methodology, where we make active use of the semantic information found in the IRR databases in order to group prefixes into families that are owned by the same organization. We consequently consider assignments made at the edge of the network by tier-2/tier-3 ISPs. We better illustrate and explain the relationships between the overlapping prefixes inside these families because a large part of our analysis is dedicated to the AS topology shared by these prefixes, whereas [Cittadini *et al.* 2010] focuses more on the dynamics of the BGP announcement and their consequences on BGP router processes. As a result, our methods are not directly comparable, even though the BGP-sides of the analyses exhibit similar global trends.

Validation of routing data based on IRR databases entries has been attempted to make the BGP infrastructure more robust, and less prone to errors and/or malice. For example, [Siganos *et al.* 2007] used IRR data to build a tool that informs network administrators of an anomaly that should be further investigated. More recently, [Khan *et al.* 2013] studied the validity of the association between a prefix and its origin AS in the IRRs. The overall conclusion of this type of work is that the quality of the data inside IRR databases is highly dependent on its RIR, but that relying on it would still improve the security level of BGP.

The existing techniques for detecting malicious subnet attacks have been extensively detailed in Chapter 2. For example, [Lad *et al.* 2006] proposed an extension of their system to cover sub-allocations and raise alerts whenever new sub-prefixes appeared. [Hu *et al.* 2007] proposed a verification based on active probing with spoofed IP addresses, which is not always possible. [Zheng *et al.* 2007] used a hop-count metric to detect network topology changes. [Zhang *et al.* 2008]'s methodology uses traceroutes, but works on a local level only. Argus, which was detailed in Chapter 2, uses a series of distributed monitors to distinguish between a prefix and its sub-prefix, taking advantage of the convergence time by dividing the Internet into an affected and an unaffected zone at the time at which the sub-prefix is first announced [Shi *et al.* 2012]. [Wählisch *et al.* 2012] correlate the routing policies and the RPKI.

4.3 Methodology and Datasets

In this Section, we first describe the datasets we use in order to cluster the prefixes inside the global BGP routing table. Then, through the use of an example, we provide a formal definition for our classification, as well as a formal definition and interpretation of the metrics that we use to analyse the overlaps among prefixes. We use this methodology in Section 4.4 and apply it to the global BGP routing table during the month of August 2014.

4.3.1 IRR Databases

We were able to secure access to the IRR databases of the five RIRs: AfriNIC, ARIN, APNIC, LACNIC, and RIPE. These databases contain information directly provided by network operators, on a voluntary basis, about their routing policies and announcements. They are composed of different objects that represent, among other things, people, IP address allocation, and AS numbers. We extract information from the `inetnum` objects, which contain "details of an allocation or assignment of IPv4 address space" [APNIC Whois Guide]. In ARIN's case, objects similar to `inetnum` are named `NetRange`.

Since BGP uses exclusively the CIDR representation, we convert the IP ranges inside `inetnum` objects into CIDR prefixes. Some ranges cannot be represented by a single CIDR prefix because they are the superposition of multiple contiguous CIDR prefixes of variable mask length. In other cases, the specified IP range explicitly removes the net address and the broadcast address from the range. Regardless of the reason, if we cannot map the exact IP range to a single CIDR prefix, we disregard the value, in order not to introduce false data into our system. The first part of Table 4.1 details the number of entries in each IRR databases, as well as the low number of entries that were discarded. These values result from the parsing of the IRR databases as of August 1st, 2014.

RIR	Total	Parsed	Discarded	Discarded (%)
AfriNIC	73,624	72,516	1,108	1.50%
APNIC	1,454,444	1,432,154	22,440	1.54%
ARIN	2,729,022	2,696,539	29,400	1.08%
LACNIC	326,051	322,828	0	0.00%
RIPE	3,890,179	3,846,706	43,473	1.12%
Total	8,473,320	8,370,743	108,411	1.28%
Filtered		8,364,909	5,834	0.07%

Table 4.1: Number of CIDR IP prefixes extracted from IRR databases per RIR as of August 1st, 2014

Moreover, for database completeness and/or user friendliness, some RIR include additional information in their databases, such as which RIR is responsible for a given class A network, or the IP blocks of private use IPs. In order to remove all of these bogus entries, we use the [Address Space Registry], and discard reserved/special use IP space. Finally, we obtain 8,364,909 distinct IP prefixes from the IRR databases, which amounts to 99.93% of the total number of parsed entries, as indicated by Table 4.1. Because we know these prefixes have been assigned to an organization, we use them as anchor points for our analysis, as will be further detailed in Section 4.3.3.

4.3.2 BGP Data

Our source of BGP data is [RIPE RIS]. We parse binary files that contain a dump of the BGP messages exchanged between the RIPE collector router and its BGP peers. We focus on `update` messages, that contain prefix announcements and withdrawals, as well as the AS path to the prefix. As detailed in Chapter 2, the AS path is an attribute that contains the list of ASNs which need to be crossed before reaching the destination, which is indicated by the last number in this list, and is known as the *origin* AS.

We process BGP `update` messages according to [RFC4271], as detailed in Chapter 2. Namely, we maintain an adjacency table for each of our peers. A prefix is reachable if at least one of our peers has announced it; and is not reachable once every peer that had announced it has withdrawn it. In this way, we are able to build our own BGP routing table, which is dynamically updated as BGP messages flow between routers.

We selected [RIPE RIS]'s Amsterdam collector (rrc00) as our data feeder. It is the best-connected RIPE collector, with over 40 geographically diversified peers. The selected time window for the analysis was the whole month of August 2014, where we counted 629,595 distinct IP prefixes.

4.3.3 Definitions

In Section 4.1, we stated why simply comparing overlapping prefixes together does not produce meaningful results. Instead, we use a combination of semantic data that we extract from the IRR databases, and routing information that we get from BGP. In this section, we present how we group these elements into *families of prefixes* that are composed of a family *father*, of *children*, and of *subfamilies*.

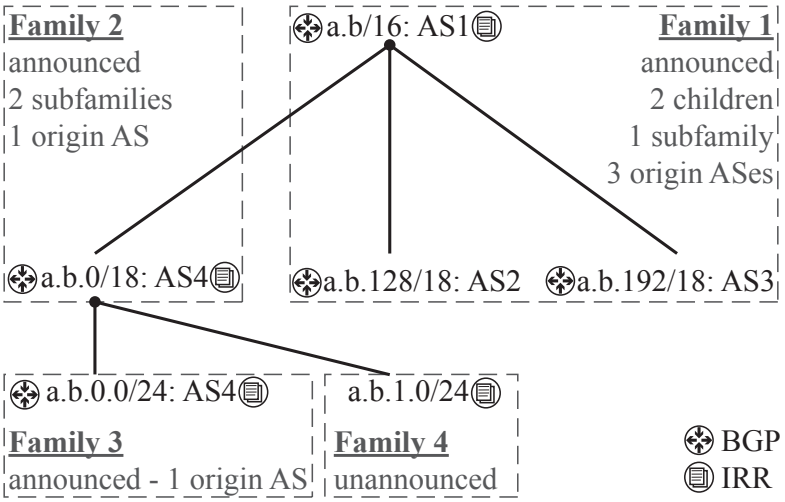


Figure 4.1: Example of constitution of families and subfamilies

Each prefix included in the IRR databases is *always* the **father of a family**. Consequently, we have as many distinct families as the number of filtered IRR prefixes (see Table 4.1). Because most of these prefixes overlap, some family fathers completely include some other family fathers. This situation leads to subfamilies. A **subfamily** is a family whose father is completely included in the IP space generated by another family's father.

For example, there are 4 distinct families in Figure 4.1, because the prefixes a.b/16, a.b.0/18, a.b.0.0/24, and a.b.1.0/24 are included in the IRR databases. Incidentally, these 4 prefixes are the fathers of their families. However, some of these fathers overlap. As a result, in Figure 4.1, Family 2 is a subfamily of Family 1, because the father of Family 2 is more specific than the father of Family 1. Similarly, Family 3 and Family 4 are subfamilies of Family 2. However, neither Family 3 nor Family 4 is a subfamily of Family 1 because Family 2 "hides" them from Family 1. This accounts for the fact that a.b.0/18 has been delegated to another entity (because it has an IRR entry). In

other words, the organization responsible for Family 2 is the one in charge to further subdivide this IP space.

Once the families have been put together, we populate them with BGP data. A prefix seen in BGP is either a family father, or a prefix more specific than a family father. In the first case, there is nothing left to do: a father already belongs to the family it defines. In the second case, the prefix is added to the family as a child prefix. A **child** is a prefix seen in BGP that is more specific than the family father, but not declared in the IRR databases as having been assigned to another entity. The child is consequently managed by the organization linked with the IRR record of its family father.

Continuing with the example depicted in Figure 4.1, three family fathers are announced in BGP: a.b/16, a.b.0/18, and a.b.0.0/24. Moreover, two non-IRR prefixes (a.b.128/18 and a.b.192/18) are also announced. Since they are both more specific than a.b/16, they are added as children in Family 1.

To summarize, we use the prefixes in the IRR databases as a binding link between an organization in the real-world, and one or several BGP prefixes. An IRR prefix induces a family, which contains a certain number of children (BGP prefixes).

4.3.4 Metrics

In this section, we present the metrics that will be used in Section 4.4 to analyze prefix families.

4.3.4.1 IP Space Overlap

The **number of children in a family** indicates the number of assignments that have been done internally in this family. In other words, this is the number of distinct IP zones that exist in this family, each possibly leading to different locations, but which should all be under the authority of the same organization. We put this number in relation with the **number of aggregated children in a family**, which is the number of prefixes resulting from an aggregation process on the children prefixes. Both sets of prefixes generate the exact same IP space, but the aggregated set does so with the minimal number of prefixes. Consequently, a difference in the number of children and the number of aggregated children indicates that internal assignments were done with contiguous IP blocks. For example, in Figure 4.1, Family 1 has 2 children: a.b.128/18 and a.b.192/18. These prefixes define IP addresses that are contiguous, and they are aggregated as a.b.128/17. Thus, Family 1 has only 1 aggregated child.

The **number of subfamilies** in a family indicates the number of prefixes that have been delegated to other entities. This number is a constant in our method, because it results from the contents of the IRR databases. We put this number in relation with the **number of announced subfamilies**, i.e. the number of subfamilies that were actually announced in BGP. We consider that **a family is announced** at time t if either the family father or one of the family child is announced in BGP at t. As an example, Figure 4.1 depicts Family 2, which has 2 subfamilies. However, since a.b.1.0/24 is not announced in BGP (and has no child), it is marked as unannounced. Consequently, Family 2 only has 1 announced subfamily.

The **children overlap ratio** is the ratio of the number of IP addresses available to family children divided by the number of IP addresses available to the family father. In the same fashion,

the **subfamily overlap ratio** is the ratio of the number of IP addresses available to the *announced* subfamilies divided by the number of IP addresses available to the family father. For example, the children overlap ratio for Family 1 of Figure 4.1 is 0.5, and the subfamily overlap ratio is 0.25.

4.3.4.2 AS-Level Topology

The **number of origin ASes** inside a family is the number of distinct originating ASes for the family father and its children. It indicates the number of AS numbers that are originating a prefix within the family. There are 3 origin ASes for Family 1 in Figure 4.1: AS1, AS2, and AS3.

If the number of origin ASes within the family is greater than 1, the family father and its children are not located in the same AS. For this reason, we consider the **ratio of family children that have the same origin AS as the family father**, which is the number of children that have the same origin AS divided by the total number of children. Similarly, the **weighted ratio of family children that have the same origin AS as the family father** is the ratio of the total number of IP addresses that are announced by children whose origin AS matches the family father's divided by the total number of IP addresses in the family. These ratios are both 0 for Family 1 of Figure 4.1.

The **ratio of family children that are behind (resp. in front of) the family father** is the ratio of the number of children prefixes whose origin AS is located behind (resp. in front of) the family father's origin AS divided by the total number of children in the family. In other words, these are children (resp. fathers) whose AS paths contain the family father's (resp. the children's) origin AS. In the same fashion, we have the **weighted ratio of family children that are behind (resp. in front of) the family father**, which is the ratio of the number of IP address that are announced by children that are behind (resp. in front of) the family father's origin divided by the total number of IP addresses generated by the family father. If we state that, in Figure 1, all the AS paths for the two children either end by `2 1` (i.e. AS2 is located before AS1), or by `1 3` (i.e. AS3 is located behind AS1), the regular ratios are both 0.5, while the weighted ratios are 0.25.

In contrast, the **ratio of family children that are in front of the family father** is the ratio of the number of children prefixes whose origin AS is located in front of the family father's origin AS divided by the total number of children in the family. In other words, these are children whose origin AS are in the family father's AS paths. In the same fashion, we have the **weighted ratio of family children that are in front of the family father**, which is the ratio of the number of IP addresses that are announced by children that are in front of the family father divided by the total number of addresses generated by the family father prefix. We will relate those concepts of being "behind" and "in front" of the family father with practical deployment examples in Section 4.4.3.

As these definitions underline, we always compare family children with the family father. This is because the family father is the semantic anchor point that connects the IRR databases with the BGP data. However, if the family father is not announced in BGP, the comparison cannot be done. In this case, these metrics are just not computed.

4.3.4.3 Temporal Evolution

We saw in Section 4.3.2 how we conserve the routing dynamics by parsing BGP messages and recreating the BGP routing table. This enables us to compute time averages for each of the family

metrics. The **weighted average $\mathbf{M}_j$ of metric $\mathbf{m}_j$** is

$$M_j = \sum_i \delta_{t_i} \, m_j(t_i)$$

with δ_{t_i} being the normalized time weight

$$\delta_{t_i} = \frac{t_{i+1} - t_i}{T}.$$

In other words, the weight δ_{t_i} is the time duration $t_{i+1} - t_i$ between which the value $m_j(t_i)$ at time t_i of metric m_j remained the same, normalized by the total duration T during which the family was announced. In case some metric could not be computed (e.g. because the family father was not announced during δ_{t_i}), the contribution of the non-computed term to the average is set to 0.

4.4 Behind IP Prefix Overlaps

This section presents the results of the application of the metrics defined in Section 4.3.4 to real-world BGP data. We then show how different companies make use of sub-allocations, and how distinct companies with similar business interests also make similar use of their Internet resources.

4.4.1 BGP vs IRR Databases

In this section, we compare the prefixes inside the IRR databases and the prefixes announced in BGP.

Figure 4.2 plots the number of prefixes according to their mask lengths, as seen in BGP and in the IRR databases. The distribution of BGP prefixes peaks for class A, class B, and class C prefixes, as reported before, notably by [Potaroo]. This heavily contrasts with the distribution of prefixes inside the IRR databases. First of all, as we mentioned in sections 4.3.1 and 4.3.2, there are over 13 times more prefixes in the IRR databases than in the BGP routing table.

Figure 4.3 plots the number of announced and unannounced families according to the length of the mask of the family father. By definition, the sum of the values of these two bars equals the value of the IRR bar in Figure 4.2 for the same mask length. Here again, we see that the number of announced families abruptly decreases for family fathers whose mask is longer than 24.

In both figures, we see that the prefix behaviours in BGP and in the IRR databases are very different for prefix masks longer than 24. There are two possible reasons for such a difference. First, these prefixes have a long mask, and BGP good practices indicate that prefixes longer than /24s should not be propagated [Hu *et al.* 2007]. This is confirmed by the fact that the total number of such prefixes seen in BGP amounts to around 1% of the total number of prefixes. Second, IRR databases entries are not restricted to BGP users. Any assignment of IP blocks, for example by an ISP, is a potential entry in the IRR databases, even though the ISP and its customer are not connected via BGP (but, for example, via DSL or cable). This also explains the high number of /32 entries in the IRR databases (i.e. single IP addresses): these may be dedicated servers, and an entry in the IRR provides the rightful technical contact information. For unannounced families, there is a difference between the owner of the IRR prefix and the (BGP) manager of the prefix. The manager of the prefix is generally the ISP of the owner, the one that makes sure that the network is

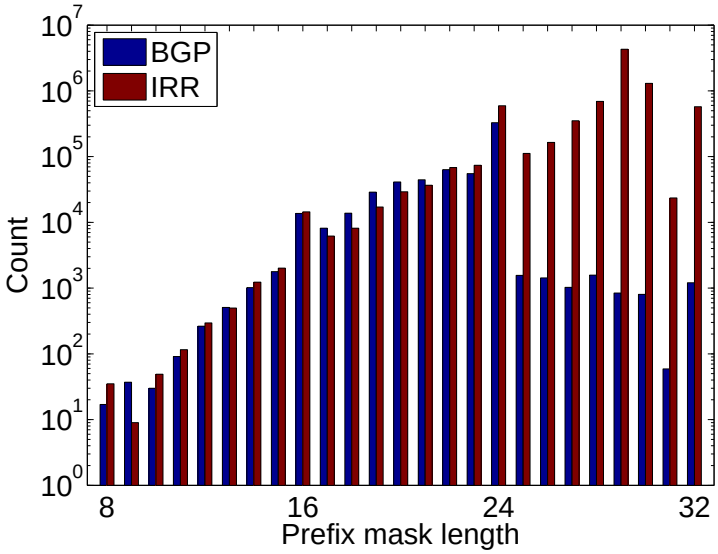


Figure 4.2: Distribution of prefix mask lengths in the BGP routing table, the IRR databases

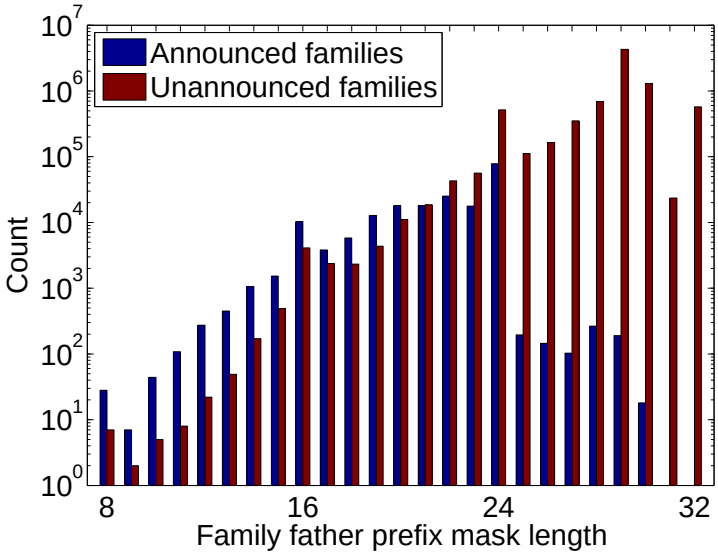


Figure 4.3: Distribution of prefix mask lengths of BGP-announced family fathers, and BGP-unannounced family fathers

adequately connected. The owner of the prefix is the organization actually hosting machines on the IP addresses within the prefix, which is what the IRR entry specifies. For example, one of Eurecom's prefix is 193.55.113.0/24, which is announced from our provider, Renater, as an aggregated /15 prefix. However, the `inetnum` object for the prefix points to Eurecom, even though it is maintained by Renater.

We now focus on the relative size of children and subfamilies in a family. Figure 4.4 plots the distribution of the mask length of children prefixes according to the mask length of the family father. The x axis represents the mask length of the family father, and the y axis represents the mask length of the child. The plot data is the histogram of the distribution: the thicker the line is at a coordinate, the more prefixes there are of this size. As we can see, the bulk of the distribution is around children with a mask length of 24, regardless of the father. The fact that the distribution of children prefixes does not depend on the size of the father is surprising. Indeed, one would expect larger families to divide their IP space into bigger zones. The sparsity of available IPv4 addresses could explain this observation since RIRs and, consequently, ISPs prefer to distribute smaller blocks.

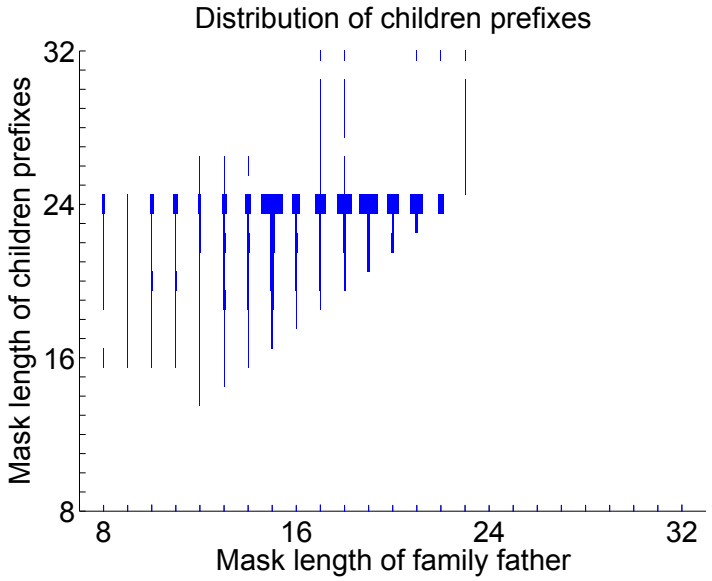


Figure 4.4: Distribution of children prefix mask length depending on family father mask length

Figure 4.5 plots the distribution of the mask length of subfamily fathers prefixes according to the mask of the family father. Here, the bulk of the distribution is around /29, regardless of the mask length of the family father. This raises the question of why these assignments appear to be so popular. In our view, a /29 prefix contains 6 usable IP addresses, which, in today's Internet, is just enough for a small-to-medium size corporation: a couple of IP addresses for publicly accessible servers, plus a couple more for NAT gateways. As tier-3 ISPs typically offer Internet access to a number of SMEs, this could naturally result in a predominance of /29 assignments.

Finally, we have 194,465 families announced in BGP. This amounts to only 2.32% of the total number of families from the IRR databases. The results that we present now apply only to those announced prefixes; nothing else can be said about the other ones strictly from a BGP point of view. Moreover, the figures in the remainder of this section *always* plot the *time-weighted average* of the specified metrics. As a result, plots of discrete metrics show continuous values. For example, a plot showing 0.1 child could mean that there was a single child, but 10% of the time.

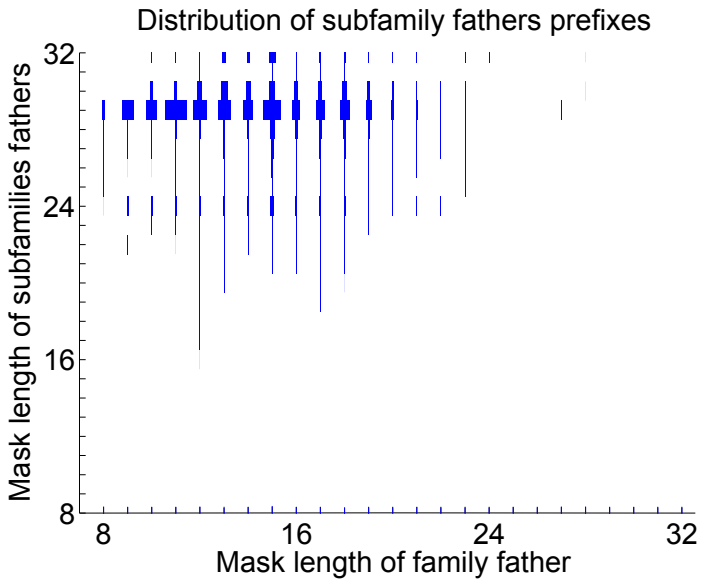


Figure 4.5: Distribution of subfamily fathers prefix mask length depending on family father mask length

4.4.2 Children and Subfamilies

In this section we study the number of children and subfamilies a family has. We look at the IP space occupied by children and subfamilies; and we look at the correlation between children and subfamilies.

Figure 4.6 plots the number of children per family, and the number of aggregated children per family. It shows that only around 25% of families have, on average one or more children, while the probability of having a large number of children decreases very rapidly. Comparing with the number of aggregated children, we see that in 16% of cases, the families have one aggregated child. This means that, for 16% of families, the IP space dedicated to children is contiguous. This indicates that prefix owners prefer to assign contiguous IP space in order to avoid fragmentation (which may lead to more complex and error-prone network configurations).

Figure 4.7 plots the number of subfamilies per family. The plot indicates that in the IRR databases 83% of families don't have a single subfamily. This can be explained by the rather large number of very-specific assignments (masks $\geq /29$) in the IRR databases: these prefixes are directly allocated to end networks, not to networks providers. On the other hand, only 6% of announced families have at least a child, with 1% of them having less than a child on average.

Figures 4.6 and 4.7 reveal that the vast majority of announced families have neither children nor subfamilies. Table 4.2 shows the proportion of announced families, according to having children or subfamilies. 73% of families do not have children or subfamilies. In other words, for 73% of the announced families, the prefixes announced in BGP match the prefixes that were assigned, as shown in the IRR databases. No further sub-allocation was done by the end-user, either internally (i.e. using child prefixes), or externally (i.e. subfamilies).

Furthermore, there is no correlation between the number of children and the number of subfamilies. The Pearson correlation coefficient, as well as the Spearman correlation coefficient have values between 0.14 and 0.25, depending on if we include or not families without any child or subfamily. In other words, a lot of children implies neither few, and neither a lot of subfamilies; and vice versa. We

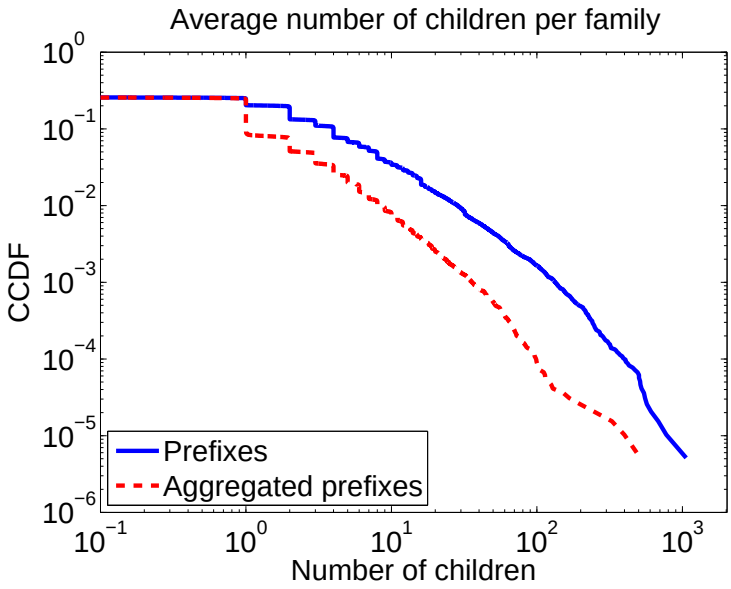


Figure 4.6: Number of children per family

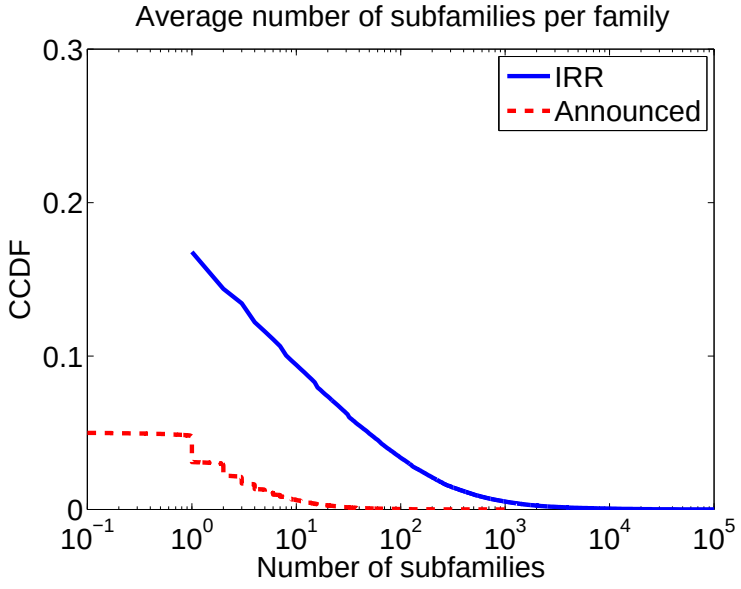


Figure 4.7: Number of subfamilies per family

further study these two dimensions (number of children, and number of subfamilies) in Section 4.4.4, where we present case studies.

Because we study the relationships between overlapping prefixes, we must limit our analysis to the 27% of families that *do* have children, subfamilies, or both (see Table 4.2). Consequently, the results presented in the remainder of this section only apply to these 27% of (announced) families.

Child	Subfamily	Count	%
N	N	141,883	72.96%
N	Y	1,930	0.99%
Y	N	42,734	21.96%
Y	Y	7,918	4.07%
Announced families		194,465	100%

Table 4.2: Announced families

We now focus on the fraction of IP space of the family father that was allocated to children or to subfamilies. Figure 4.8 plots the children overlap ratio and the family overlap ratio. It shows that there is no overlap by subfamilies for about 80% families. This is because, as indicated in Table 4.2, in most cases, there is no subfamily when there are children. In contrast, children can occupy a much larger fraction of the family father IP space, up to 100% in 45% of the cases. In other words, the owner of a prefix tends to allocate externally (i.e. to other institutions, leading to sub-families) in a conservative manner, while the allocation is much more generous when it is done internally (i.e. between the branches of a company).

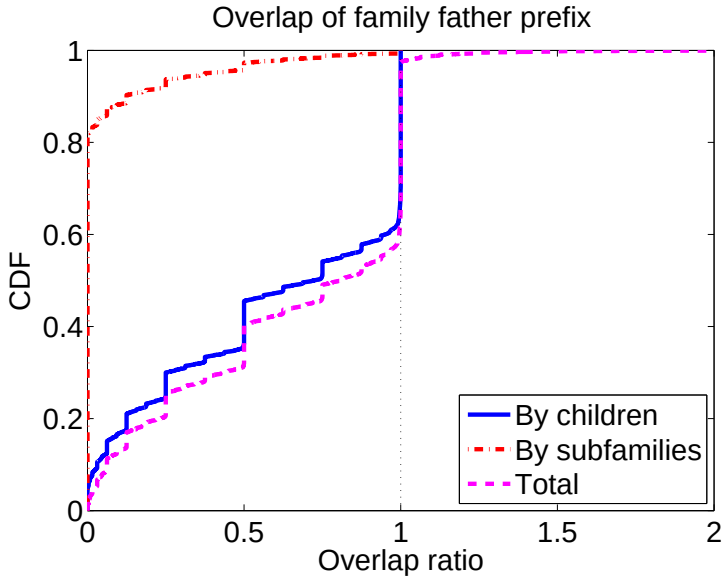


Figure 4.8: Prefix overlap within a family

Figure 4.8 also plots the sum of both of these ratios for the families. Interestingly, this ratio exceeds 1 for a few cases. Effectively, this means that, for about 3% of families, children prefixes and subfamily prefixes overlap the family father more than once. Consequently, they also overlap each other. An example from the real world for this situation is the following one. The IRR databases

list five prefixes: 5.102.0.0/19, 5.102.{0,8,16,24}.0/21. All these prefixes are also announced in BGP, plus two more: 5.102.{0,16}.0/20. As a result, the /19 family has four subfamilies that fully overlap the family father, and two children, which also fully overlap the family father. All prefixes are originated by a single AS, and belong to the same organization (a tier-3 ISP). It is worth noting that the time-weighted average values of these metrics were over 0.9 in both cases, indicating that this configuration was not transient.

4.4.3 AS-Level Topology

In this section, we study the AS-level topology of families and their children.

Figure 4.9 plots the number of origin ASes inside a family. 81% of families only have one origin AS, 10% have two, 3% have three, and the remaining ones have four or more. In other words, the vast majority of families are originated from a single AS. However, it is worth reminding that an AS is an abstract construct that is not necessarily bound to a single physical location. As a result, the networks behind the children prefixes might be distinct. For example, the prefixes originated by Cogent (AS174), which is a tier-1 ISP, end up in various countries around the world.

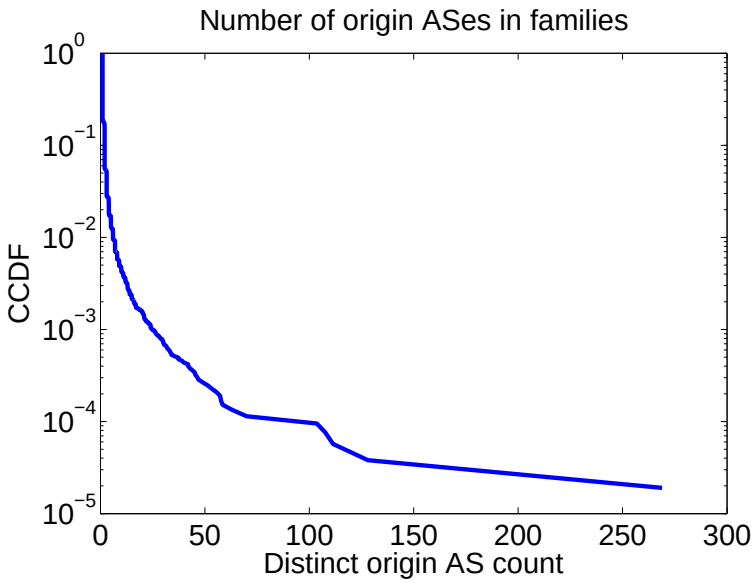


Figure 4.9: Number of origin AS per family

As mentioned in Section 4.3.4.2, the other topology metrics, which study the relative position of children prefixes with respect to the father, can only be computed if the family father is announced alongside its children; otherwise the comparison cannot be made. Figure 4.10 plots the duration of announcement of the family father over the duration of announcement of the family for families that have children prefixes. It shows that for about 42% of the families, the family father is announced during the whole observation period. Moreover, the global shape of the plot suggests that either the father is never there, or it is there all the time. Such a large contrast between the two possibilities is clearly the result of a policy choice. This means that, for about 58% of families with children, the BGP good practice of announcing the assigned prefix as-is is not met. For example, IBM has been assigned a (legacy) /8 prefix and announces a handful of prefixes through BGP, the largest of which

is a /16. In the case of the 58% of families where the father is not announced, only the children prefixes are announced in BGP, and the values of the topology metrics cannot be computed.

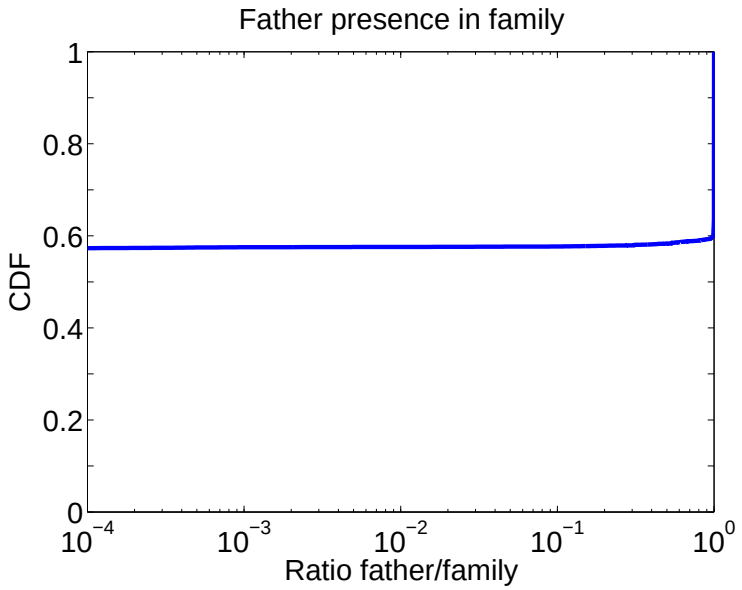


Figure 4.10: Ratio of father presence in the family, for families that have children

Table 4.3 breaks down the proportions of the 42% of families with children into the three possible AS topologies (which are also depicted in Figure 4.12). In 87% of cases (which globally amounts to 9.79% of all families), children are originated from the same origin AS as the family father; in 10% of cases, children are located behind the origin AS of the family father, whereas for the remaining cases, children are located in front. The last column indicates the topology proportion with respect to the total number of announced families (from Table 4.2).

Topology configuration	Number of families	%
Same origin AS	19,045	9.79%
Child behind father	2,140	1.10%
Child in front of father	640	0.33%

Table 4.3: Topology configurations

We now analyze each possible scenario in more details. Figure 4.11 plots the ratio of family children that have the same origin AS as the family father (full line), and the weighted version of the same metric (dashed line). It shows that, for nearly 70% of families, all children are originated from the same origin AS as the family father. Additionally, for more than 25% of the families, the children fully overlap the family father. For about 45% of families, more than 50% of the IP space is occupied by children. There are two possible AS topologies when a child has the same origin as its family father. Either the AS path for both prefixes is the same, in which case the configuration appears to be redundant from a BGP point of view (depicted in Figure 4.12(i), where only the top AS path would be seen); or the AS paths for both prefixes are different and the child is used in a different location as the father, such as for off-site servers (depicted in Figure 4.12(i)). According to [Huston 2014], the proportion of these in 2013 were respectively between 40% and 45% for the

same AS path, and between 20% and 25% for a distinct AS path.

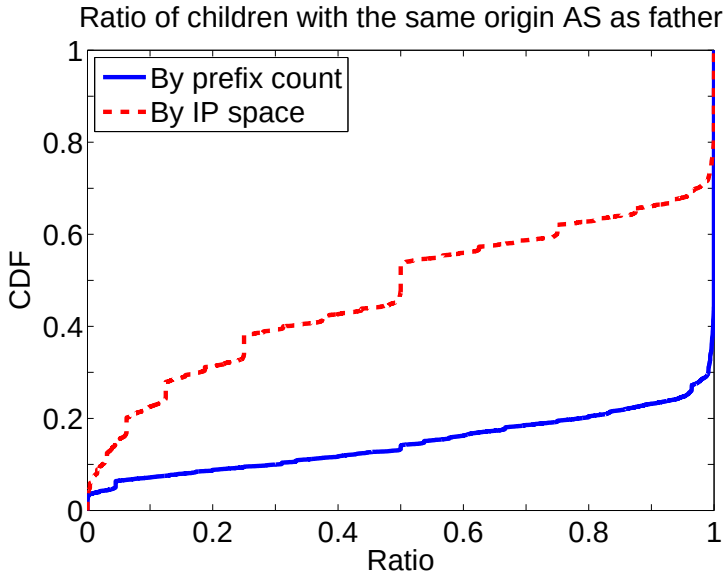


Figure 4.11: Ratio of family children that have the same origin AS as the family father

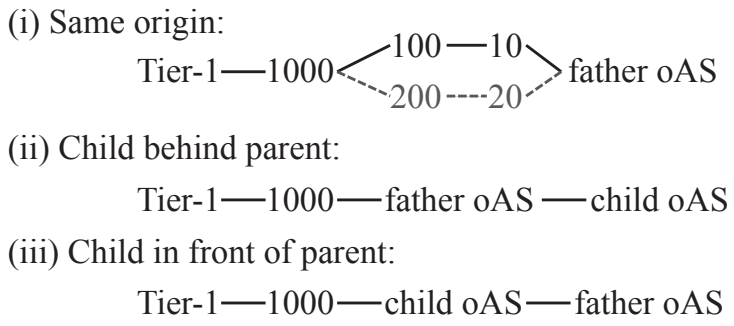


Figure 4.12: Possible AS paths topologies between a family father and its children

Figure 4.13 plots the ratio of family children that are behind the family father (full line), and the weighted version of the same metric (dashed line). This AS topology situation is depicted in Figure 4.12(ii). In 28% of cases, all children are behind the father; additionally, for a little less than 10% of the families, the children fully overlap the family father, and are located behind it. For about 25% of cases, more than 50% of the IP space is located behind the father. In the cases we investigated, this was the result of a tier-3/tier-2 ISP providing IP addresses and connectivity to customer (stub) networks that own their own AS number.

Figure 4.14 plots the ratio of family children that are in front of the family father (full line), and the weighted version of the same metric (dashed line). This configuration is depicted in Figure 4.12(iii). In 50% of cases, all children are located in front of the family father; additionally, in about 23% of cases, all children are located in front of the father, and completely overlap it. For about 50% of cases, over 50% of the IP space is located in front of the father. Our investigations showed two use cases for such a topological configuration. The first one is a situation in which an organization owns multiple AS numbers, i.e. sibling ASes [Gao 2001], and AS path is sometimes leaked through one of these origins and goes through the other. The other use case deals with cloud

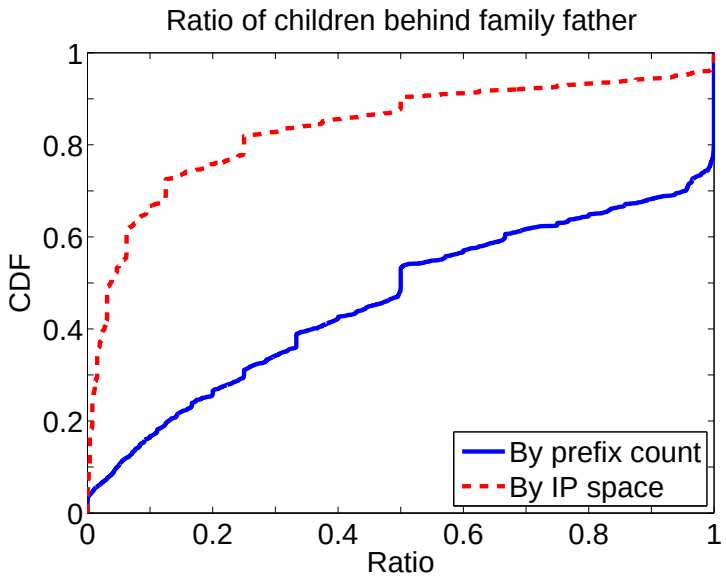


Figure 4.13: Ratio of family children that are behind the family father

service providers. Some providers offer anycast, failover services, or cloud hosting services, and allow the use of customer prefixes. The customer then chooses a subset of their network that will be used for the cloud service provider.

4.4.4 Real-World Case Studies

We consider in this section a few real-world cases to illustrate the typical relationship that can exist between the business of a company and the breakdown of its prefixes into subfamilies and children. We pick 21 companies – listed in Table 4.4 – that can be classified into three categories: tier-1, tier-3 and private corporations. The classification is approximate because companies acting as tier-1 providers can also run a tier-3 business at the same time, i.e. directly connecting end-users/small companies to the Internet. This is, for instance, the case of AT&T and Deutsche Telekom.

When looking at ISPs, and regardless of their size (i.e. tier-1 or tier-3), we observe a trend of having a large number of subfamilies and a comparatively smaller number of children. The sheer number of subfamilies suggests that ISPs routinely insert information about prefix delegation in the IRR databases. This is in line with expectations: ISPs typically offer Internet access to other companies, and thus assign a set of IPs to its clients. Doing so, the ISPs choose to push this information into the IRR databases, because it can be used for administrative purposes. We also observe again the trend outlined in Figure 4.2, namely that only a small fraction of these families are announced in BGP. This is because ISPs mostly provide Internet connectivity local businesses or home users, that would reap no benefit from the complexity and overload of running a BGP router.

For private corporations, the number of children is much higher than the number of subfamilies. We attribute this to corporations considering internal network policies as private information, thus not wanting to reveal additional company information it in the IRR databases (e.g. branch office location). We see two noticeable exceptions: Yandex and OVH. Yandex operates the largest search engine in Russia, along with a number of additional services (cloud storage, etc). The reason for

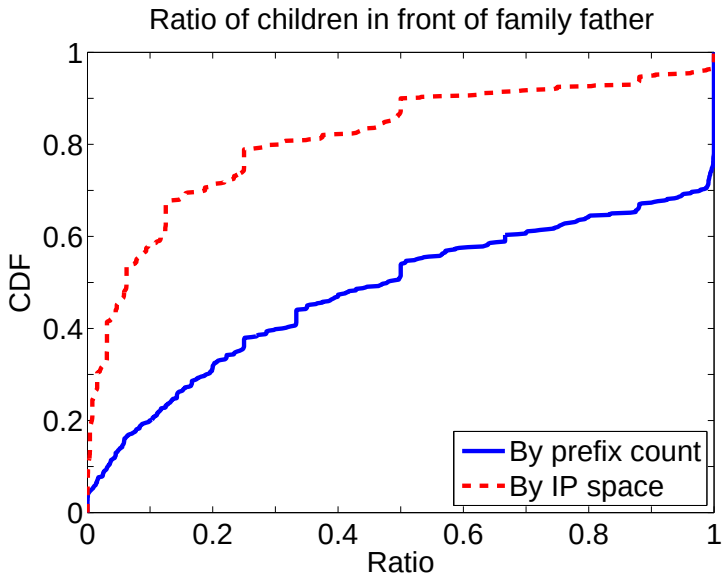


Figure 4.14: Ratio of children that are in front the family father

Business type	Name	#fathers	#children	#subfamilies
Tier-1 ISP	AT&T	64	363	582,863
	Cogent	39	87	1,416
	DeutscheTlkm	26	5	58,055
	NTT	152	466	2,744
	TeliaSonera	9	0	247
Tier-3 ISP	Belgacom	15	0	3,710
	Comcast	66	119	14,945
	Free	15	8	3,864
	Rogers	36	187	23,778
	Tele2	29	4	2,852
Private Corp.	Amazon	18	1	15
	Apple	2	196	1
	BBC	2	2	61
	DHL	2	21	0
	eBay	5	1	0
	HSBC	5	6	0
	Microsoft	40	86	3
	OVH	43	9	27,489
	Philips	8	0	0
	Sony	3	2	0
	Yandex	49	18	2,191

Table 4.4: Real-world case studies

the large number of subfamilies might be due to Yandex pushing up information concerning client companies (e.g. in the case of Web hosting service) in the IRRs. The case of OVH is easier to diagnose: OVH offers PaaS (Platform as a Service) and IaaS (Infrastructure as a Service) services, and reports in the IRR databases the set of addresses assigned to each clients, just like an ISP would do.

4.5 Validating Sub-MOAS Announcements

So far, we have studied the relationships between overlapping prefixes in the BGP routing table. We compared prefixes in BGP by grouping them in families of prefixes, according to the `inetnum` objects in the IRR databases. Because these objects indicate IP range assignments, it is unlikely that the announcements corresponding to these family fathers are the result of a subnet prefix hijacking attack. Therefore, the malicious sub-MOASes should be located within family children. Table 4.2 showed that only 26.03% of families had children. However, in order to be sub-MOAS, these children need to be originated from a distinct AS than the father's. Figure 4.9 showed that only 19% of families had more than one origin AS, i.e. that only 19% of families had sub-MOASes. As a result, the number of suspect sub-MOAS amounts to between 8000 and 9000, which is too high to suspect they are all the result of a malicious attack, and also tedious to investigate manually.

In the remainder of this Section, we present a prototype that validates these sub-MOAS announcements. We achieve this by extending our use of the IRR databases, and by searching for signs of a business relationship between the involved ASes. We use an AS-based topology filtering of sub-prefixes that takes advantage of the relationship between the origin of the sub-MOAS and the regular origin. Finally, we build an IPv4-wide dataset of SSL/TLS public keys that are running HTTPS services in order to infer if the network before and after the announcement of a sub-MOAS is identical. Please note that this verification is only possible for new sub-MOASes, not for existing ones.

Please note that the extended IRR analysis, as well as the SSL/TLS dataset based verification are possible thanks to our partners at the Technische Universität München (TUM).

4.5.1 Architecture

The architecture of our system is depicted in Figure 4.15. In a nutshell, given two origin ASes from the detected sub-MOASes (① in Figure 4.15), we look for a relationship in the RIPE IRR database (② in Figure 4.15). We also use a set of AS-topology heuristics (③ in Figure 4.15). Finally, we compare the public SSL/TLS key before and after the sub-MOAS announcement (④ in Figure 4.15). If this key is identical, we validate the announcement; otherwise, we do not. Please note that, for now, only the extended IRR analysis – which is available courtesy of our partners at the Technische Universität München (TUM) – for the RIPE version of the IRR database has been implemented. As a results, the validation of sub-MOAS is only partial (i.e. only for the RIPE region). However, the methodology that we present below is possible with each part of the IRRs, and work is being done on extending our system's coverage.

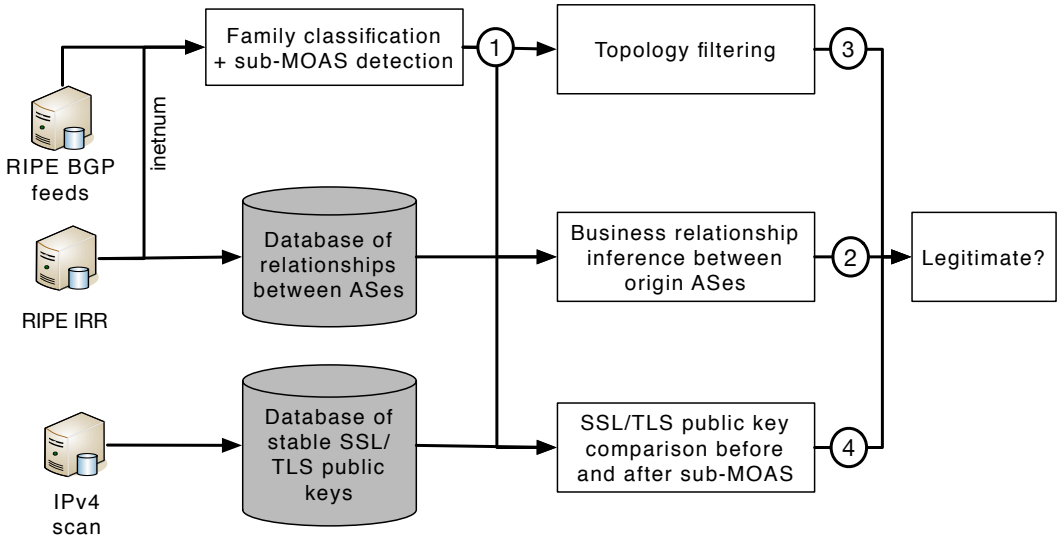


Figure 4.15: System architecture for validating sub-MOAS announcements

4.5.1.1 The RIPE IRR Database

The IRR databases contains many more objects than the `inetnum` objects we have used so far, in particular objects describing ASNs, people and organizations.

Structure

Figure 4.16 depicts the objects included in the RIPE database that we will use, as well as the cardinality of their relationship. [RIPEDB Docs] details all available objects in the RIPE database as well as their usage, attributes, and possible relations. We now details the objects depicted in Figure 4.16.

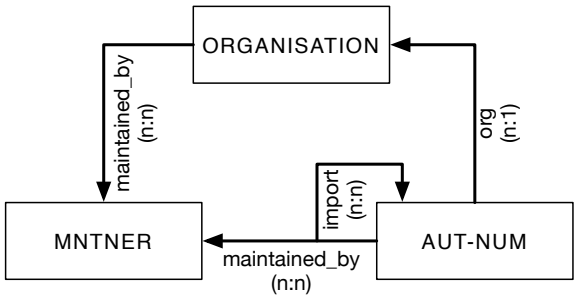


Figure 4.16: Objects within the RIPE IRR database and their relationships

The `mntner` objects models access rights and authorize creation, deletion, or modification of other objects. Objects that are linked to a `mntner` can only be modified by the owner of the authentication credentials behind the corresponding maintainer account. This link has now been made mandatory for all objects.

The `organisation` objects provide information about companies, non-profit groups, or individuals who hold an Internet resource. They mainly contain business contact information such as post

address, emails, phone numbers, etc. They are maintained directly by the RIPE NCC, and linking another object with an `organisation` required an explicit authorization from this organization.

Similarly, the `aut-num` objects represent the assignment of an AS number.

Tables 4.5 and 4.6 detail the number of objects and relationships parsed as of June 2014. These objects and relationships are at the core of our exploitation of the data inside the IRR database. We assume that an authorized party deliberately created the corresponding objects in the database, and that these relations are reliable. In order to avoid problems, we discard (and thus cannot validate the related sub-MOASes) incomplete, or ambiguous relationships. Please note that this exploitation is also possible thanks to our partners at TUM.

Object	Count
`mntner`	48,465
`organisation`	81,260
`aut-num`	27,616
`inetnum`	3,871,827
`route`	236,604

Table 4.5: Number of objects in our parsed version of the RIPE database

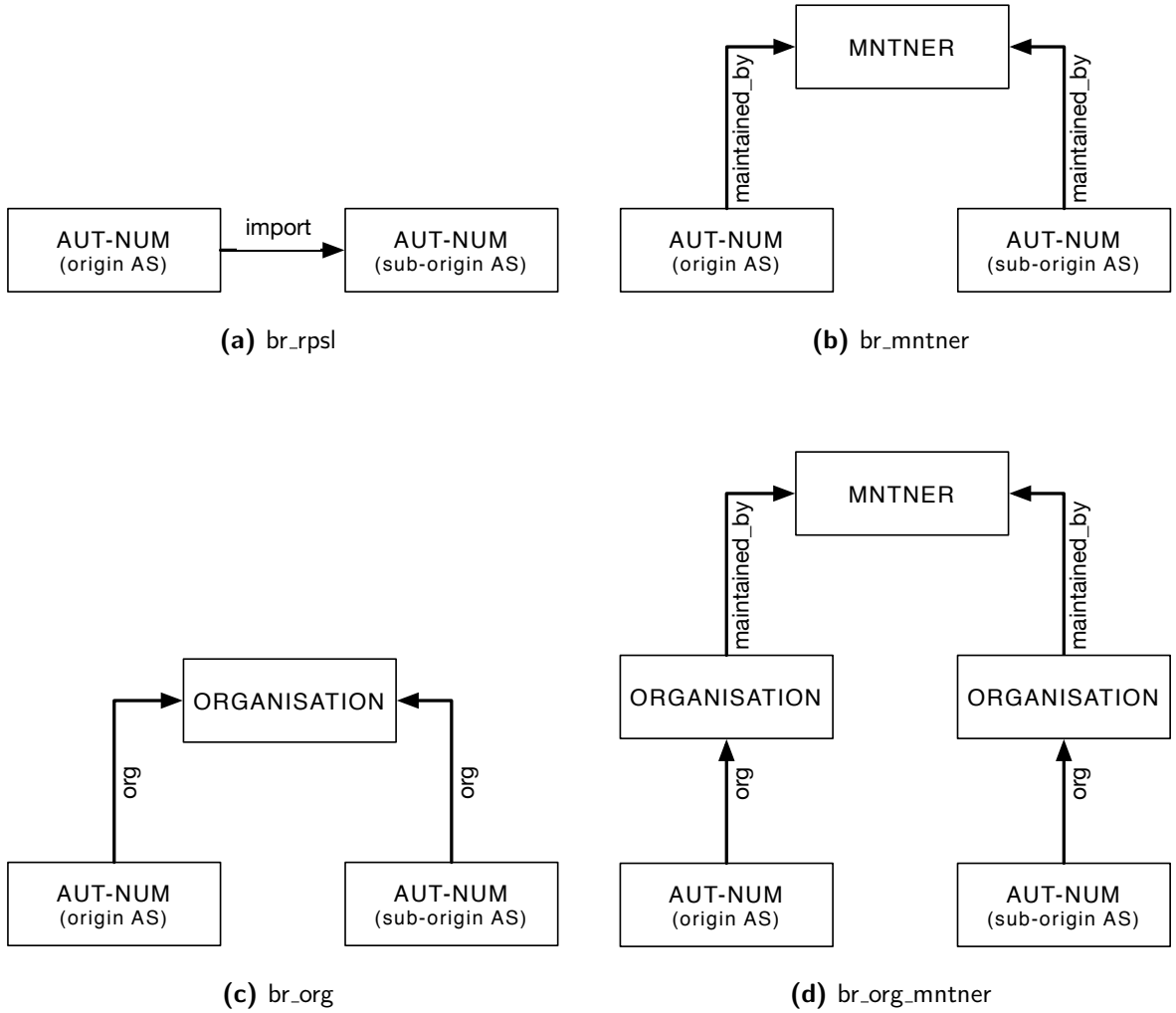
Relation	Count
maintained_by	5,307,883
org	199,644
import	221,690
origin	245,381

Table 4.6: Number of relationships in our parsed version of the RIPE database

Validating Sub-MOAS Announcements
We identify legitimate announcements by looking for evidence of a business relationship between the two origin ASes, i.e. if we can find one or more of the relations depicted in Figure 4.16 between the two origins. In other words, we consider that there is a business relationship between the origin AS of the less specific prefix and the origin AS of the sub-prefix if the origin AS deliberately updated the RIPE database to document its willingness to accept the sub-prefix's origin AS's routes (Figure 4.17a). Similarly, we infer a business relationship from these two ASes if share a common `mntner` object (Figure 4.17b), a common `organisation` object (Figure 4.17c), or from two different organisations that have the same `mntner` (Figure 4.17d). In all these cases, we validate the sub-MOAS announcement as resulting from business practices.

4.5.1.2 AS-Topology Based Filtering

In the first part of this Chapter, Table 4.3 illustrated that a number of children were located in a different AS than the family father; and that this AS was located upstream of the family father's origin AS (Figure 4.13), or downstream of the family father (Figure 4.14). As detailed in Section 4.4.3, if the sub-MOAS is located downstream of the father's origin AS, it is mostly due to a local ISP providing Internet connectivity through BGP means for a stub AS. When the sub-MOAS is located upstream of the family father, this can be due to sibling ASes or cloud service providers failover services.



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

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

Figure 4.17: Rules to infer a business relationship between two ASes

Let's consider these situations from a hijacking perspective. If the sub-MOAS is announced from an AS that is downstream of the family father's origin AS, the traffic destined to the sub-MOAS will need to go through the family father's AS before it reaches the sub-MOAS's origin AS. In the situation of a prefix hijacking, the family father's origin AS would most likely simply forward traffic internally to the corresponding machine, and not downstream to the hijacking AS. Moreover, it is reasonable to expect that an AS would not forward announcements hijacking its own IP space: this announcement would be filtered out. As a result, we discard the cases where the sub-MOAS origin AS is located behind the origin AS of the family father.

Unfortunately, we cannot draw any conclusion for the cases where the sub-MOAS is located in front of the origin AS of the family father because it can also correspond to the signature of a man-in-the-middle attack. In order to mark the announcement as benign, we would need to verify that the AS-level link between the sub-MOAS's origin and the family father origin is genuine. For this, we would need a technique to verify that the AS path has not been tampered with, and that the family father's origin AS number has not been maliciously inserted in the AS path in order to mask the attack.

4.5.1.3 Cryptographic Filtering

Our final filter uses data obtained from a regular IPv4-wide scan of the SSL/TLS public keys obtained from opening HTTPS connections. This dataset is available thanks to our partners in TUM. The goal is to legitimize sub-MOASes by verifying that a machine that was using a known SSL/TLS public key before the sub-MOAS announcement is still using it afterwards. Assuming that an attacker cannot get hold of the private SSL/TLS key associated with a machine, this implies that the same network is reachable with and without the sub-MOAS, consequently suggesting that the routing change is not the result of a hijacking.

Due to the fluctuating nature of the Internet, two subsequent scans are carried out in order to establish a ground truth. This first scan resulted in over 27 million IP addresses reachable on port TCP/443, for which the public key was saved. One month later, these IP addresses were rescanned. Unresponsive hosts, hosts for which the public key had changed, hosts with identical public keys, and hosts for which a sub-MOAS was introduced between (or during) the scans were discarded. There remained above 5 million hosts that we consider as *stable*, i.e. the IP address and the public key had remained unchanged.

4.5.2 Results

During our experiment, which lasted from June 2nd, 2014 to June 12th, 2014, we were able to "legitimize" 46.5% of all sub-MOAS events by applying our filters as depicted in Figure 4.15. Table 4.7 details the efficiently of each filter in the system. Recall that this validation is only based on the use of the RIPE version of the IRR database, including the use of `inetnum` to group the prefixes into families, as detailed in the first part of this Chapter. The RIPE IRR database only provides insights into around 60% of all detected MOAS cases. However, even with we are still able to legitimize around half of the sub-MOASes in the global routing table, this suggests that using similar data from other RIRs would yield even better results.

	%
All sub-MOASes	100.00
Families + IRR analysis	10.78
Topology filtering	31.72
SSL/TLS scans	22.93
Total legitimized sub-MOASes	46.53

Table 4.7: Efficiency of each sub-MOAS filter individually using the RIPE IRR database

Table 4.8 details the efficiency of the IRR rules depicted in Figure 4.17. The rules based on the `mntner` account and on the RPSL policy are, by far, the most efficient ones.

Finally, the SSL/TLS-public-key-based filter is able to legitimize over 85% of sub-MOAS events for networks in which there is at least one live host. This suggests that using additional popular Internet services running over SSL/TLS (such as IMAPS), or, by extension, any service for which we can capture the public key (such as SSH) can help further increase the accuracy of this filter.

	% (tot.)	% (RIPE DB)
All sub-MOASes	100.00	59.41
br_rpsl	20.52	34.54
br_mntner	29.41	49.52
br_org	2.89	4.87
br_org_mntner	8.22	13.84

Table 4.8: Efficiency of the IRR database rules

In the end, these results are quite encouraging given that we can achieve a high validation rate, even though we are missing authoritative data for over 40% of the IPv4 address space. It also illustrates that taking advantage of the AS-level topology of suspicious events can be useful to filter out a sizable amount of false positive. Finally, it shows how powerful active probing can be. Please note that our probing is lightweight for remote systems as it barely requires a full SSL/TLS handshake to be performed. On the other hand, there is a lot of insight to gained by gathering persistent and discriminant information about existing systems, and not by solely relying on volatile live-system responses, such as used by the existing methods in Chapter 2.

4.6 Summary and Conclusion

In the first part of this Chapter, we detailed how we use assignment data from the IRR databases as semantic anchor points in order to cluster prefixes from the BGP routing table into families, inside of which we can non-ambiguously study the overlap among these prefixes.

We showed that the IRR databases contain many times more prefixes than the BGP routing table. This is particularly true for prefixes with a mask length longer than 24. At the same time, we found that only 2.32% of the families induced by these IRR entries were effectively seen from BGP. We attribute this difference to the fact that IRR entries are not restricted to BGP players, but can exist due to any IP assignment. For example, there are single IP addresses (i.e. /32 prefixes) with an IRR entry for administrative reasons. Moreover, we interpret the large number of entries maintained by ISPs in the IRR databases as proof that the ISPs actively contribute to the IRRs and, hence, that the information contained in the IRRs should not be completely be dismissed. We plan to integrate active measurement techniques, such as reverse DNS lookups, in order to improve the quality of the families we build, thus circumventing possible problems due to heavy reliance on IRR data.

We showed that 74% of the announced families do not have children. This means that, for these families, only the prefix that was assigned is announced in BGP, which does not lead to (additional) routing table entries. It is also in accordance with the standard BGP good-practice of always announcing the assigned prefix. For about 15% of all families – but about half of the families with children or subfamilies – this practice is not met. Moreover, we showed that, whether or not the assigned prefix is announced is the result of a policy choice by the prefix owner, and not due to transient side-effects, such as a low route visibility.

For 81% of families with children, the children are originated from the same AS as the family father. This does not imply that these children are without purpose. Because an AS is an abstract

construct, two prefixes originated from the same AS can be hosted at completely different IP endpoints. Nevertheless, we observed that, in some cases, the overlap ratio is greater than one, which implies that every address behind the family father prefix is routed multiple times by different prefixes. Since only the most specific prefix will be used in the end, the other flavours of the announcement are of no use. When the origin AS of a child and the one of the family father are different, the child can be located behind the father, in which case the child is a sub-allocation of IP space that uses the father network as transit. If the child is located in front of the father, the configuration can be used as a DDoS mitigation technique, and the child effectively protects machines hosted within the father network.

A key take-away from our study is that a joint analysis of BGP and the IRR databases sheds light on the way the IRR is used, and also enables to uncover different types of business practices. For instance ISPs (large, or small) are more likely to register their customer in the IRR databases, leading to a greater number of subfamilies than children. Clients of ISPs being, most of the time, relatively small, the most popular flavour of subfamily is a prefix of mask length 29, which constitutes enough addresses for a small business. Private corporations (whose business is not related to IT services), are more likely to consider their global network structure as proprietary, and thus not divulge sub-allocation information into the IRRs. This results in far less subfamilies than for an ISP.

In the second part of this Chapter, we presented a prototype able to validate legitimate sub-MOAS announcements by combining three distinct filters on the origin conflicts inside prefix families. The first filter extends our usage of the IRR databases by taking advantage of the relationships between the different objects inside the RIPE IRR database. We defined a set of rules from which we are able to infer a business relationship between the origin AS of the family father and the origin AS of the sub-prefix. The second filter uses an AS topology based heuristic for discarding sub-MOASes, based on the AS-level relationship between the two origins. The third filter uses an active probing of the whole IPv4 address space to localize a set of persistent and stable IP addresses that are responding to HTTPS requests. By comparing the public key for an IP address before and after the sub-MOAS, we can infer if the routing change is due to a subnet hijacking attack, or to a benign engineering choice. The current prototype is able to validate almost half of all sub-MOAS events, even though it is only making use of the RIPE IRR database. We believe that by extending this framework, we can sufficiently reduce the search space for sub-prefix hijacking to a manageable size that can be investigated manually, and, eventually, lead to the creation of other subsequent filters.

Publications

The material presented in this chapter led to the following publications:

- Johann Schlamp; Ralph Holz; Oliver Gasser; Andreas Korsten; Quentin Jacquemart; Georg Carle; and Ernst Biersack.
  "Investigating the nature of routing anomalies: closing in on subprefix hijacking attacks".
  In 7th International Workshop on Traffic Monitoring and Analysis (TMA 2015), April 2015.

- Quentin Jacquemart; Guillaume Urvoy-Keller; and Ernst Biersack.
  "Behind IP Prefix Overlaps in the BGP Routing Table".
  In 17th International Passive and Active Measurements (PAM 2016) Conference, March 2016.

- Johann Schlamp; Georg Carle; Ralph Holz; Quentin Jacquemart; and Ernst Biersack.
  "HEAP: Reliable Assessment of IP Subprefix Hijacking Attacks".

In IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Measuring and Troubleshooting the Internet: Algorithms, Tools, and Applications, 2016.

The IP Blackspace

5

In Chapter 4, we focused on prefix overlaps in the BGP routing table, and we grouped BGP prefixes based on allocation information that we extracted from the IRRs (Internet Routing Registries); thereby clustering the prefixes into families. In this Chapter, we focus on the situation in which a BGP prefix cannot be inserted into a family because no allocation information is available. This situation can arise because a small part of the IPv4 address space has still not been assigned for use to any organization. However, some of this IP space is announced through BGP, and is, therefore, globally reachable. These prefixes, which are a subset of the *bogon* prefixes, constitute what we call the **blackspace**. It is generally admitted that the blackspace stands to be abused by anybody who wishes to carry out borderline and/or illegal activities without being traced.

The contribution of this Chapter is twofold. First, we propose a novel methodology to accurately identify the IP blackspace. Based on data collected over a period of seven months, we study the routing-level characteristics of these networks and identify some benign reasons why these networks are announced on the Internet. Second, we focus on the security threat associated with these hijacked blackspace address blocks by looking at their application-level footprint. We identify live IP addresses and leverage them to fingerprint services running in these networks. Using this data we uncover a large amount of spam and scam activities. Finally, we present a case study of confirmed fraudulent routing of IP blackspace.

5.1 Introduction

The global BGP routing table now contains over 600k distinct IPv4 prefixes, including a few *bogons*: prefixes that should not be globally announced, such as the private IP space. A subset of bogon prefixes, which we call the *blackspace*, is composed only of prefixes that have not been assigned for use to any organization.

These unallocated, yet globally announced and reachable blackspace prefixes traditionally hold a bad reputation. On top of uselessly cluttering up the global routing table, there have been reports of DDoS (Distributed Denial of Service) attacks originated from blackspace address blocks [Thomas

2001]. Spammers are also believed to abuse the blackspace in order to stealthily announce and abuse routes [Feamster *et al.* 2004]. By extension, it is admitted that the blackspace stands to be abused by anybody who wishes to carry out borderline and/or illegal activities without being traced.

Because it is unallocated, hijacking a blackspace prefix is more likely to go unnoticed. Indeed, there is no victim network whose NOC (Network Operations Centre) or CERT (Computer Emergency Response Team) will notice the unusual announcements or the related traffic side effects. Moreover, the traditional hijacking detection tools, such as Argus [Shi *et al.* 2012], use signatures that are triggered by a network being simultaneously announced in different ways. In the case of blackspace prefixes, there is no rightful owner, and thus no default announcement on which to depend upon.

Consequently, it is recommended to filter out bogons (including the blackspace), so as to minimize the window of opportunity of potential abusers. Unfortunately, the blackspace constantly varies in size and shape, according to new prefix assignments and prefix returns that are carried out daily by different Internet actors. Filtering out bogons is therefore inconvenient and tricky. In order to automate the process as much as possible, Team Cymru's [Bogon Reference] provides multiple lists with different levels of granularity that can be included directly in a BGP router's configuration.

This Chapter focuses on the study of blackspace prefixes and aims to clarify what the blackspace contains. A partly similar study, which encompassed all bogon prefixes [Feamster *et al.* 2004], was carried out over 10 years ago. The formal reporting of malicious events carried out from the blackspace, [Thomas 2001], is even older. Back then, the IPv4 landscape was much different from today's, and the results provided by these works are not applicable anymore in today's Internet.

We first by detail the method that we use to isolate the blackspace prefixes from the BGP routing table. We then provide a thorough study of the blackspace networks on two different levels. First, we look at the information we extract from the BGP control plane and study the size of the blackspace. We then study the persistence and change in the blackspace through time. We characterize the origin ASes (Autonomous Systems) that actively announce blackspace by actively using semantic information (e.g. WHOIS records). Second, we look at the data plane and focus exclusively on the security threat associated with the blackspace prefixes. In order to do so, we actively seek live IP addresses and extract the domain name for these machines. We check the websites running in the blackspace, analyze their content, and check if their URLs are known to be malicious. We use an IP blacklist to locate hosts that are associated with adware, scam, phishing, and other malicious activities. Finally, we check for spamming activities and show how some spammers skillfully abuse the unallocated IP space in order to remain anonymous.

This Chapter is organized in the following way. Section 5.2 details the method and the datasets we use in order to locate the blackspace inside the BGP routing table. Section 5.3 details our analysis results: Section 5.3.1 study the size and variation of the blackspace; Section 5.3.2 details the BGP topology characteristics of the blackspace prefixes; Section 5.3.3 details the active measurements we do on blackspace networks, as well as a detailed threat analysis. Finally, Section 5.4 addresses the shortcomings of our approach.

5.2 Isolating the Blackspace

In this Section, we detail how we isolate the blackspace prefixes within the global BGP routing table by using a combination of distinct datasets that provide information about IP assignments. This step is necessary because there is no information on how the current [Bogon Reference] list is

populated. We show later, in Section 5.5, that our methodology for identifying the IP blackspace is more accurate and finer grained than previous efforts.

5.2.1 IP Space Assignation Hierarchy

To better understand our methodology, it is perhaps best to first briefly mention how the IP address space is divided into multiple blocks by distinct institutions before being assigned to end users, such as ISPs, corporations, or academic institutions. First, the IANA (Internet Assigned Numbers Authority) is in charge of distributing /8 prefixes to RIRs (Regional Internet Registries). There are 5 RIRs, each responsible for a different geographical area. In turn, RIRs allocate IP address space to LIRs (Local Internet Registries), such as ISPs, large corporations, academic institutions, etc. LIRs enforce their RIR's policies and distribute IP address blocks at the local level, i.e. to end users [Address Management Hierarchy; RIPE LIRs FAQ].

As a side note, Chapter 5 provide inner details in the way in which LIRs allocate IP address space to end-users, and the way these end-users use this address space (which may include further sub-allocations).

5.2.2 Definitions

Bogon prefixes have traditionally been loosely defined as any IP prefix in the BGP routing table that should not be globally reachable. More precisely, following the definitions of the [Bogon Reference], a prefix is a **bogon** if any of the three following conditions is true:

1. it is a **martian** prefix, i.e. if it is a prefix that was reserved for special use by an RFC, such as the private IP address space;
2. the prefix belongs to a block that was not assigned to any RIR by the IANA;
3. the prefix belongs to a block that was not assigned by a RIR to a LIR, or to an end user.

We define the **blackspace** prefixes as the *set of bogon prefixes that are not martians and that are announced in BGP*. In other words, it is the set of BGP-announced prefixes that have not been assigned for use – either because it still belongs to the IANA pool, or because a RIR has not assigned it to an ISP or an end user. The reason we explicitly remove martian prefixes is because they are most likely the result of a local route leak caused by a misconfiguration [Feamster *et al.* 2004]. Moreover, since these prefixes are internally routed in a lot of networks, we are unlikely to reach martian-originating networks from our own, rendering any standard network diagnostics utility such as `ping` or `traceroute` pointless.

5.2.3 Internet Routing Registries

The IRRs (Internet Routing Registries) are a set of distributed databases maintained by the five RIRs where network operators can provide information regarding their network on a voluntary basis. In particular, the `inetnum` objects contain information regarding IP address space assignment [APNIC Whois Guide]. Consequently, the IRR databases sound like the ideal starting point to isolate the IP blackspace. We need to access the database of each RIR, and extract the IP ranges mentioned in

`inetnum` objects. We then have to check the prefixes announced in BGP against the ones we found in the IRRs, and keep those that do not match.

Unfortunately, things are not quite that simple. Like previously stated, providing information in the IRR databases is in no way mandatory, and even though it is considered as a good practice for LIRs to maintain their allocation information up to date, they are in no way required to do so. Additionally (and somehow consequently), the IRR databases are manually updated, and thus are plagued with typical human errors, such as typos. For example, some `inetnum` objects end their network on a .2**2**5 IP address, where the right value would be 2**5**5; some objects explicitly discard their net address, and/or their broadcast address... Due to these reasons, we cannot expect to have an exact mapping between the BGP prefixes and the IRR prefixes. As a result, if we cannot match a BGP prefix to an IRR prefix, we take into consideration `inetnum` objects that are within the BGP prefix (i.e. `inetnum` objects that are more specific than the BGP prefix). If over 95% of the address space of the BGP prefix is covered by more specific IRR prefixes, we consider the BGP prefix has having been assigned, and that providing a matching IRR entry was overlooked. Our reasoning is that each customer of LIRs (which may be other ISPs) potentially wishes to update the IRR database, if only to update the management information of their network, such as technical and administrative contact details.

5.2.4 RIR Statistics Files

Every day, each RIR publishes a report – sometimes known as the delegation report – on the current status of the use they make of resources they have been allocated, including IP address space [ARIN 2012]. This report breaks down each RIR's IP address pool into four distinct states: ALLOCATED, ASSIGNED, AVAILABLE, and RESERVED. The first two states, ALLOCATED and ASSIGNED, are similar in the fact that they both have been marked as usable by someone by the RIR, i.e. these addresses can be announced. The difference is that ALLOCATED space ought to be used by LIRs for suballocation, whereas ASSIGNED space should not – i.e. it should be used directly by the LIR or end user. As the name suggests, the AVAILABLE state contains addresses that have not been ALLOCATED or ASSIGNED to any entity. Finally, the RESERVED state is somehow an intermediate between the other states: it has not been ALLOCATED (or ASSIGNED) to anybody, but is also not AVAILABLE for such purposes. For example, these addresses might be reserved for the growth of a LIR, returns that have not been cleared yet, or experimental space [ARIN 2012]. In this classification, the blackspace is shared between RESERVED and AVAILABLE states: in both cases there should not be any public BGP announcement for these addresses.

5.2.5 Blackspace Computation Process

Our BGP dataset is built on the data provided by the [RIPE RIS] collectors. We daily fetch the routing table of each of the 13 active, geographically diverse routers, and create a list of all globally reachable routes. In the same time, we daily extract all `inetnum` objects from each IRR database, and we compare these two datasets as described in Section 5.2.3. We then remove from the remaining BGP prefixes the parts for which there exists an IRR entry. For illustrative purposes, let's consider (a real-world case) where a /21 prefix is announced in BGP, and where only one of the /22 more specific prefixes has an `inetnum` entry. We remove the /22 that is in the IRR from the blackspace, leaving only the other /22 in it. At this point, there is a one-to-n relationship between the prefixes in the blackspace and the prefixes as announced in BGP: a single BGP-announced prefix can result in multiple entries in the blackspace once the registered parts have been removed.

We further filter the results by discarding prefixes that are marked as ASSIGNED or ALLOCATED by RIRs in their statistics files. Once more, there are cases in which the remaining prefixes are in multiple states with respect to the statistics files states, e.g. the IP space is ALLOCATED and RESERVED. In this situation, we only keep the part of address space that is either RESERVED or AVAILABLE.

It is noteworthy that, although using both the IRRs and the statistics files might appear redundant, there are documented inconsistencies between the two distinct datasets [RADI]. Because we aim at investigating the blackspace, it is essential to use these multiple sources in order to circumvent the limitations inherent to each dataset and to focus exclusively on real blackspace prefixes so as to avoid introducing bias in our results.

5.3 Blackspace Analysis

In this Section, we study the blackspace networks over a period of seven months, between September 2014 and March 2015. In Section 5.3.1 and Section 5.3.2, we consider the routing-level characteristics of the blackspace networks, and identify some patterns for legitimate blackspace announcements. Then, in Section 5.3.3, we seek to determine the security threat posed by the blackspace networks by looking at the application-level services running in these networks, and by checking whether they were involved in some malicious activities like spamming or scam website hosting. Finally we provide a case study of a confirmed case of cybercriminals who carried out nefarious activities such as spamming by abusing AVAILABLE IP space.

5.3.1 Prevalence and Persistence

In this Section, we focus on a few essential aspects of the blackspace by looking at the size, temporal characteristics, and variation of the blackspace. In order to observe those, we computed the blackspace once per day between September 1st, 2014 and March 31, 2015 with the method detailed in Section 5.2. The reason we compute the blackspace once a day is because the IRR databases we use and the RIR statistic files are only updated with this same frequency.

During our observation, the number of globally distinct prefixes from our collector routers varied between 550k and 600k prefixes. These prefixes route around 180 equivalent /8 IP addresses, i.e. the equivalent of 180 class A networks, or 180×2^{24} IP addresses. The reason we focus on the number of IP addresses instead of the number of prefixes is that, because of the methodology explained in Section 5.2, the relationship between a BGP prefix and a blackspace prefix is a one-to-many. By taking an aggregated BGP prefix and removing parts of it, we virtually inflate the number of prefixes in the blackspace, even though this larger number of prefixes actually represents a smaller IP space, rendering the prefix count meaningless. Figure 5.1 plots the daily number of IP addresses in the blackspace, as seen from a global BGP point of view. It shows that the blackspace size normally varies between between 10^{-2} and 10^{-1} eqv. /8. It also shows that this number is relatively stable, apart from two peaks in October 2014 and January 2015. We investigated the reasons behind these peaks and attributed them to the leak of 192.0.0.0/2 between October 15, 2014 and October 20, 2014; and a series of smaller prefixes between January 24, 2015 and January 29, 2015. We classify these events as routing leaks because of the low visibility we get for these routes. Only 3 collector routers received the a route for 192.0.0.0/2 in October, and only 1 received the multiple prefixes in January 2015. Moreover, in both cases, only a single AS path was seen, and the origin AS was a

private AS number. All in all, Figure 5.1 shows that the entirety of the blackspace could generally be contained in a single prefix, whose CIDR length would be between a /10 and a /15.

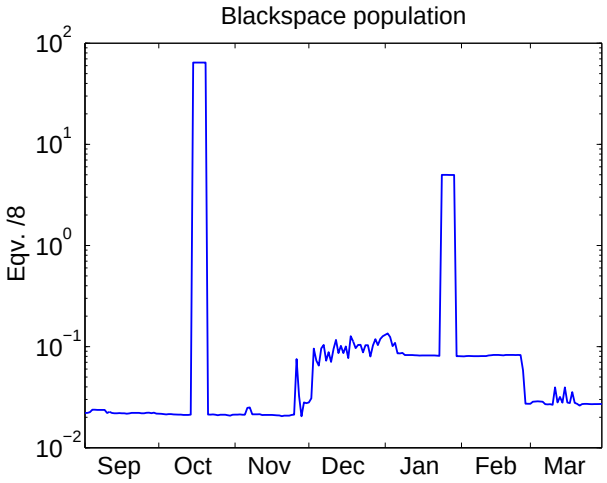


Figure 5.1: Number of IP addresses in the blackspace, between September 1st, 2014 and March 31, 2015

As mentioned in Section 5.2, a prefix in the blackspace has no `inetnum` entry in the IRR, and has not been allocated for use by a RIR. Figure 5.2 breaks down the statuses attributed to these IP addresses. Route leaks excluded, most of the blackspace is actually due to RESERVED resources, which are set aside by RIRs because they cannot be allocated right away.

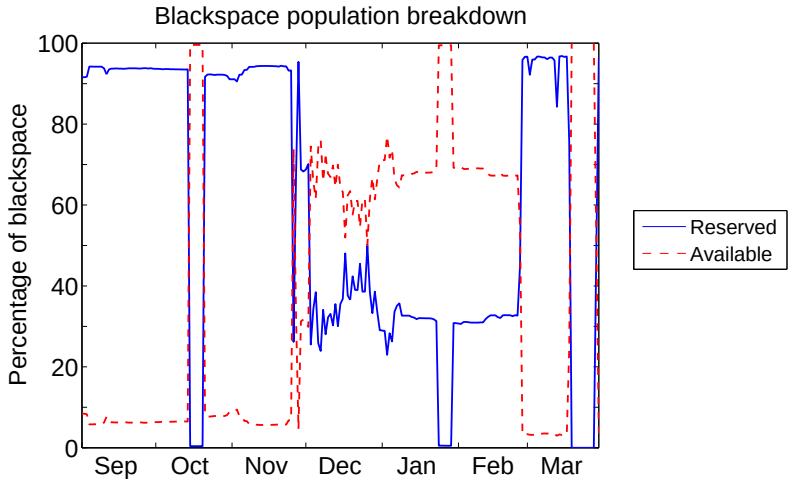


Figure 5.2: Daily proportion of RESERVED and AVAILABLE address space in the blackspace, between September 1st, 2014 and March 31, 2015

Figure 5.3 plots the number of consecutive days a single prefix was included in the blackspace. The full line plots this duration for all blackspace prefixes, including the many transient ones that were the results of the two route leaks already observed in Figure 5.1. The dashed line plots the same duration, but excludes the prefixes resulting from these leaks. The difference between these two curves implies that a lot of distinct prefixes were added to the blackspace due to the leak of routes. Indeed, the regular CDF shows that most blackspace prefixes are detected during 4 or 5 consecutive

days, which is precisely the duration of the two leaks observed in Figure 5.1. On the other hand, the dotted CDF shows that 50% of blackspace prefixes that are not the result of these leaks are seen for at least 12 days, and that around 28% of them are seen during 1 day or less. Figure 5.4 shows variation of the blackspace by plotting the Jaccard index in-between two successive days. We compute the Jaccard index between days d and $d+1$ as the ratio of the number of blackspace prefixes that are detected on both days, divided by the total number of distinct blackspace prefixes detected on day d and $d+1$. A Jaccard index value of 1 indicates that the computed blackspaces for days d and $d+1$ are identical. Conversely, a Jaccard index value of 0 indicates that the computed blackspaces for days d and $d+1$ are 100% different. The closer to 1 the value is, the more similar the two blackspaces are. Once again, the variation is quite high when the route leaks start and finish, as shown by the full line; but there is not a lot of daily variation otherwise (as shown by the dashed curve).

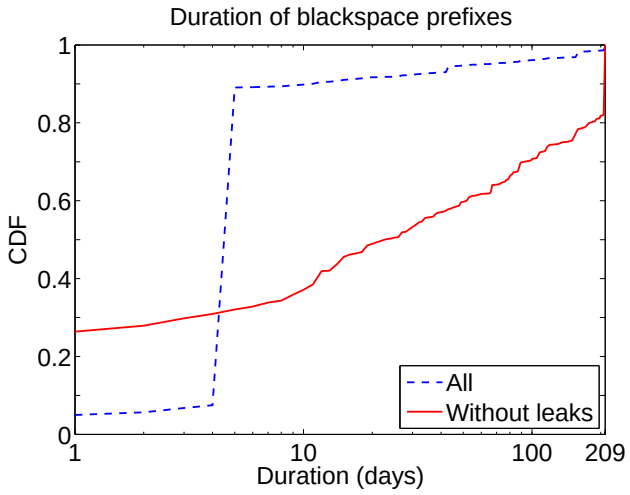


Figure 5.3: Persistence of blackspace prefixes

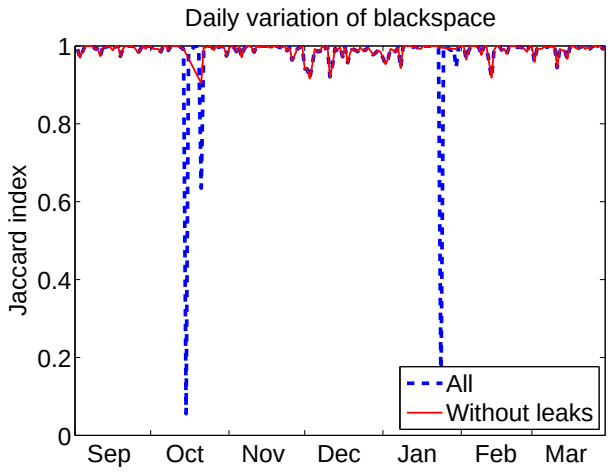


Figure 5.4: Day-to-day variation of the blackspace prefixes

The duration of a prefix in the blackspace (Figure 5.3) as well as the variation of the blackspace (Figure 5.4) imply that some prefixes leave the blackspace. This is possible if any of the three

following conditions are met:

1. the prefix is withdrawn from BGP;
2. an `inetnum` entry is added in the IRR;
3. the prefix is marked as ALLOCATED or ASSIGNED by a RIR.

Figure 5.5 plots the distribution of each event for prefixes that exited the blackspace during our observation period. Again, the values are plotted for all entries, and also only for entries that were not the result of route leaks. In both situations, the most likely cause is that the prefix has been withdrawn. The second cause is the creation of an `inetnum` entry in an IRR database. If the IRR entry is more specific than the blackspace prefix, another (more) specific prefix will be included in the blackspace instead. Consequently, a bit less than 45% of prefixes leave the blackspace because the BGP announcement was withdrawn. On the other hand, the other 55% become allocated (in one way or another) afterwards; which implies that half of the prefixes included in the blackspace are, potentially, used in good faith by the announcers. However, the other half, which globally amounts to a /11 network, does not end up as a registered network.

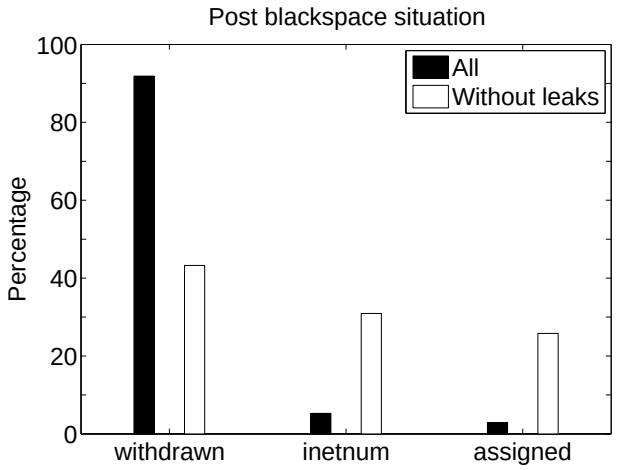


Figure 5.5: Situation of the prefix after it left the blackspace

5.3.2 BGP Characterization

In the previous Section, we saw that there are many blackspace prefixes, many of which are long-lasting. In this Section, we focus on the BGP characteristics of blackspace prefixes. We first focus on the origin AS of the blackspace prefixes to shed light on their uses. Where we can't, we look at the temporal evolution of the blackspace prefix along with its origin AS in order to better understand the root cause.

AS numbers are assigned a status by RIRs, just like IP blocks (see Section 5.2): either ALLOCATED, ASSIGNED, AVAILABLE or RESERVED. Figure 5.6 plots the daily proportion of each AS status for ASes that originate a blackspace prefix. The plot has been further broken down by explicitly classifying the private AS numbers (between 64512 and 65535 [RFC6996]) separately from the RESERVED set. As can be seen by the black/squared line private ASNs are responsible for a large

number of prefixes, but only during the two route leaks. In fact, all leaked prefixes are originated from a private ASN. ALLOCATED, ASSIGNED and RESERVED ASNs all roughly account for a third of blackspace prefixes, and AVAILABLE ASNs account for less than 10% of those. Just like with IP blocks, RESERVED and AVAILABLE ASNs are not ALLOCATED, and thus should not be in use. Yet, two thirds of the blackspace prefixes are originated by these ASes.

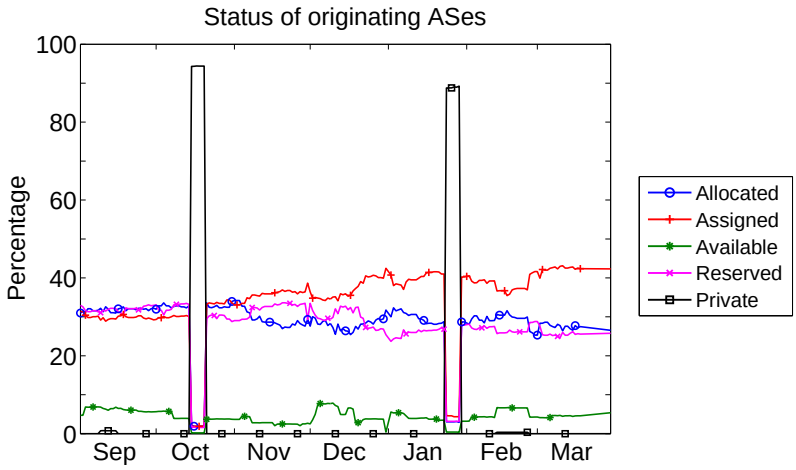


Figure 5.6: Daily status of the ASNs originating a blackspace prefix

Figure 5.7 plots the percentage of blackspace prefixes for ASes that announce (at least) one blackspace prefix. The plot is further subdivided by AS status, but we excluded the private AS numbers, as they were the result of route leaks (see Figure 5.6). Here, both of the ALLOCATED and ASSIGNED statuses behave similarly, with more than 90% of them announcing less than 1% of blackspace prefixes. Less than 10% of ALLOCATED (and around 20% of ASSIGNED) ASes originate more than a quarter of blackspace prefixes. On the other hand, close to 70% of RESERVED and AVAILABLE ASes *only* announce blackspace prefixes. To put this into perspective, the (global) average number of announced prefixes by ALLOCATED ASes is 229; by ASSIGNED ASes is 340; by RESERVED ASes is 4, and by AVAILABLE ASes is 2. In order to find out who operates these networks, we look at the names of the corporations behind these ASes (using [Potaroo: Autnums]). We get 185 network names for ALLOCATED or ASSIGNED ASes that originate blackspace prefixes, for which we located the corporation website using mostly popular web search engines. We were able to resolve 178 names to mostly telephone or cable companies and ISPs (of all sizes and shape: tier-1 to tier-3, from dial-up to business-grade fiber providers, all around the world), hosting and cloud providers, data centers, IT service companies, and world-wide tech companies. Other companies operated as advertising, airlines, bank and insurances, constructions, courier and parcel delivery services, e-commerce, Internet exchange points, law firms, medical companies, military contractors, and online news. We could not resolve 7 names. One was established as a company, but the website did not work, one used a name too generic to be found, and for three we could not locate any further information. The two remaining ASes appear to have been registered by individuals in Eastern Europe who also own other ASNs which are known to send spam – but do not originate blackspace prefixes at the same time.

Because the RESERVED and AVAILABLE ASes are not registered, we were not able to find registration information for them. Instead, we looked at the BGP topology of these prefixes, and investigated on the evolution of the blackspace prefix through time. For 33% of the cases where a blackspace prefix is originated from a RESERVED AS, the origin AS remains RESERVED throughout

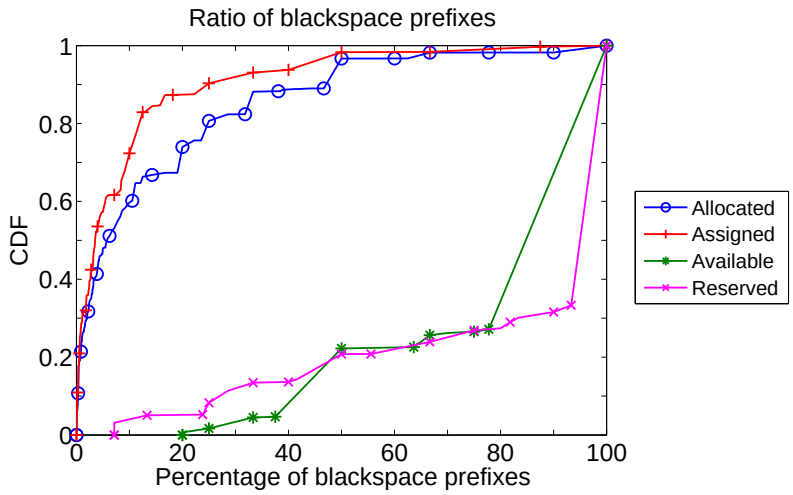


Figure 5.7: Percentage of blackspace prefixes originated by ASes according to that AS's status

the whole observation period. The prefixes were marked as RESERVED. These networks are usually single-homed and peer either directly with a tier-1 provider, or with a tier-3. The other 66% prefixes show a state transition from, or to, RESERVED. In all the cases we observed, this was due to a network owner either bringing up a new network, or decommissioning an old one. For example, half a dozen blackspace prefixes were originated from a RESERVED AS for 6 months through a tier-1 AS. On one day, the AS status changed to ASSIGNED and the name matched a well-known airline. The next day, the prefixes were all given inetnum entries in the IRR. Our interpretation is that the prefixes and ASN were RESERVED for the growth of said airline, and that they started using these resources before the paperwork had been fully processed. In another case, the prefixes and ASN were ALLOCATED, but one day turned to RESERVED. By looking up the company's name, we were able to find a letter from ICANN, informing the company that they had breached their registrar accreditation agreement by failing to meet technical requirements, and also by failing to pay the accreditation fees. The day following the date of the letter, all of that company's resource where changed to RESERVED. In some cases, there are transitions from ALLOCATED, to RESERVED, and then back to ALLOCATED. In this situation, we believe the situation was similar to the one of the last example, except that they corrected their behaviour to meet the requirements during the grace period. In the case of AVAILABLE ASes, there were only a handful of situations in which the AS (and the announced blackspace prefix) ended up as ALLOCATED or ASSIGNED. In these situations, it was the result of a new network being connected to the global Internet.

In conclusion, by looking at the routing-level characteristics, we were able to identify a set of blackspace prefixes that appear to be benign. Some prefixes appear to be in the blackspace because they have just been allocated, or because they are being phased out. Moreover, some blackspace networks are originated by tier-1 ISPs. Consequently, these networks are unlikely to be maliciously announced. All other networks need to be further analyzed in order to assess their threat level.

5.3.3 Data Plane and Application-Level Analysis

In the previous Section, we looked at the BGP characteristics of blackspace prefixes and were able to identify some cases in which these prefixes were announced for benign reasons. In other cases, we need to further analyze the networks in order to know if they are announced with malicious intent.

To carry out this analysis, in this Section, we focus on uncovering the application-level services running in the blackspace and look for hosts associated with malicious network activities.

5.3.3.1 Introduction

In the previous Sections, we have explored the routing-level characteristics of blackspace networks. We have identified a small number of network practices leading to benign blackspace announcements. In order to be able to assess the security risk that is posed by the remaining set of blackspace prefixes, we need to know more about their network activities, e.g. which application-level services are running and whether they are known to be the source of some malicious network traffic. For this, we first need to find out live IP addresses and domain names, and we will then look at the services that these machines are running and check them against logs of malicious network activities.

To find that out, we lightly probed each of the blackspace networks once per day in February and March 2015, except for 10 days between Feb 16 and Feb 26 when our modem broke down. Using [ZMap], we sent a TCP SYN packet to each IP address included in a blackspace prefix on ports 21 (FTP), 22 (SSH), 25 (SMTP), 80 (HTTP), 137 (NetBios), 179 (BGP), and 443 (HTTPS). We run the scan from a machine located in AS3215 (Orange), and wait for SYN/ACK replies.

Figure 5.8 plots the number of SYN/ACK received per day and per port from the blackspace. There is quite a large number of web servers running in the blackspace. We customarily get replies from between 6k and 8k machines on port 80, and 2.5k machines on port 443. Next is port 22, with around 1k daily SYN/ACKs. There are around 100 FTP servers, and around 50 hits on port 179, suggesting suggesting that these IP addresses are border routers. Finally, we only get a handful of TCP replies on the NetBios port, and no reply at all on port 25.

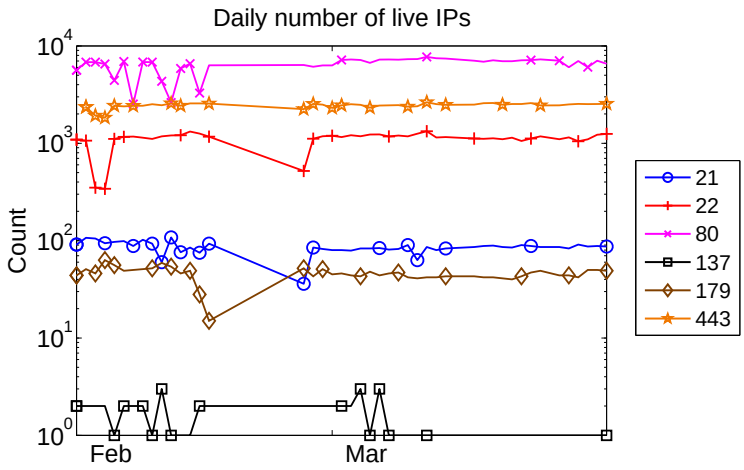


Figure 5.8: Daily number of SYN/ACK packets received from the blackspace

Figure 5.9 plots the variation of the live IP addresses in the blackspace, which indicates the persistence of these IP addresses. As we can see, the variation is quite high. These results need to be put into perspective of Figure 5.4 which showed that there was a very small variation in the blackspace networks. This suggests that the hosts inside blackspace networks are not static, but dynamically come and go. In other words, **these networks appear to be actively managed, and not left in a 'legacy' state**.

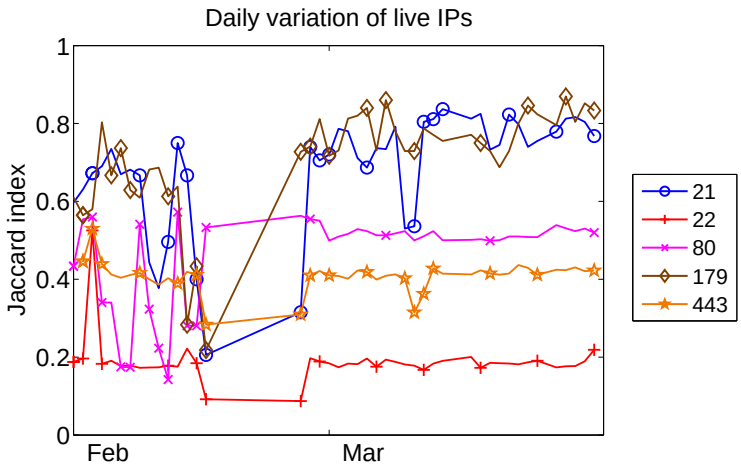


Figure 5.9: Day-to-day variation of live IP addresses in the blackspace

5.3.3.2 URLs, Websites, and Domain Names

In the previous Section, we located a set of highly volatile live IP addresses in the blackspace, and we saw that we found thousands of web servers daily. In this Section, we look at the contents of these websites and their associated URLs and domain names which we match with a domain whitelist and blacklist. A simple way to know what's going on with these servers is to check the web page they serve. As a result, we supplement our scan with a simple HTTP client that just fetches the default page returned by the server, using the simple request `GET / HTTP/1.0`. Additionally, we fetch the error 404 (document not found) page by requesting a random document, and the error 400 (bad request) page by sending gibberish. The reason behind these extra requests is to force the server to return an error page, which, sometimes, contains information about the server, such as its hostname.

Using the returned HTTP headers, we find that over 90% of pages inside the blackspace are served by an Apache server; then come IIS, and Cisco IOS. Other pages are returned by nginx and lighthttpd, various application platforms, even including a print server. Because we get thousands of pages per day, we cannot manually go through all of them. In order to help our analysis, we used an unsupervised machine learning tool that clustered our pages based on the similarity of their raw content. We get between 60 and 80 clusters. The most important one contains over 4000 Apache error pages. This implies that, for the most part, the default page of web servers located in the blackspace is an Apache error page. Other clusters include default web pages of each HTTP daemon (e.g. "your installation was successful"), log-in interfaces for router or other applications. Websites hosted in the blackspace are usually in small clusters containing 2 or 3 IP addresses. By manually checking the smaller clusters, we found various websites, ranging from discussion boards to SME websites. In some rare cases, the page content contained a lot of obfuscated JavaScript code. We used [Wepawet] [Cova *et al.* 2010] to check it out, and it always remained benign.

Since most of the web pages in the blackspace are served by Apache, we can take advantage of the default Apache error pages, which include the server hostname. By doing this, we uncovered 102 hostnames in 61 domains. At the same time, we check if the page redirects the browser to another location – we look for the appropriate header, or for the appropriate `<meta>` HTML tag. Overall, 101 IP addresses redirect the browser to 95 URLs on 90 hosts in 44 domains.

We matched 24 of these domains with a list obtained from a security software vendor. All of these were whitelisted, and belonged to well-known web applications, airlines, and technology companies. This suggests that the domains we found were most probably benign. However, we only were able to map around 150 domain names from over 10k distinct IP addresses, which could also simply imply that only the benign domains do not try to hide this kind of information. It is worth noting that we decided to extract the domain names out of the web pages after we failed to get results out of reverse DNS requests.

5.3.3.3 Malicious IP Addresses

In order to locate host-level malicious activities inside blackspace prefixes, we were able to secure a list of malicious IP addresses compiled by a security software company. These IP addresses were classified as either adware, phishing, scam, and other kinds of miscellaneous activity.

We looked for IP addresses that were included in blackspace prefixes exclusively on the days during which we detected the prefix in the blackspace. In other words, we explicitly discarded any matching IP address and its covering blackspace prefix where a match occurred outside of the blackspace period, even if there were matches during the blackspace period. The reasoning behind this (overly) strict matching is that we are looking for malicious activity that is the result of an individual abusing the blackspace in order to remain hidden. Thus, any matching malicious activity outside of the blackspace period could be argued to be the result of a previous owner of the prefix, and not from the blackspace itself. With these strict matches, we matched 46 malicious IP addresses in 28 distinct blackspace prefixes. Four of these IPs addresses were involved in scam activities, and the remaining 42 others in phishing activities.

We then looked into these 8 BGP prefixes to see if we could obtain more information from the announcements. One of the BGP prefixes was RESERVED and originated by an AS that was marked as AVAILABLE, through what appears to be a tier-3 ISP in Thailand. Six of the other BGP prefixes were also all RESERVED, and originated by registered ASes. Two of these were country-wide ISPs, one was a television by satellite broadcaster, and one belonged to a hosting provider. A European prefix was being announced by the AS of a Japanese corporation, on which we were unable to find any information.

The remaining BGP prefix is 192.0.0.0/2, which we had previously classified as a route leak. This prefix was announced between October 15, 2014 and October 20, 2014. This announcement resulted in an additional 2970 prefixes in the blackspace (see Figure 5.1). Among these, 22 contain IP addresses marked as malicious, exactly during the announcement period. More precisely, a single /24, as well as a /19, both contain 11 individual malicious IP addresses, a /22 contains 5, a /20 contains 2. The remaining 4 IP addresses are spread across different blackspace prefixes. It is important to stress that the matches were done exclusively on the blackspace period. Actually, none of these prefixes were routable before or after this leak. The route also had a low visibility: it was only seen by 3 (out of 13) RIPE RIS collector routers; and there is only one single AS path leading to the origin. The origin AS was 65,000, a private AS number (Figure 5.6), and the route was propagated through one cloud services and hosting provider, and then through a tier-3 ISP in the USA. We further discuss this situation in Section 5.4.

5.3.3.4 Spam Campaigns

In an effort to further characterize the footprints of blackspace prefixes while they were announced and determine whether they pose a security threat to the Internet, we extracted spam source IP addresses in these prefixes that were blacklisted in [Spamhaus]'s SBL (Spamhaus Block List), and DROP (Don't Route Or Peer); [PSBL] (Passive Spam Block List); [WPBL] (Weighted Private Block List); and [Uceprotect]. Furthermore, we retained only those IP prefixes where spam activities were exclusively reported while the prefixes were announced as blackspace to ensure that the observed activities were not related to the previous or next status of the prefixes. We identified a total of 206,404 distinct spam sources in 58 IP prefixes. Figure 5.10 shows the BGP announcements and blacklisted spam sources related to a sample of 15 out of 58 blackspace prefixes while they were announced as blackspace.

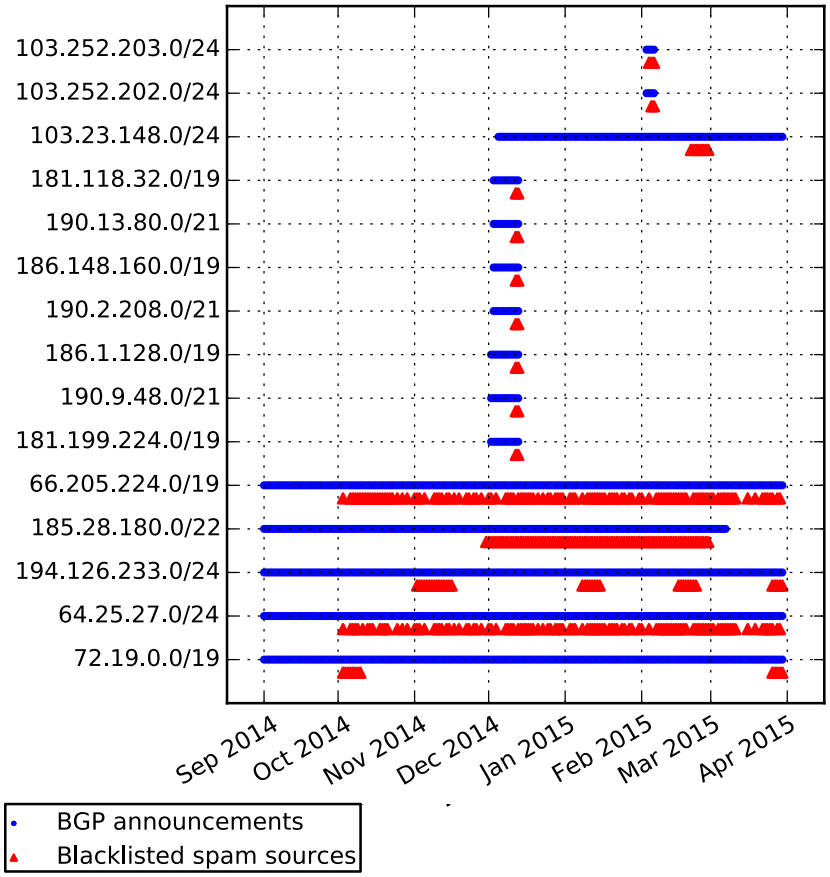


Figure 5.10: BGP announcements and blacklisted spam sources related to IP prefixes while they were announced as blackspace. For the sake of conciseness, only 15 out of 58 prefixes that were blacklisted are depicted.

Finally, we also correlated the list of blackspace IP prefixes with the output of SpamTracer [Vervier *et al.* 2015], a system specifically designed to identify network IP address ranges that are hijacked by spammers to enable them to send spam while remaining hidden. Relying on a combination of BGP and traceroute data collected for networks seen originating spam and a set of specifically tailored heuristics, the system identifies those spam networks that exhibit a routing behavior likely indicating they were hijacked. We found that 82 IP prefixes were reported by SpamTracer as hijacked spam networks at the same time we identified them as being part of the blackspace.

5.3.3.5 Case Study

Starting from the 82 particularly suspicious blackspace prefixes we uncovered a very interesting phenomenon that we describe in-depth here below. Looking closely at how these 82 network prefixes were announced in BGP revealed that they were all advertised via one AS: AS59790 "H3S Helge Sczepanek trading as H3S medien services". Based on this intriguing observation, we decided to extract from all identified blackspace IP prefixes every of those that were advertised via AS59790. Surprisingly we discovered that no less than *476* IP prefixes in total (82 of them seen originating spam by SpamTracer) were advertised via AS59790 between October 17, 2014 and January 8, 2015 and that *all of them were part of the blackspace* at the time of the BGP announcements. Furthermore, all blackspace prefixes actually correspond to IP address space allocated by the IANA to AfriNIC (the African RIR) but not yet ALLOCATED or ASSIGNED by AfriNIC to any organization. Looking at the AS paths in the BGP announcements of the 476 networks

$$\{AS_{collector}, \ldots, AS174, \textbf{AS59790}\} \tag{5.1}$$

$$\{AS_{collector}, \ldots, AS174, \textbf{AS59790}, AS201509\} \tag{5.2}$$

reveals that AS59790 was always connected to a single upstream provider AS174 "Cogent Communications (US)", a cross-continent tier-1 ISP. From the AS paths we can also see that when AS59790 did not appear as the BGP origin AS (case 5.1) it was apparently used to provide transit to AS201509 (case 5.2). AS59790 "H3S Helge Sczepanek trading as H3S medien services (DE)" was ASSIGNED on September 30, 2014 and AS201509 "Sky Capital Investments Ltd. (DE)" was ASSIGNED on October 17, 2014, shortly before they started to be used to announce the blackspace prefixes. Both ASes were registered in the RIPE region to what appear to be organizations active in the finance industry in Germany. However, we were unable to find any information regarding these organizations through extensive web searches. The description of AS59790 and AS201509 in the IRR reveals that they are in fact under the control of the same person.

In summary,

- AS59790 and AS201509 were used to announce a total of 476 blackspace prefixes over a period of approximately three weeks;
- these ASes were never used to announced any non-blackspace prefix;
- some of the blackspace prefixes announced were used to send spam, according to [Vervier *et al.* 2015].

The evidence presented here above suggests that these ASes were involved in malicious BGP announcements of IP blackspace. Moreover, [Madory 2015] recently reported on similar evidence about AS59790 being involved in fraudulent routing announcements of unallocated African IP address space. This case study thus tends to confirm the assumption that blackspace IP prefixes are purposefully used to source different types of malicious network traffic, such as spam, likely in an effort to hinder traceability.

5.4 Discussion

In this Section, we address the shortcomings and weaknesses of our methodology.

The results presented in Section 5.3 offer a granularity of 1 day. This can be explained by the following reasons. First, the data sources that we use to compute the blackspace – i.e. the IRR databases and the RIR delegated files – are only updated once per day. Second, because we are actively probing the blackspace networks, we are effectively limited by the capacity of our Internet connection. In order to comfortably run this scan in its entirety (i.e. the equivalent /10 blackspace on 7 ports, with the additional web crawling), we need, on average, 17 hours. As a result, we cannot do more than a single scan per day. Third, and consequently, we use routing table dumps from RIPE RIS instead of BGP messages. Routing table dumps are generated every 8 hours and contain the entirety of the routes known by the router. The dumps of BGP messages are generated every 5 minutes and contain all the BGP messages exchanged between the collector routers and one of its peers. With those, we would obtain a much better granularity of data, maybe even include more prefixes in the blackspace. However, since we were mainly focusing on the accurate detection of blackspace prefixes, and on the discovery of the network footprints that they have, as well as the malicious activities they carry out, we think our results are still representative. Short-lived hijacks occurring in the blackspace would not enable an attacker to host a scam website, for example.

Our probing is done from a single machine located in AS3215 (Orange). While this gives us plenty of control over the environment in which our experiment is deployed, it comes at the price of a few drawbacks. First, we don't know anything regarding the BGP-view of the network we are connected in. In other words, we are using BGP data from RIPE RIS as the source of our control-plane data, and the Orange network in order to explore the connectivity. Even though Orange is a tier-1 network, we could not find any direct peering between 'our' AS and a RIPE collector. Actually, AS3215 is routed through AS5511 – better known as OpenTransit – which contains Orange's tier-1 infrastructure. This potentially leads to false negative in our measurements, especially in the case low-visibility prefixes, such as the route leak of 192.0.0.0/2 in which we detected malicious IP addresses (Section 5.3.3.3). Would probes sent from our vantage point have reached the originating network, or would they have been dropped because there would be no "route to host"? The optimal way carry out these measurements is from a machine that runs BGP so as to assess the reachability of the destination.

At the beginning of Section 5.3, we saw two BGP events leading to a sudden and massive increase of the blackspace size. We classified these events as route leaks because they were only seen by a handful of RIPE collectors – 3 collectors for the leak in October; 1 collector for the one in January – and because there was only a single AS path between the collector(s) and the origin. However, because we also detected malicious activities inside of them, the question of whether these events were deliberate attacks disguised as route leaks needs to be raised. Unfortunately, we cannot provide a definite answer. But [Toonk 2015] recently underlined highly localised BGP hijacks, engineered to have a very low footprint, and to remain invisible from the point of views of route collectors.

5.5 Related Work

The oldest report of malicious activities carried out from the bogon address space dates back to 2001 with [Thomas 2001], where the author provided an analysis of the attacks carried out against an active web site. A large proportion of attacks originated from bogon addresses: 13% from within the bogons of classes A, B, and C; 53% from classes D (multicast) and E (future use). All in all, by properly filtering incoming traffic at a border router, 66% of attacks could easily be mitigated.

As a result, Team Cymru set up the [Bogon Reference] project, which precisely defines the different categories of bogon prefixes. We used this as the basis of our definitions in Section 5.2.

Additionally, multiple lists of bogon prefixes are offered to network owners who wish to filter bogons out of their networks, which can be retrieved in many convenient ways and formats. These lists vary according to the desired level of precision. The bogon lists contain the prefixes still reserved in the IANA pool, as well as prefixes reserved by RFCs for specific use cases. The *full* bogon list supplements these prefixes with prefixes that have been allocated to RIRs by the IANA, but not by RIRs to ISPs or end users. These lists are dynamic, and network operators that use them should update their filters accordingly. Unfortunately, the methodology used to populate these lists is not disclosed. By comparing the full bogon list with our blackspace list, we were able to identify key differences. First, the full bogon list does not make use of the IRRs, as evidenced by many prefixes for which an `inetnum` object could be found. Second, the full bogon list appears to implement some heuristics based on the status of the prefixes. For example, we noticed that prefixes whose status transitioned from either ALLOCATED or ASSIGNED to RESERVED were not listed in the full bogon list. We also noticed that some prefixes that were RESERVED for a long time were not listed, although it might be that the transition happened before our data was gathered. We ignore the motivations behind these heuristics. However, the comparison of our blackspace list with the full bogon list on the same day shows that using the IRR databases in addition to the RIR delegation files improves the accuracy of the list.

[Feamster *et al.* 2004] provided the first formal study of bogon prefixes by looking into the prevalence and persistence of bogon announcements, as well as the origin ASes leaking these prefixes. However, the authors did not explicitly focus on the blackspace, but rather on the equivalent of the (simple) bogon list. Consequently, 70% of the analyzed events actually involve the prefixes reserved for the private IP space. 40% of the events lasted longer than a day. In our analysis, this value is of 75% (Figure 5.3). The rest of the study cannot be directly mapped onto our results, even though the beginning of Section 5.3 provides results to similar questions. However, with the authors' methodology, there is a one-to-one mapping between the BGP routing table and the bogon analysis. With this, they can focus on the number of bogon prefixes announced by an AS. In our case, we have a one-to-n relationship between the BGP prefix and the blackspace prefixes because we divide the BGP announcement in separate parts that may have been assigned independently. The authors also focus on the effect of bogon filtering and show that network operators who filter out bogon prefixes usually do not update their filters in timely fashion, resulting in reachability issues and potential denial of service. It is also worth noting that the bogon prefixes used for the study were composed of the 78 /8 prefixes that still belonged to the IANA pool back then (excluding class E). Today, the IANA pool only consists of one single /8 prefix, 0.0.0.0/8 (also excluding 240.0.0.0/4). As a result, the IP address space inside which our studies have been conducted is much different.

5.6 Summary and Conclusion

In this Chapter, we focused on the IP blackspace, which is composed of the set of prefixes that are globally announced through BGP but have not been assigned for use to any entity. We presented a thorough methodology to compute the blackspace by using a combination of data sources reflecting the current allocations of the IP space. We saw that the daily blackspace address space is equivalent to a /10 prefix, and that the prefixes that compose it change little over time. We actively studied those networks from the BGP control plane point of view, and also from the data plane point of view. While we showed that some of the blackspace is composed of prefixes that are either being phased out of the Internet or being installed, a significant part of it does not result from normal network operations, such as assignments and decommissions. By cross-checking with various reliable security data sources, we were able to isolate malicious activities that only occurred during a period

in which the monitored prefixes were inside the blackspace. Even by using our strict matching rules, and our limited, targeted view of these networks, the amount of this malicious activities is significant. In particular, we showed through a validated case study that cybercriminals **do** abuse blackspace prefixes to carry out nefarious activities while also hindering their tracability.

Consequently, this Chapter confirms how important it is to precisely filter blackspace prefixes out of BGP. However, filtering out the whole blackspace implies a loss of connectivity to the networks that are being added to the Internet. Especially if we consider the fact that, when a bogon filtering system is in use, it is not updated often enough and thus obsolete [Feamster *et al.* 2004; Bush *et al.* 2007]. Moreover, the current state-of-the-art source of bogon filtering [Bogon Reference] does not take into account `inetnum` entries from IRR databases, thus including – and preventing access to – networks that have been assigned to a customer.

This Chapter also underlines the difficulty of using a ground truth in BGP. Even though the prefixes that we focused on all have in common the fact that they should not even be used on the public Internet, we were able to show cases where their use was the result of legitimate practices. As a result it is still quite difficult to automate the estimation of the danger resulting from a particular prefix in the blackspace.

Publications

The material presented in this chapter led to the following publication:

- Quentin Jacquemart; Pierre-Antoine Vervier; Guillaume Urvoy-Keller; and Ernst Biersack. "Demystifying the IP Blackspace".
  In 18th International Symposium on Research in Attacks, Intrusions and Defense (RAID 2015), November 2015.

Conclusions and Future Perspectives

6

In this Dissertation, we set out to study prefix hijacking from the third-person point of view, i.e. an external observer looking at BGP routing events who tries to locate hijacks. For this purpose, existing state-of-the art tools are inadequate because of the high rate of false positives they generate. The reason is that they are designed with network operators as target audience. These network operators know which behaviour they expect for their IP prefixes and AS numbers. As a result, they only need to focus on a very specific number of alerts, can effortlessly dismiss the false positives, and act upon the real alerts. For us, however, as external observers, two issues are key. First, we cannot filter the alerts based on the network that raised the alert. We don't have a network, we want to know if and when something suspicious is happening. Second, we don't know the correct configurations behind any of the involved networks. In other words, we don't know if an alert is the result of a configurational change in the observed network, or the result of a prefix hijacking attempt.

All in all, in order to be able to look at prefix hijacking as an external observer, we need to solve these two problems. First, we need a detection technique that provides us with an *alert list of manageable* size. If there are too many alerts, it is not possible to investigate for the root cause of every event. In order to reduce as much as possible the number of false positives events to investigate, we provided detailed analyses of standard network practices. With these, we uncovered a set of benign configurations that we could easily remove from the set of suspicious events. As a result, a non-negligible part of this Dissertation focused on the analysis of the Internet as seen from BGP, which leads to contributions with regards to the understanding and the architecture of the Internet in general.

Second, we need to be able to look at these alerts and either mark them as a benign change in the routing behaviour, or as an attempt of prefix hijacking. Because we do not own the networks involved in the suspicious event, we need to **gain access to ground-truth**. Relying on feedback from the rightful owner is impractical: network operators are usually unwilling to disclose too much information about their own network. Moreover, this operation cannot be automated, meaning that the feedback will only be available if someone at the owner's network operation center is willing to take the time to reply. To circumvent this, we rely on auxiliary datasets that extend the

reach of our BGP data: IRRs (Internet Routing Registries), which provide registration information about ASes (Autonomous Systems) and IP prefixes; security information, such as spam blacklists; and application-level information, such as NetFlows collected on Munich's Scientific Network, port scanning, and web crawling. With this augmented data, we are able to look at a BGP event and infer the context in which it happened. This context includes who the involved parties are, and how much their behaviour was altered during the suspicious event. Using all this information, we are able to closely approximate ground-truth in an automated and repeatable way.

In this Dissertation, we specifically looked at three vectors of attack for prefix hijacking. In Chapter 3, we started by looking at concurrent ownership attacks which result in a MOAS (Multiple-Origin AS) prefix, i.e. an IP prefix being originated simultaneously by multiple ASes. First, we updated previous works by considering MOAS events as a set of events that are related to IP prefixes. By doing this, we successfully showed that short-lived MOASes are not, unlike previously reported, the result of misconfigurations, but the result of origin instability or route flapping. Second, we introduced a **taxonomy of MOAS networks by classifying the MOAS occurrences into three distinct patterns**. In 70% of all MOAS cases, the origins were directly peering with one another in a stable manner, resulting in a *peering MOAS*. In a little less than 30% of MOAS cases, the two origins were disjoint. We call this situation a *classical MOAS*, as this is the configuration that is usually thought of when talking about MOASes, and the one that had been the focus of previous works. In the remaining cases, we have a *me-too MOAS* pattern, which combines both of the other patterns altogether. This last situation is most often encountered during a network topology change, such as when a stub network decides to change its upstream provider. In order to know if our measurement and classification were stable, we provided the analysis of two full year of data collected ten years apart. Even though the Internet substantially grew during these 10 years, our measurements showed a staggering similarity between these two years, thereby underlining the sustainability of our approach. Armed with this taxonomy, we presented a set of heuristics in order to **filter out more than 80% of false positive alerts** compared to existing state-of-the-art approaches. We then illustrated our approach with a case study, the **Bulgarian case**, a suspicious MOAS case where a spammer was believed to have hijacked IP space. Even though we remain indecisive on the final outcome, this case study showed that a casual correlation of security datasets with BGP data is not enough to conclude on the maliciousness associated with a routing event. We showed how we were able to use the IRR databases in order to obtain ground-truth that would otherwise have required direct feedback from the network owner.

This work can be extended as follows. On the network analysis front, future work includes expanding our ground-truth sources with verified peering information to supplement WHOIS data. This would permit further validation and classification of peering MOASes. Another direction is to deeper study the flipping between SOAS and MOAS related to a single prefix, which could be the result of an intervention of the owner, in order to comply to the terms of a peering agreement (e.g. exceeded bandwidth). Finally, our analysis currently makes use of a single vantage point to analyse MOASes, which certainly results in under-estimating the number of MOAS events seen, particularly in terms of peering MOASes. (Classical MOASes try, by design, to diversify the AS paths as much as possible.) Although we are confident that the global trends and orders of magnitudes we exposed in this study remain true regardless of the vantage point, using (multiple) different route collectors would certainly provide better estimates. On the hijack analysis front, a deeper integration of the various parts and datasets involved in the proposed system would enable us to perform automated large scale analysis of suspicious hijack cases.

In Chapter 4, we looked at the way **prefixes overlap in the BGP routing table**. This aspect is important because BGP naturally favours most specific routes when forwarding traffic. Consequently,

any undesired more specific prefix announced by an unauthorized AS implies that the traffic to the legitimate network will be blackholed. The network analysis part of Chapter 4 made an extensive use of the **IRR databases in order to cluster the prefixes announced through BGP into families of prefixes related to the same entity**. In other words, prefixes inside a family have all been allocated to the same end-user, which implies that we can study the way end-users divide their own IP space into smaller parts. We showed that almost three quarters of all allocated prefixes are announced as-is in the BGP routing table, without additional internal overlap. This implies that most of the entries in the global routing table are due to the sheer number of allocations that have been made. Only one quarter of all families announce one or more specific prefixes than the one they were allocated, and, for the most part, this prefix is originated in the same AS as the less specific one. This can be explained by traffic engineering practices. When a child is not located in the same AS as its parent, the child can be located downstream of the parent, which implies a sub-allocation of IP space; but the child can also be located upstream of the father, which is used, for example, as a DDoS mitigation technique. We also saw that the IRR databases contain a lot of records, and that companies of all sizes, particularly ISPs (Internet Service Providers), appear to actively populate it. This strengthens our confidence in relying on IRR databases in order to obtain ground-truth for analyzing BGP hijacking cases. Then, we presented the prototype of a **system in order to validate sub-MOAS announcements**, i.e. when a child prefix is located in another AS than its father's. This prototype uses the IRR databases in order to infer a business relationship between the involved ASes. It also makes use of a dataset of SSL/TLS public keys in order to check if the machines hosted on the network behind the sub-prefix are the same as the machines that were on the original network before the sub-prefix was announced. Even though our prototype currently only makes use of the IRR of RIPE NCC, **we were able to mark as benign over 50% of all sub-MOAS cases**.

This work can be extended as follows. On the network analysis front, we would like to further study unannounced families, i.e. the many entries in the IRR that do not appear to be announced in BGP. By doing so, we hope to sort out for good whether the information inside the IRR is stale, or whether the families appear to be offline to our vantage point, but might exist somewhere further away from the core (default-free) zone of the Internet. A first step would be perform IP-level measurements, such as traceroutes, in order to see how the IP-level topology for addresses within the unannounced family differs from the topology inside the announced family. Second, we want to investigate the large number of subfamilies maintained by ISPs. We saw that ISPs maintain a large number of families in the IRR database, i.e. they are willing to spend time creating these objects. We argue that they spend time doing this because they find the IRR database valuable, which implies that information contained in the database is not bogus. Finally, we want to study the relationship between a family and its subfamilies, since our study so far was only done intra-family. An interesting topic would be the AS-level topology between these families, e.g. how far apart these families are. We expect subfamilies of tier-1 ISPs to be – at least partially – located behind the tier-1. On the prefix hijacking detection front, our prototype for sub-MOAS validation shows encouraging results. We are in the process of extending its reach by making it compatible with the other IRRs. Moreover, the database of public keys that we use as ground-truth can be easily extended by including keys gathered from other popular SSL/TLS based services, such as email protocols (POP3S, SMTPS, IMAPS), FTPS, LDAPS, ...

In Chapter 5, we focused on the **blackspace**, i.e. the IP space that has not been allocated for use, but that is routable, and thus usable. We presented a **methodology** that makes use of the IRRs and of the RIR statistic files in order to compute the unallocated space, which we cross-check with the BGP routing table in order **to locate the blackspace**. By doing this, we effectively start our study by knowing the ground-truth in advance: any IP prefix in this space should *not* be publicly routed.

Nevertheless, we showed that there is a significant number of networks included in the blackspace, and were able to show that a part of it results from regular operations on the Internet, such as networks being decommissioned or assigned. This underlines, again, how tricky BGP ground-truth is: even though these networks should *not* be in use, and should *not* be announced, an in-depth look at their content and evolution showed a number of somehow legitimate use. On the other hand, by crawling the websites available in the blackspace, as well as correlating the blackspace with well-known security-related blacklists, we were able to successfully locate a number of spammers that abuse the blackspace to send spam emails. By looking at ASes that announce both blackspace and *dormant space*, i.e. IP space that has been allocated but was previously unannounced, we were able to locate and study an occurrence where a single entity abused the blackspace to send spam and host a scam infrastructure. Specifically, almost 500 prefixes were abused in approximately 3 weeks, from ASes that were used exclusively for this purpose. This implies that **some parties appear to be able to locate the desirable defenseless resources of the Internet in order to abuse them**. The blackspace is particularly fragile in this aspect because, since it does not belong to any entity, it is less likely to be monitored by anybody.

This work can be extended as follows. First, we plan to define a set of reliable heuristics that would discard benign blackspace announcements and only retain those that are potentially malicious, thus increasing the quality of existing filters that can be installed on routers. Second, we would like to supplement our probing system with a traceroute infrastructure that would enable us to geographically locate the origin of these networks, and the diversity of their connectivity. This would let us see if there are specific parts of the network that hijackers prefer to abuse. Third, we would like to improve our measurement platform, by performing our scans and HTTP crawling from a vantage point that has a direct BGP visibility. For this, we would need a set of geographically diversified machines that run BGP – each connected to a different set of peers – and from which we can run our measurement experiments. If this can be achieved, a bonus point would be to make the system run real time, by detecting and probing networks as they come and go in the BGP routing table.

The work presented in this Dissertation was carried out exclusively on the IPv4 space. It would be interesting to know how the results presented, particularly in terms of network analysis, transpose to the larger IPv6 space. To the best of our knowledge, there is currently no work that discusses prefix hijacking that has focused on IPv6, meaning that the existing experiments could, and should, be done over IPv6. We believe that our system is pretty much IPv6 proof: the vast majority of the underlying code works with abstract IP addresses, therefore using IPv6 instead of IPv4 simply means initializing the data structures with a larger bit vector. However, several external aspects would be lacking. First, there is a shortage of security-related blacklists that take into account IPv6. Second, the infrastructure we currently use to carry out our measurements, such as the scans and the web-crawling in Chapter 5, does not support IPv6, because our ISPs either does not support it, or our IT support has not started deploying it yet. We will investigate ways to move this infrastructure outside of our Institution. An option would be to rent a VPS service (Virtual Private Server) from an IPv6-enabled company. However, we should cautiously check out the acceptable terms of services in order to know if we can carry out our full experiments from their address space.

Due to lack of available time, the work presented in this Dissertation does not focus on the last attack vector for prefix hijacking attacks: the AS path. Some of the heuristics proposed in Chapter 3 and Chapter 4 depend on the AS-level relationship between the involved ASes. This implicitly means that we do trust the AS path to provide reliable information. As a result, being able to verify the contents of the AS path would not only consolidate our heuristics, but it would also enable us to include the ones that we had dismissed because they could be the result of a man-in-the-middle

attack. A passive verification of the AS path should be possible by using a variety of datasets, including the routing policies from the IRRs and peering relationship databases.

Finally, each of the technical chapters of this Dissertation focused on tackling a different challenge, i.e. a different vector of attack for prefix hijacking. Consequently, these sub-systems could be integrated into a larger monitoring system, which would be able to accurately report prefix-hijacking attempts in near real-time. The advantage of such a system would be considerable. First, to network operators, it would be an incremental upgrade over the services that they can now get. They can use it to monitor their resources. Because the system would provide fewer false positive events, they would receive less alerts. For the networking and security community, this system would provide a unique source of highly suspect BGP routing events from which a thorough investigation can begin. Such a system is necessary in order to finally be able to study the prevalence of prefix hijacking attacks, whether or not they are malicious, and their characteristics. With a clearer view on the phenomenon, we hope the Internet community will evolve towards taking the necessary steps to properly secure BGP.

Déceler les attaques par détournement BGP : synthèse

A

Internet est composé de dizaines de milliers de *systèmes autonomes* (**Autonomous Systems**, AS) indépendants, appartenant, tous, à des organisations diverses, telles que compagnies privées, fournisseurs d'accès Internet (FAIs), universités, centres de recherche, etc, qui les administrent. Ces AS échangent des informations d'accessibilité (c.-à-d. des préfixes IP) grâce à BGP (**Border Gateway Protocol**). Lors de la conception de BGP, les objectifs étaient principalement la robustesse et la tenue en charge, la sécurité n'étant, à cette époque, pas considérée comme nécessaire. De plus, l'ajout à BGP d'éléments de sécurisation aurait nécessité une surcharge de travail non négligeable pour les routeurs BGP. Ainsi, il existe une forme de **confiance mutuelle** *implicite* entre les AS : chaque AS est incapable de vérifier la validité des routes qu'il reçoit des autres AS.

Le **détournement de préfixe** (*prefix hijacking*) tire parti de cette confiance mutuelle afin d'introduire des routes falsifiées sur Internet. Ce phénomène peut être *accidentel*, c.-à-d. le résultat d'une erreur de configuration d'un routeur, mais il peut aussi être le résultat d'une attaque *délibérée* contre l'infrastructure de routage dans sa globalité, ou contre un réseau en particulier. Le détournement de préfixe peut être utilisé pour créer un trou noir dans le réseau (qui résulte en un déni de service pour la victime), prendre l'identité de la victime, espionner (avec possibilité d'attaques de type de l'homme au milieu), ainsi que d'autres activités malveillantes (telles que le pourriel).

Plusieurs méthodes ont été proposées afin de *sécuriser BGP*, notamment en introduisant des mécanismes permettant de vérifier les informations transmises aux AS grâces à des opérations cryptographiques. Cependant, ces opérations se traduisent en une très grande charge de travail supplémentaire pour les routeurs BGP ; et, donc, ces méthodes ne seront probablement pas adoptées dans le futur à proche ou moyen terme. En conséquence, un certain nombre de **techniques de détection** d'attaques par détournement, qui, elles, sont déployables à l'heure actuelle, ont été conçues. Ces techniques génèrent un nombre extrêmement grand d'alertes, principalement dues à des **faux positifs** résultant d'opérations de routage courantes et bénignes. Dans ces conditions, ces techniques de détection ne permettent pas d'étudier les attaques par détournement de préfixe parce qu'un **observateur externe ne connaît** en général **pas** la **réalité de terrain** (*groundtruth*)

nécessaire afin d'identifier les faux positifs de manière aisée, rendant ainsi nécessaire une inspection manuelle de chaque alerte.

L'objectif principal de cette dissertation est d'identifier la cause principale des évènements de routage (c.-à-d. des alertes de détournement de préfixes) de manière indubitable. A cette fin, d'une part, nous **réduisons le nombre global d'alertes** en analysant un grand nombre de faux positifs. Grâce à cette analyse, nous mettons en avant des structures, composées de schémas et de tendances représentant des pratiques de routage standard, mais variées. Ensuite, nous considérons l'impact de ces structures dans le cadre d'une attaque par détournement. D'autre part, afin de **contourner le manque de connaissance de la réalité de terrain**, nous analysons plusieurs aspects distincts des évènements suspects, grâce à un certain nombre de sources de données auxiliaires qui nous donnent des informations à propos des réseaux impliqués, par exemple les registres d'enregistrement, des signes d'activités malveillantes connus, ainsi que des informations relatives aux applications actives sur ces réseaux.

Précisément, nous considérons trois cas distincts de détournement par préfixe. Premièrement, nous considérons les **préfixes à origines multiples** (*Multiple Origin AS*, MOAS), c.-à-d. les préfixes qui sont annoncés simultanément par plusieurs AS. Nous présentons une taxonomie des MOAS, et proposons une série de filtres qui permettent d'**écarter** automatiquement **plus de 80% de faux positifs**. Nous présentons aussi l'analyse d'un cas issu du monde réel où un MOAS a coïncidé parfaitement avec du pourriel et du trafic web d'arnaque en ligne. Nous montrons que l'approche courante – qui consiste à corréler les évènements de routage et les activités malveillantes – est *insuffisante* pour prouver une attaque intentionnelle et malveillante de détournement de préfixe, et illustrons que notre méthodologie permet d'apporter une meilleure compréhension des évènements de routage. Ensuite, nous considérons les *recouvrements de préfixes*, et, en particulier, le cas des préfixes plus spécifiques (more specific prefixes). Nous analysons la table de routage BGP, clarifions les pratiques courantes d'ingénierie derrière ces préfixes, et présentons un prototype qui permet, pour l'instant, d'**éliminer approximativement 50% de faux positifs**. Enfin, nous explorons l'**espace noir IP** (*IP blackspace*), composé de l'espace IP qui est activement annoncé sur Internet, bien qu'il n'ait jamais été assigné. Nous étudions les caractéristiques BGP de ces réseaux et identifions un certain nombre de raisons bénignes pour lesquelles ils sont annoncés. Ensuite, nous nous concentrons sur les risques associés à ces réseaux en regardant leurs traces au niveau applicatif, en identifiant les machines actives, et mettons au jour une grande quantité de pourriels et de sites d'arnaque en ligne localisés dans l'espace noir.

A.1 Introduction

Dans cette section, nous introduisons les problèmes que nous allons discuter dans cette dissertation. Premièrement, nous présentons l'architecture d'Internet et la façon dont BGP (Border Gateway Protocol) y est utilisé. Ensuite nous présentons les attaques par détournement de préfixe – un type d'attaque contre BGP – et le problème auquel nous nous attaquons. Nous montrerons pourquoi cette attaque est toujours problématique aujourd'hui, et illustrerons, avec des cas réels, les répercussions qui découlent de type d'attaque. S'en suivra une courte discussion de l'état de l'art des techniques de détection et de défense contre le détournement de préfixe ainsi que les raisons pour lesquelles ces méthodes sont insuffisantes. Enfin, nous présenterons notre méthodologie ainsi que des améliorations qu'elle apporte comparé aux systèmes existants.

A.1.1 Internet et BGP

A l'heure actuelle, Internet est composé de plus de 50 000 réseaux d'ordinateurs distincts, appelés **réseaux autonomes** (*Autonomous Systems*, AS). Ces AS permettent à des centaines de millions d'ordinateurs de tous types (centres de données, ordinateurs personnels, ...) d'échanger des informations tout aussi variées. En conséquence, Internet peut être considéré comme le plus grand système à jamais conçu par l'humanité. La première itération d'Internet, alors appelée **ARPAnet** avait pour but de partager les ressources de calcul par ordinateur des laboratoires de recherche. En 1969, ARPAnet était composé de 4 ordinateurs ; en 1972, de 15. Au cours des années 1970, plusieurs réseaux similaires à ARPAnet furent développés ; en particulier, CSNET et NSFNET. Ces réseaux avaient un but similaire à ARPAnet, mais étaient financés par des organismes différents. En 1975, un travail majeur dirigé par Vinton Cerf et Robert Kahn a mené à la formalisation d'une première version de TCP (Transmission Control Protocol) ; un protocole définissant une architecture et un langage commun pour tous ces différents réseaux. Les évolutions apportées à cette version de TCP ont permis la définition des protocoles qui sont, aujourd'hui, toujours au coeur d'Internet : IP (Internet Protocol), UDP (User Datagram Protocol), et une version moderne de TCP. Le 1er janvier 1983, les protocoles ARPAnet, qui était alors composé de 200 hôtes, ont été remplacés par la suite TCP/IP. En 1989, ARPAnet et NSFNET furent connectés ensemble. A cette fin, le protocole **BGP** (Border Gateway Protocol) a été conçu pour permettre l'échange des informations d'accessibilité entre ces divers réseaux autonomes. [Potaroo ; Kurose et al. 2010 ; NSF1 ; NSF2 ; Tanenbaum 2002 ; van Beijnum 2002]

Cette courte introduction à l'histoire d'Internet illustre deux aspects fondamentaux de ce réseau. Premièrement, Internet est un réseau de réseaux, c.-à-d. un ensemble de réseaux indépendants utilisant les mêmes protocoles pour échanger des informations. Ces *réseaux autonomes* sont gérés par des entités différentes, telles que universités et laboratoires de recherche, compagnies privées, FAIs, ... Puisque ces réseaux sont indépendants, la manière dont ils sont exploités et gérés est variable, même lorsque les opérateurs de réseaux distincts rencontrent des difficultés techniques similaires. Deuxièmement, les protocoles qui sont aujourd'hui au coeur du réseau ont été créés quand les réseaux informatiques étaient taille très modeste. En fait, il ne serait pas surprenant que personne, à cette époque, n'avait imaginé qu'un réseau informatique atteindrait un jour la taille de notre Internet. De plus, pendant près de 20 ans, les utilisateurs des réseaux précurseurs d'Internet étaient des scientifiques utilisant ces réseaux afin de partager leurs ressources et connaissances. En conséquence, il y avait une **confiance mutuelle** *implicite* entre ces utilisateurs. Par ailleurs, puisque la conception et le développement de BGP ont eu lieu fin des années 1980 et début des années 1990, presqu'aucun mécanisme de sécurité n'y a été intégré, ce n'était, à l'époque, pas nécessaire, et, de plus, aurait nécessité une plus grande complexité au niveau du protocole, ainsi qu'une charge de travail supplémentaire pour les machines relativement peu performantes de l'époque. Au final, le protocole BGP est simple et robuste, mais aussi presque totalement dépourvu de mécanisme de sécurité.

A.1.2 Le protocole BGP

La section A.1.1 a montré qu'Internet est composé d'un ensemble de réseaux indépendants, appelés **réseaux autonomes** (*Autonomous Systems*, AS). Historiquement, un AS était un domaine de routage sous l'autorité d'une administration unique : un protocole de routage interne unique était sélectionné (par exemple OSPF), et une politique commune de routage était définie [RFC4271]. Aujourd'hui, cette vision des choses est trop simplifiée : différentes parties d'un même AS peuvent

se comporter de manière totalement différente parce qu'elles sont administrées par des personnes différentes [Bush et al. 2009]. Cependant, les routeurs appartiennent toujours à la même organisation, et, donc, de l'extérieur, *paraissent* avoir un comportement cohérent.

Dans cette section, nous introduisons le **fonctionnement de base** de BGP, définit, pour la première fois dans [RFC1105]. Un an plus tard, BGP-2 était proposé dans [RFC1163]. La troisième version fut finalisée en octobre 1991 dans [RFC1267] ; et la version courante, BGP-4, fût introduite en 1994 dans [RFC1771], puis révisée dans [RFC4271] en 2006.

A.1.2.1 Introduction

L'objectif principal de BGP est d'**échanger les informations d'accessibilité des réseaux** entre deux routeurs BGP. Si ces routeurs sont situés dans le même AS, on parle de routeurs *pairs internes*, ayant établi une *session BGP interne*. Ce type de session sert à la dissémination, à l'intérieur d'un AS, de l'information d'accessibilité connue grâces aux AS voisins. Les routeurs BGP internes doivent adopter une topologie de réseau maillé complet, topologie qui pose des problèmes à grande échelle. Dans le reste de cette dissertation, nous ne considérerons que les **sessions BGP externes**, c.-à-d. la situation dans laquelle les routeurs BGP pairs font partie de deux AS différents.

BGP est, *de-facto*, l'unique protocole de routage inter-AS. BGP est aussi un protocole dont le routage peut être influencé par une politique de service permettant à chaque AS d'imposer des critères de routage stricts. Cette politique est souvent le fruit de considérations économiques, telle que le prix de la bande passante, et dépend du type de relation existant entre deux AS voisins [Gao 2001]. En plus, BGP est un protocole à vecteur de chemins : il garanti le routage sans boucle en conservant la liste de tous les sauts d'AS dans un *attribut* connu sous le nom de **chemin d'AS** (*AS path*). Ces notions seront formalisées dans la section A.1.2.3.

Bien que le but premier de BGP soit d'échanger les plages d'adresses IP accessibles dans chaque AS, BGP est implémenté dans la couche applicative et dépend de TCP pour établir une session entre deux routeurs pairs. Le port officiel attribué à BGP est le port n°179 [PNR]. Cela signifie qu'un routeur BGP doit être capable d'établir une connexion TCP avec son pair sans nécessiter d'information d'accessibilité apprise via BGP, ce qui est possible en pré-configurant le routeur, par exemple avec une route statique.

A.1.2.2 Préfixes IP et numéros d'AS (ASN)

Depuis sa version 4, BGP propage l'information d'accessibilité avec le standard CIDR [RFC4632]. Dans la représentation CIDR, la plage d'adresse IP d'un réseau est représentée par un **préfixe IP**. Un préfixe IP est composé de deux parties séparées par une barre oblique. Par exemple, le préfixe IPv4

192.168.0.0/16

indique que l'*adresse réseau* est 192.0.0.0 et que les 16 bits les plus significatifs du *masque réseau* sont vrais. En d'autres termes, ce préfixe IP représente les 2^{16} adresses IP entre 192.168.0.0 et 192.168.255.255. La notation standard équivalente à ce préfixe IP est 192.168.0.0/255.255.0.0, où 255.255.0.0 est le masque réseau.

Puisque les préfixes IP représentent un ensemble d'adresses IP, des relations entre les préfixes peuvent être définies. Il est possible qu'un préfixe IP soit entièrement recouvert par un autre préfixe. Par exemple, 192.168.128.0/24 est entièrement inclus dans 192.168.0.0/16. On dit du premier qu'il est **plus spécifique** que le second ; et que le second est **moins spécifique** que le premier. Pour être plus spécifique qu'un autre préfixe, un préfixe doit avoir la même adresse réseau que l'autre et un masque plus long. De plus, si A est plus spécifique que B, B est moins spécifique que A. On dit aussi que A est un **sous-préfixe** de B, et que B est un **sur-préfixe** de A.

D'une manière similaire au fait que chaque machine connectée à Internet est identifiée par une adresse IP globalement unique, chaque AS dispose d'un **numéro d'AS** (*AS Number*, ASN) unique d'une longueur de 4 octets qui l'identifie [RFC6793]. Un AS est donc peut donc être identifié uniquement par son numéro. Par exemple

AS2200

identifie Renater par son numéro unique : 2200.

A.1.2.3 Fonctionnement

Dans cette section, nous introduisons le fonctionnement du protocole BGP. Bien que BGP soit un protocole extrêmement complexe, les principes fondateurs dont ses opérations dépendent sont simples et purs. Vu le contexte de synthèse inhérente à ce chapitre, cette introduction sera courte, et sera restreinte aux concepts nécessaires à la compréhension des faiblesses du protocole menant au phénomène de détournement de préfixe. Une version détaillée du fonctionnement de BGP dans le cadre d'une session externe est disponible à la section 2.1.

Dans le cadre du fonctionnement de BGP, une **route** est composé d'une destination, c.-à-d. un préfixe IP, ainsi que d'une série d'*attributs*. Ces routes sont échangées grâce aux messages BGP de type *update*, qui sont composés de deux parties distinctes :

1. les destinations à retirer,
2. les routes annoncées.

Le contenu de ces messages permet à chaque routeur BGP de constituer sa **table de routage**, à partir de laquelle il transmet les paquets IP.

Chaque route est propagée entre deux AS voisins, c.-à-d. deux AS inter-connectés directement via BGP. Par exemple, la figure A.1 montre comment la route 1.0.0.0/8 (en rouge) est propagée à travers les différents AS. AS1 est l'**origine** du préfixe. Autrement dit, c'est dans AS1 que les machines dont les adresses sont incluses dans 1.0.0.0/8 sont connectées. AS1 *annonce* cette route à son voisin AS3. De la même manière, AS3 l'annonce à ses voisins AS2 et AS4 ; et ainsi de suite.

En plus du préfixe IP de destination de route (soit, 1.0.0.0/8), un *attribut* appelé le **chemin d'AS** est adjoint. En résumé, le contenu du chemin d'AS est modifié à chaque fois que la route est propagée à un nouvel AS. Toujours en utilisant la figure A.1 comme référence, lorsque AS1 annonce la route à AS2, il y insère son numéro d'AS. Etant donné qu'il est l'origine, le chemin d'AS a pour seul valeur 1. En regardant la table de routage de AS3, on voit que la route 1.0.0.0/8 est adjointe du chemin d'AS contenant pour seule entrée 1. Lorsque AS3 va propager cette route à ses voisins, il

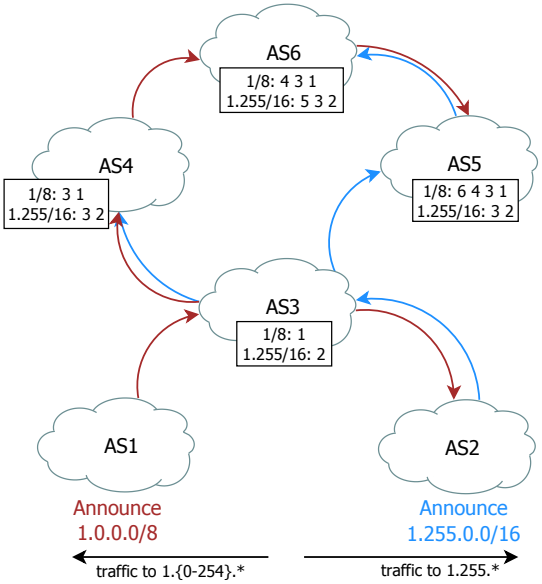


Figure A.1: Exemple de propagation de routes entre AS via BGP. Chaque AS est représenté par un nuage dont la table de routage est détaillée dans son rectangle. Les flèches indiquent le sens de propagation de la route.

va ajouter son numéro d'AS au début du chemin d'AS. Donc, lorsque AS4 reçoit la route 1.0.0.0/8, le chemin d'AS aura pour valeur 3 1. De même, lorsque AS4 propage cette route à AS6, il y ajoute son numéro d'AS, etc. Au final, le chemin d'AS indique les numéros des AS à traverser entre le point local et l'origine du préfixe. Par exemple, pour une machine située dans AS5, le chemin d'AS de la route 1.0.0.0/8 indique qu'il faut traverser successivement AS6, puis AS4, puis AS3, pour arriver à l'origine de la destination : AS1. Ainsi, par construction de proche en proche, le **dernier** AS indiqué dans le chemin d'AS est l'AS d'origine de la route. Pour cette raison, les implémentations du protocole BGP considèrent en général que la longueur du chemin d'AS est un bon indicateur de la distance entre deux réseaux. Il en résulte que, pour un même préfixe, une route avec un chemin d'AS plus court sera choisi au détriment d'une autre avec un chemin d'AS plus long.

Comme indiqué à la section A.1.2.1, le chemin d'AS permet d'éviter les boucles de routage dans le réseau. Ainsi, lorsqu'une route est reçue par un routeur BGP, celui-ci consulte le chemin d'AS. S'il y trouve son propre numéro d'AS, cela signifie qu'il est déjà utilisé par ses voisins (proches ou lointains) comme faisant partie du chemin entre eux et l'origine de la route. Dans ces conditions, le routeur ignore la route et ne doit pas considérer une insertion dans sa table de routage.

Dans la section A.1.2.1, nous avons aussi indiqué que BGP est un protocole dont le routage peut être influencé par une politique d'administration. Ainsi, chaque AS a le loisir de propager (sélectivement ou non) une route à ses voisins. Dans la figure A.1, AS3 a pour politique de ne pas propager les routes annoncées par AS1 à AS5. Autrement dit, AS3 refuse de faire directement transiter le trafic entre AS5 et AS1 par son réseau. En conséquence, les machines situées dans AS5 doivent emprunter le chemin traversant AS6 et AS4. De cette façon, les politiques de routage des AS influent sur le choix des routes, et la meilleure route BGP n'est pas forcément la route topologiquement la plus courte.

Comme tous les autres protocole de routage (intra-domaine) IP, BGP utilise toujours la route la plus spécifique pour transmettre les paquets IP. La figure A.1 montre que lorsqu'AS2 annonce la

route 1.255.0.0/16, qui est plus spécifique que la route annoncée par AS1, tout le trafic destiné à la plage d'adresses IP 1.255.0.0-1.255.255.255 est automatiquement transmis en direction de AS2 et pas de AS1. Par contre, le trafic dont la destination se situe dans la partie du préfixe 1.0.0.0/8 non recouverte par 1.255.0.0/16, autrement dit le trafic dont la destination se trouve dans la plage d'adresses IP 1.0.0.0-1.254.255.255, est, quant à lui, transmis en direction de AS1.

A.1.3 BGP et les détournements de préfixes

La section A.1.2 a introduit les principes de fonctionnement de BGP. La section A.1.1 a illustré que la conception du protocole s'est fait dans un contexte de confiance mutuelle. De fait, comme le montre la façon dont ses opérations sont effectuées, BGP ne dispose d'aucun mécanisme de sécurité. En particulier, il est impossible pour un routeur BGP de vérifier

- qu'un AS est autorisé à annoncer un préfixe,
- l'identité d'un AS distant (c.-à-d. plus loin que ses voisins directs),
- qu'une route et/ou ses attributs sont corrects, complets, et inchangés.

En conséquence, toute route annoncée à un routeur BGP pourra toujours être insérée dans la table de routage, sauf si le numéro d'AS du routeur est dans le chemin d'AS, ou si la politique d'administration l'exclut explicitement. Cela implique que n'importe quel routeur de n'importe quel AS peut prétendre être n'importe quel (autre) AS, et annoncer n'importe quel préfixe, y compris des préfixes dont il n'est pas propriétaire.

Ces manquements rendent possible le phénomène de *détournement de préfixe*, phénomène étudié dans cette dissertation. Le **détournement de préfixe** est l'action d'absorber partiellement ou totalement le trafic à destination d'un autre AS par la propagation de routes BGP fausses ou erronées, rendue possible par l'existence de l'implicite confiance mutuelle entre tous les AS. Le détournement de préfixes peut être accidentel, c.-à-d. dû à une mauvaise configuration d'un routeur [Mahajan et al. 2002], ou intentionnelle [Ballani et al. 2007 ; Hu et al. 2007 ; Qiu et al. 2007 ; Tahara et al. 2008 ; Butler et al. 2010].

Indépendamment de l'intention de l'auteur du détournement, nous l'appellerons l'**AS pirate** (*hijacking AS*). De manière équivalente, la route annoncée par l'AS pirate est la **route piratée** (*hijacked route*). Le réseau dont la route a été piratée est l'**AS victime**, et sa route est la *route légitime*, ou **route originale**.

En procédant à une attaque par détournement, un pirate peut [Zheng et al. 2007]

- créer un trou noir dans le réseau (par exemple en rejetant les paquets),
- se faire passer pour la victime en imitant ses services (par exemple site web dupliqué),
- intercepter le trafic de la victime pour espionner (ou enregistrer) les données entrantes et sortantes, et ensuite renvoyer le trafic vers la victime (attaque de type homme du milieu),
- envoyer du pourriel, et/ou autres activités malveillantes.

A.1.3.1 Taxonomie des attaques par détournement de préfixe

Une attaque par détournement de préfixe peut s'effectuer de diverses façons. Dans cette section, nous présentons les différents vecteurs qui permettent ce type d'attaque. Un travail similaire a été présenté par [Lad et al. 2006] et [Hu et al. 2007], mais notre taxonomie rend compte des derniers développements dans le domaine.

Annonces simultanées

Une attaque par détournement de type annonces simultanées se produit lorsque l'AS pirate prétend être le propriétaire légitime du préfixe IP. Autrement dit, le pirate annonce le préfixe avec son numéro d'AS comme origine. Cette route fourbe est annoncée en même temps que l'annonce légitime du vrai propriétaire du préfixe IP. Ainsi, le préfixe paraît être annoncé depuis deux AS distincts. Cette situation est connue sous le nom de **préfixe à origines multiples** (*Multiple Origin AS*, **MOAS**). Puisque le pirate se présente comme l'origine du préfixe, le chemin d'AS de la route piratée sera plus petit que celui de la route légitime. La route piratée sera donc préférée – ne serait-ce que par les voisins directs du pirate – pour ce préfixe. La section A.2 se consacre à l'étude des MOAS.

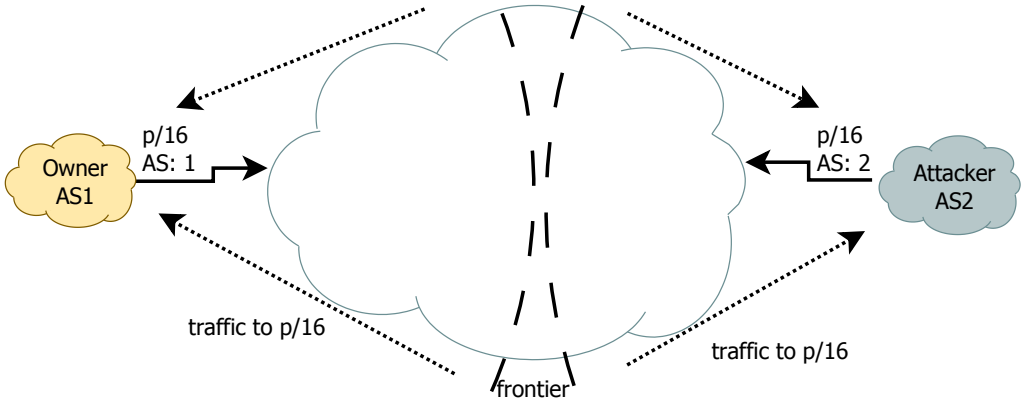


Figure A.2: Attaque par détournement de préfixe de type annonce simultanée

La figure A.2 illustre une situation d'attaque par annonce simultanée. On y voit que le préfixe p est annoncé par le propriétaire (AS1) et le pirate (AS2). Une frontière approximative en traits discontinus illustre la façon dont les réseaux topologiquement plus proches de l'une ou l'autre origine préfèrent la route qui leur sont la plus proche. Au final, Internet est divisé en deux zones distinctes qui renvoient le trafic soit vers le propriétaire, soit vers le pirate.

Recouvrement de préfixe

Une attaque par détournement de type recouvrement de préfixe se produit lorsque le pirate annonce un ou plusieurs préfixe(s) plus spécifique(s) que celui de la victime. Puisque le routage BGP se fait toujours vers le préfixe le plus spécifique, tout routeur qui accepte la route piratée va instantanément transmettre le trafic vers ce sous-préfixe vers l'AS pirate.

Si l'AS d'origine de la route plus spécifique diffère de l'AS d'origine de la route légitime, on parle de **sous-MOAS**, c.-à-d. que la route plus spécifique est annoncée simultanément par un autre AS que la route moins spécifique. La section A.3 étudiera ce phénomène.

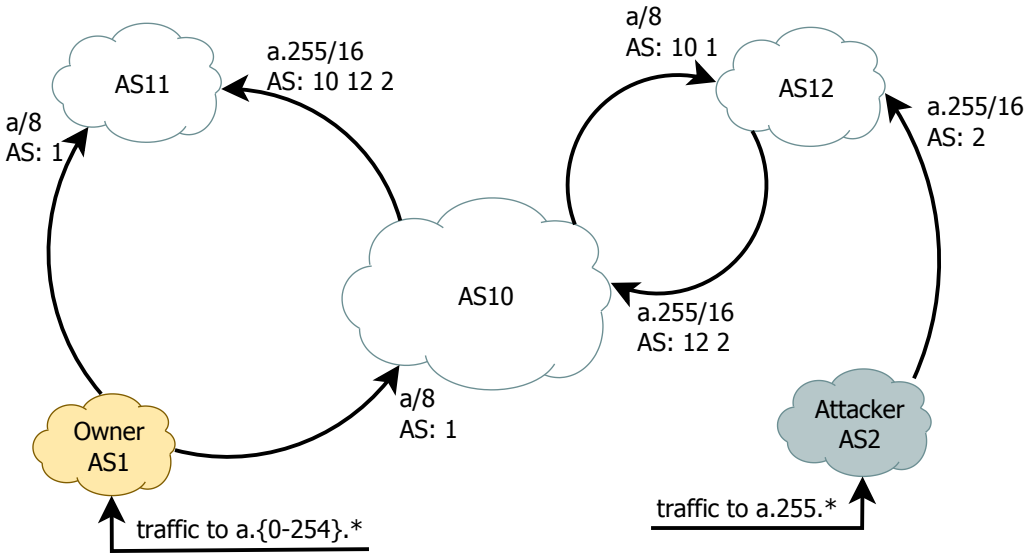


Figure A.3: Une attaque par détournement de type recouvrement de préfixe

La figure A.3 illustre ce type d'attaque. AS1, le propriétaire du préfixe $a/8$ annonce son préfixe sur Internet. Son voisin, AS10, accepte cette route et la propage à AS12. A ce moment, le trafic en direction de $a/8$ est transmis vers AS1. Lorsqu'AS2 détourne un sous-préfixe de $a/8$, ici $a.255/16$, il l'annonce à AS12. Ce dernier l'accepte et le propage à AS12, qui, à son tour, le propage à AS11. Puisque BGP transfère le trafic en utilisant la route la plus spécifique correspondante à l'adresse IP de destination, tout trafic en direction de $a.255/16$ sera transmis vers AS2. Cependant, le trafic vers le reste de $a/8$ sera toujours transmis en direction de AS1. En ciblant bien le sous-réseau à détourner, le pirate pourra s'attaquer à un service particulier de la victime. Par exemple, lors du détournement de YouTube, le pirate a annoncé le sous-réseau contenant les serveurs web et DNS de YouTube [RIPE NCC 2008a].

Liens d'AS forgés

L'AS pirate peut aussi décider de modifier les informations contenues dans le chemin d'AS. De ce fait, le chemin d'AS indique une **adjacence d'AS forgée**, c.-à-d. le chemin d'AS indique que deux AS sont voisins alors qu'ils ne le sont pas.

Avec ce type d'attaque, la longueur du chemin d'AS est plus grande qu'elle ne l'aurait été avec une attaque par annonces simultanées, mais elle est plus difficile à détecter car il n'y a pas de MOAS. Pour maximiser l'efficacité de l'attaque, le pirate doit se placer en seconde position dans le chemin d'AS, la quantité de trafic détournée en se trouvant plus loin diminuant fortement [Ballani et al. 2007]. Dans le cas particulier où le pirate décide d'utiliser l'ASN de la victime pour annoncer le préfixe de la victime, le lien forgé se situe entre l'origine et son réseau en amont. [Schlamp et al. 2013] étudie cette situation, baptisée **détournement d'AS** (*AS hijacking*).

La figure A.4 illustre une attaque par liens forgés sur le préfixe $p/16$, propriété d'AS1. AS1 annonce son préfixe à son voisin, AS10, qui le propage au reste du réseau. L'AS pirate, AS2, annonce le même préfixe, $p/16$, et insère AS1 dans le chemin d'AS. De ce fait, AS2 prétend être voisin direct de AS1, ce qui n'est pas le cas au vu de la topologie représentée à la figure A.4. Ce lien forgé est représenté par une ligne rouge en traits discontinus. Cependant, puisqu'AS12 est un voisin direct de l'AS pirate AS2, AS12 préférera probablement la route piratée à la route légitime, parce que la

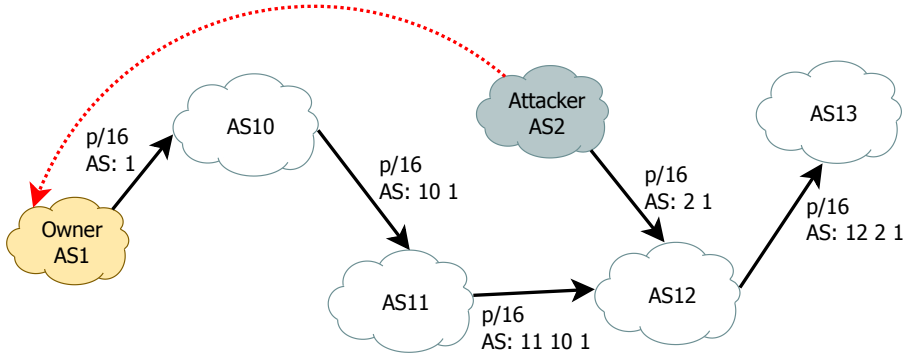


Figure A.4: Une attaque par détournement de préfixe de type liens d'AS forgés

longueur du chemin d'AS de la route piratée est plus courte que la longueur du chemin d'AS de la route légitime. Et donc, AS12 propagera la route piratée à son voisin, AS13.

Attaque de l'homme au milieu

Dans le but de pouvoir rediriger le trafic détourné vers son destinataire légitime, l'attaque de l'homme au milieu au niveau BGP présentée par [Pilosov et al. 2008] combine une attaque par recouvrement de préfixe avec une attaque par liens d'AS forgés. Dans un premier temps, l'attaquant doit identifier un chemin d'AS entre son propre AS et la source légitime du préfixe. Ce chemin, appelé le *chemin de retour*, sera utilisé pour renvoyer le trafic vers le propriétaire légitime. Ensuite, l'AS pirate annonce un ensemble de préfixes plus spécifiques que le préfixe de la victime et le recouvrant complètement. A ce préfixe, il adjoint un chemin d'AS forgé contenant l'ensemble des numéros d'AS des réseaux se trouvant sur le chemin de retour. Puisque cette route piratée est plus spécifique que la route légitime, elle est choisie par BGP pour transmettre le trafic (attaque par recouvrement). De plus, puisque tous les AS sur le chemin de retour sont dans le chemin d'AS, ceux-ci ne vont pas ajouter la route à leur table de routage. De cette façon, le chemin de retour ne sera pas affecté par la route piratée, et l'attaquant pourra simplement renvoyer le trafic sur ce chemin précis pour qu'il arrive à sa destination initiale. Ce type d'attaque, comme toutes les attaques de l'homme au milieu, est particulièrement furtive.

Une attaque de l'homme au milieu via BGP est illustrée aux figures A.5 et A.6. La figure A.5 illustre les routes normales vers AS1, origine de $p/16$. Si AS11 décide de faire une attaque de l'homme au milieu sur p, il doit pouvoir renvoyer le trafic vers AS1. Pour cela, il a besoin que la route entre lui-même et AS1 ne soit pas affectée, c.-à-d. que AS10 ne soit pas affecté. A la figure A.6, AS11 annonce les deux préfixes $/17$ qui recouvrent $p/16$, en ajoutant AS10 et AS1 au chemin d'AS. AS10 ignorera donc ces deux routes vers les $/17$, mais tous les autres réseaux vont les utiliser pour transmettre le trafic à destination de p. Ce trafic sera transmis vers AS11 qui, lui, pourra le transmettre à AS10 pour qu'il atteigne son origine légitime : AS1.

Espace dormant

L'**espace dormant** est composé de l'espace d'adresses IP allouées, mais non annoncées sur Internet. Par exemple, IBM dispose d'un préfixe $/8$, mais n'annonce publiquement que quelques préfixes, le plus large étant un $/16$. [Vervier et al. 2015] a montré que certains pirates ciblent ces réseaux non-annoncés afin de masquer leur identité et d'effectuer des opérations malveillantes sur Internet.

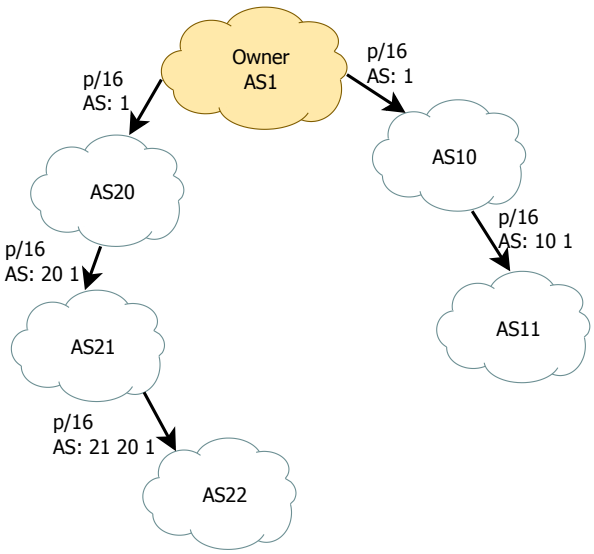


Figure A.5: Attaques de l'homme au milieu : routes normales menant vers à la victime

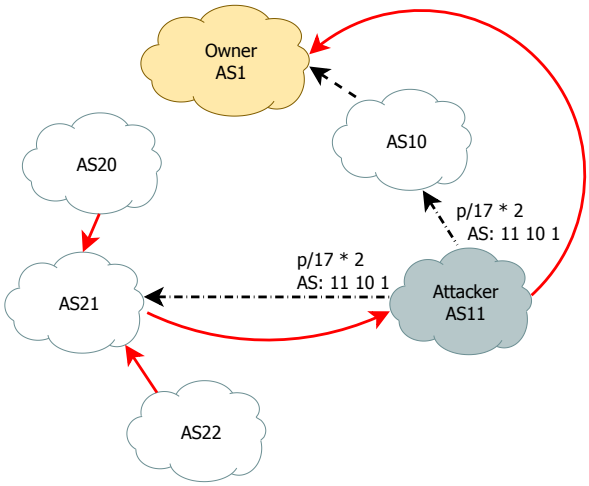


Figure A.6: Attaques de l'homme au milieu : routes piratées et chemin de retour

Espace non-alloué

Une grande partie de l'espace IP a été alloué. Le reste, appelé l'**espace noir** (*blackspace*), est toujours en attente d'affectation. En annonçant un préfixe qui fait partie de cet espace noir, un pirate peut envoyer du trafic sur Internet. De plus, étant donné que ces préfixes n'appartiennent à personne, ils ne sont en général pas surveillés, rendant l'attaque plus furtive. La section A.4 étudie ce phénomène en détails.

A.1.3.2 Occurrences d'attaques dans le monde réel

Un nombre important de cas de détournement de préfixes ont déjà eu lieu sur Internet. La section 2.2.2 les détaille. Nous résumons ici deux attaques.

La première est une attaque par détournement accidentel. En d'autres termes, c'est à la suite d'une erreur de configuration que l'attaque a eu lieu, l'AS pirate n'ayant, a priori, pas l'intention de nuire à sa victime. Le 24 février 2008, le gouvernement pakistanais décide d'interdire l'accès à YouTube [RIPE NCC 2008a]. YouTube est annoncé en tant qu'un préfixe /22. Pakistan Telecom décide d'appliquer la législation en utilisant BGP. Pour ce faire, il annonce un /24 qui contient les serveurs web et DNS de YouTube. Cependant, il n'a pas limité son annonce à son propre réseau, mais l'a aussi propagé à son fournisseur d'accès en amont, PCCW Global, qui accepte la route et la propage au reste du monde. Pendant approximativement 1h30, tout le trafic mondial à destination de YouTube arrive sur le réseau de Pakistan Telecom. YouTube a contre-attaqué en annonçant les huit préfixes /25. Au final, 2h30 après le début de l'attaque, PCCW Global retire les routes de Pakistan Telecom, et YouTube reprend sa configuration initiale.

La seconde est une attaque par détournement intentionnel. En d'autres termes, ici, c'est bien un acte délibéré commis par des criminels du cyber-espace. Entre février et mai 2014, 19 fournisseurs d'accès Internet ont été détournés dans le but d'utiliser la puissance de minage de crypto-monnaie présente sur ces réseaux [Litke et al. 2014]. Les pirates ont réussi à détourner les machines de minage vers leurs propres serveurs distribuant du travail. Le minage était alors effectué, mais les machines ne recevaient plus de récompense pour leurs efforts, qui étaient mis en poche par les pirates. Au

total, il y a eu 22 attaques par détournement, chacune d'une durée moyenne de 30 secondes. Les pirates ont réussi à voler l'équivalent de 83 000 dollars US en crypto-monnaie.

A.1.4 Sécurisation de BGP

Dans la section A.1.3, nous avons présenté le détournement de préfixe et les diverses façons de porter une telle attaque. Nous avons aussi montré que ces attaques sont bien réelles, et que l'infrastructure de routage inter-domaine est, en fait, assez fragile. Dans cette section, nous introduisons les diverses techniques qui ont été proposées afin de pallier aux manquements du protocole BGP et, ainsi, de rendre impossible les attaques par détournement.

La première ligne de défense consiste à appliquer un certain nombre de *bonnes pratiques* BGP. Il s'agit d'implémenter une politique de routage défensive sur les routeurs BGP, de manière à rendre la propagation de routes erronées plus difficile. Ces pratiques sont détaillées à la section 2.3.2, et ne sont en aucun cas obligatoires. De plus, leur implémentation nécessite un travail important, que ce soit au niveau de la conception initiale de la politique, ou de sa maintenance. Ce travail à fournir est, par ailleurs, bien trop large pour un opérateur dans le coeur du réseau (tier 1).

Un certain nombre de propositions ont été faites dans le but de modifier BGP et d'y ajouter des opérations permettant de certifier et de vérifier l'information propagée entre les AS. La proposition la plus complète, présentée dans [Kent et al. 2000], est **Secure-BGP** (S-BGP) : un protocole conçu pour remplacer et prendre en charge le travail de BGP, tout en prenant en compte le besoin de sécuriser les informations de routage. S-BGP utilise l'infrastructure à clé publique et les certificats X.509 afin de vérifier qu'une organisation est bien le propriétaire d'un préfixe. Ainsi, des certificats lient les numéros d'AS avec leur organisations, et les organisations avec leurs routeurs. Le deuxième élément clé de S-BGP est composé des *attestations*, dont il existe deux types. Les *attestations d'adresse* sont signées par le propriétaire d'un préfixe et autorisent un ensemble d'AS à annoncer le préfixe. Les *attestations de route* sont signées par chaque AS propageant une route, et en certifiant l'AS vers lequel la route a été propagée. Le troisième élément est l'utilisation d'IPsec pour sécuriser la connexion entre deux routeurs BGP pairs. Au final, S-BGP permet de certifier qu'un AS est l'origine légitime d'une route, et que le chemin d'AS correspond bien à la topologie réelle du réseau. Cependant, S-BGP nécessite que chaque propriétaire de préfixe ou d'AS, ainsi que chaque routeur, possède au moins un certificat de signature. Ces certificats devront à chaque fois être validés par les routeurs, ce qui ferait augmenter le temps de convergence d'une route. De plus, le stockage nécessaire à ces divers éléments est non négligeable, et la capacité de calcul requise pour toutes ces opérations cryptographiques à grande échelle n'est pas disponible sur le matériel actuel. A l'heure actuelle, le déploiement de S-BGP est très improbable.

Afin d'alléger les conditions nécessaires au déploiement d'une version sécurisée de BGP, d'abord **secure-origin BGP** (soBGP) [White 2003], et ensuite **Pretty Secure BGP** (psBGP) [van Oorschot et al. 2007] essayent de trouver un juste milieu entre la sécurisation complète proposée par S-BGP, et un niveau de sécurité acceptable dans des conditions réelles. De manière générale, ces protocoles réduisent le nombre de certifications nécessaires (et donc, en même temps, la sécurité qui en résulte), mais offrent toujours plus de certitudes que BGP. De nombreuses questions se posent aussi sur l'utilisation et la déployabilité à grande échelle de ces protocoles, et, au final, il semble peu probable qu'une version sécurisée de BGP soit adoptée à court ou moyen terme.

A l'heure actuelle, **RPKI** (Resource Public Key Infrastructure) paraît être la seule à satisfaire au déploiement à grande échelle. Cette infrastructure est activement soutenue par des acteurs majeurs

d'Internet. RPKI n'est pas un protocole subvenant à BGP, mais une infrastructure parallèle qui permet aux routeurs BGP participants de vérifier qu'un AS est autorisé à annoncer un préfixe via des certificats spécifiques. En août 2015, 6% de l'espace IPv4, et 9% de l'espace IPv6 étaient sécurisés par RPKI [RPKI Dashboard].

A.1.5 Détection des détournements de préfixes

Dans la section A.1.4, nous avons vu que, bien qu'il existe des propositions afin de sécuriser complètement l'infrastructure de routage inter-domaine, il est peu probable qu'une d'entre elles soit adoptée dans un futur proche. Même si des signes encourageants indiquent que RPKI pourrait un jour couvrir complètement l'espace IP, le nombre de routes actuellement certifiées par RPKI est assez faible. De plus, RPKI n'offre que la certitude qu'un AS peut annoncer une route. En d'autre termes, RPKI est plus un système empêchant une attaque par détournement accidentelle qu'un moyen de se défendre contre une attaque intentionnelle. Bien qu'il s'agisse d'une première étape nécessaire vers la sécurisation de l'infrastructure fondamentale d'Internet, elle n'est pas suffisante, ni aujourd'hui, ni demain.

Pour cette raison, il est nécessaire, et même primordial, d'arriver à détecter les attaques par détournement. Même si celles-ci ne peuvent pas être empêchées, des contre-mesures peuvent être utilisées afin de limiter leurs effets. Plusieurs méthodes ont été proposées à cette fin. Elles sont en général classifiées selon la méthode de détection qu'elles utilisent. D'un côté, il y a les méthodes basées sur le **plan de contrôle**, c.-à-d. sur les informations de routage échangées entre les routeurs BGP. De l'autre côté, il y a les méthodes basées sur le **plan de données**, c.-à-d. sur le chemin suivi par les paquets IP entre un site d'observation et un site surveillé. Plus récemment, ces deux approches ont été intégrées dans des systèmes de détection plus complexes.

Les méthodes basées sur le plan de contrôle, comme [Ballani et al. 2007 ; Lad et al. 2006 ; Gao 2001], d'une manière générale, créent un modèle d'Internet qui en représente le comportement normal souhaité. Lorsque la réalité diffère de ce modèle, une alerte est générée. La complexité et la diversité du modèle créé est l'élément permettant une bonne qualité de détection. En général, ces techniques peuvent être utilisées directement, parfois de manière commerciale : il suffit à un propriétaire de préfixe de s'inscrire au service, en général via un site web, pour recevoir des notifications d'anomalies, ce qui se fait, en général, par courrier électronique. Les méthodes basées sur le plan de données, comme [Hu et al. 2007 ; Zheng et al. 2007 ; Hong et al. 2009 ; Zhang et al. 2008], fonctionnent grâce à la mesure continue de la topologie d'Internet, et/ou des machines disponibles sur le réseau surveillé. De fait, lorsqu'une attaque par détournement a lieu, un changement de topologie significatif a lieu, et le réseau pirate est différent du réseau légitime. Le type de mesure utilisée, ainsi que la diversité des facteurs considérés permettent une bonne qualité de détection. En général, ces techniques nécessitent d'être déployées par le propriétaire d'un préfixe qui réalise alors ses propres mesures. De plus, la mesure continue génère du trafic réseau, et n'est donc pas utilisable à grande échelle. Ces deux types d'approches ont récemment été combinées avec Argus [Xiang et al. 2011 ; Shi et al. 2012], qui utilise un module pour surveiller le plan de contrôle, et ensuite un autre module pour mesurer le réseau où l'anomalie a été détectée. L'état d'alerte est basé sur la corrélation qui existe entre le résultat de ces deux processus.

Malheureusement, ces outils, même les plus perfectionnés, génèrent un nombre très important d'alertes, qui correspondent principalement à des évènements réseaux bénins, tels que l'ingénierie de trafic. Pour les propriétaires de ces espaces IP, ça ne pose pas de réel problème parce qu'ils connaissent le comportement que leur réseau doit adopter. En connaissant cette réalité de terrain,

ils peuvent très facilement écarter les faux positifs, et, si besoin, prendre les mesures nécessaires. De plus, étant donné qu'ils ne surveillent que leur propre réseau, le nombre d'alerte qu'ils reçoivent est suffisamment bas, pour ne pas être submergés par les alertes. En tant qu'observateur externe, nous ne disposons pas de la réalité de terrain nécessaire pour écarter les faux positifs. En plus, nous observons l'entièreté d'Internet. Il en résulte que le temps et les ressources réseaux nécessaire à l'analyse de ces alertes rendent impossible l'utilisation systématique des techniques de détection d'attaques par détournement de préfixe existantes. De ce fait, différencier une attaque par détournement d'un évènement de routage légitime est non trivial et demande une quantité de travail importante. Nous illustrerons ce fait dans la section A.2.

A.1.6 Contributions

Dans la section A.1.5, nous avons vu que malgré qu'il existe un grand nombre de techniques de détection d'attaques par détournement de préfixes, celles-ci ne permettent pas de surveiller Internet dans sa globalité ; et, par extension, elles ne permettent pas l'étude de ce type d'attaque. En effet, les utilisateurs cibles de ces applications sont les opérateurs réseaux qui n'ont besoin de surveiller que leurs propres ressources réseaux. De ce fait, ils peuvent aisément filtrer les alertes et discerner les faux positifs des vrais positifs grâce à la réalité de terrain qu'ils imposent. Pour des observateurs externes tels que nous, examiner chaque alerte afin de discerner une attaque par détournement d'un évènement bénin résultant d'un changement de configuration n'est pas trivial.

De plus, pour examiner systématiquement et efficacement chaque évènement de routage afin d'en trouver la cause profonde réelle, nous devons trouver un moyen pour connaître cette réalité de terrain. Lorsqu'on doit évaluer la qualité d'une nouvelle technique de détection en la présentant dans un papier, un retour direct des opérateurs est, en général, obtenu en entrant en contact avec eux directement. Bien sûr, cette solution n'est possible que sur le court terme, et dépend totalement de la volonté et du temps disponible des divers opérateurs réseaux pour répondre à ce genre de questions. Au final, cette approche n'est ni automatisable, ni possible sur le long terme.

En conséquence, vu qu'elle est nécessaire pour faire la différence entre une attaque et un évènement de routage bénin, la réalité de terrain doit être obtenue sans avoir besoin de contacter l'opérateur réseau responsable. L'approche que nous allons utiliser dans cette dissertation est l'utilisation d'un ensemble varié de sources de données afin d'observer un unique évènement sous une multitude de perspectives. Ces données auxiliaires doivent nous informer sur l'identité des parties impliquées, et sur ce qu'elles faisaient à ce moment précis. Si possible, il faudrait aussi comparer ces identités et comportements avec ce qui se passait avant, afin d'observer un éventuel changement. En d'autres termes, examiner un évènement de routage unique afin d'en trouver la cause de manière indubitable peut nécessiter plusieurs jours, voire semaines. Ceci nous amène donc à deux problèmes distincts. Le premier est de trouver les sources de données auxiliaires intéressantes pour étudier les évènements réseaux, et d'acquérir les droits de consultation. Le second est de réduire le nombre d'évènements à analyser de manière à limiter leur nombre.

Afin d'obtenir la réalité de terrain, nous avons mis en place une collaboration, détaillée à la section A.1.7. Grâce à celle-ci, nous avons accès à des sources de données de sécurité, qui nous renseignent, en particulier, sur l'existence d'activités d'envoi de pourriels (*spam*), d'arnaques en ligne (*scam*), et de maliciels (*malware*). Nous avons aussi accès à des données NetFlows qui nous permettent d'identifier les applications utilisées par les réseaux étudiés. En combinant toutes ces informations, nous avons donc une idée assez précise de l'activité réseau générée par un réseau précis à un moment donné. De plus, nous utilisons les IRRs (Internet Routing Registries), source de données qui nous

informe, entre autres, sur les informations d'enregistrement des plages d'adresse IP et des AS. Nous utilisons cette information afin de savoir qui se trouve derrière un réseau donné, et les liens existants entre les différents réseaux.

Afin de réduire le nombre de cas à analyser, une grande partie de cette dissertation est dédiée à l'élimination automatique des faux positifs. En partant d'une liste d'événements BGP suspects, similaire à ce qu'un outil de détection de dernière génération nous donnerait, nous analysons manuellement un grand nombre de cas faux positifs. De cette analyse, nous mettons en avant un nombre de tendances globales, c.-à-d. des constructions standards et variées qui rendent compte des pratiques standards et variées de configurations BGP. Ensuite, en considérant chacune de ces structures dans le cadre d'une attaque par détournement de préfixes, nous évaluons l'impact de cette structure au niveau de la sécurité BGP, et nous pouvons donc savoir si elle représente une menace globale. Si ce n'est pas le cas, nous pouvons éliminer les alertes résultant de cette construction, en la considérant donc comme un groupe de faux positifs.

Dans cette dissertation, nous analysons trois façons d'effectuer une attaque par détournement. A savoir, nous étudierons :

1. les annonces simultanées, où le pirate et le propriétaire légitime du préfixe annoncent tous les deux le même préfixe depuis deux AS différents (MOAS), à la section A.2 ;

2. les recouvrements de préfixes, où le pirate annonce un préfixe plus spécifique que celui annoncé par le propriétaire, à la section A.3 ;

3. l'espace noir où un pirate prend contrôle d'une plage adresses IP non alloués, à la section A.4.

A.1.7 Collaborations

A la section A.1.6, nous avons annoncé avoir mis en place une collaboration grâce à laquelle nous pouvons estimer la réalité de terrain nécessaire afin d'étudier des évènements de routage. Cette collaboration nous donne accès à un ensemble diversifié de données que nous détaillons ici.

Notre partenariat avec Symantec Research Labs s'inscrit dans le but de se concentrer sur les attaques par détournement malveillantes. En particulier, le travail de [Vervier 2014] se concentre sur le phénomène de *spammeurs par survol* (*fly-by spammers*), mis au jour par [Ramachandran et al. 2006], qui détournent un bloc IP afin d'envoyer du pourriel, et qui stoppent ensuite immédiatement leur attaque. Le but principal du travail de Vervier est d'évaluer l'importance de ce phénomène et de caractériser le mode opératoire de ces pirates. Il a montré que, sur une période de deux ans, plus de 2 000 préfixes ont été détournés de manière intentionnelle et malveillante, et que les réseaux victimes étaient en général dormants, c.-à-d. des réseaux alloués mais non-annoncés.

Notre partenariat avec Technische Universität München (TUM) s'inscrit dans le but d'enrichir les données de routage BGP avec des données relatives aux applications. En particulier, la TUM collecte des flux NetFlows sur le réseau scientifique de Munich (Münchner Wissenschaftsnetz). Dans [Schlamp 2015], l'auteur se concentre sur certaines formes d'attaques par détournement qui ne sont pas surveillées par les outils existants, en particulier sur le cas des attaques sous-MOAS. Notre travail collaboratif sur ce point précis est, par ailleurs, présenté à la section A.3. De plus, Schlamp se concentre sur les ressources réseaux dont les informations de contact sont expirées, et qui sont, de ce fait, plus facilement vulnérables.

A.2 Préfixes à origines multiples (MOAS)

Dans cette section, nous nous focalisons sur les annonces simultanées de préfixes. Cette situation se produit lorsqu'un unique préfixe p est annoncé par deux AS distincts. Elle est aussi connue sous le nom de *préfixe à origines multiples* (*Multiple Origin AS*, **MOAS**). Bien que [RFC1930] déconseille ce cas de figure, les analyses de MOAS [Zhao et al. 2001 ; Chin 2007] ont illustré un certain nombre de situations dans lesquelles les MOAS sont utilisés à des fins légitimes. Ces situations sont principalement

- la *multirésidence* (*multihoming*), qui, par exemple, survient lorsqu'un réseau est connecté à Internet via deux FAIs différents et que ces FAIs prennent en charge l'administration BGP pour ce réseau ;
- l'*anycast* ;
- les préfixes liés aux *points d'échange Internet*, qui peuvent être annoncés par les divers réseaux qui y sont pairs ;
- les compagnies *multinationales* qui possèdent des installations à plusieurs points géographiques différents, et, de manière similaire les *centres de données* ;
- les liens Internet satellitaires.

Au vu de leur utilisation, toutes ces instances sont considérées comme étant de *longue durée* (c.-à-d. qui durent plus longtemps que 24 heures). Dans le même temps, [Zhao et al. 2001] souligne aussi un nombre important de MOAS de *courte durée* dont les raisons ne sont pas claires, et attribuées, à défaut d'autre explication, à des erreurs de configuration.

Dans un premier temps, notre démarche consiste à reproduire l'expérience de [Zhao et al. 2001] afin de trouver ces évènements de courte durée, et de les analyser pour voir s'ils sont le résultat d'attaques par détournement. Dans le but de formaliser le problème, considérons la situation illustrée par la figure A.7. Nous y avons un préfixe p, qui est annoncé, à divers instants dans le temps, par 3 AS différents. Il est annoncé par AS_1 pendant la durée $]t_0, t_2[\cup]t_7, t_9[$, par AS_2 pendant la durée $]t_1, t_3[\cup]t_5, t_8[$, et par AS_3 pendant la durée $]t_4, t_6[$. En appliquant la même définition de MOAS que [Zhao et al. 2001], que nous appellerons **MOAS par évènement**, il y a trois situations de MOAS pour le préfixe p qui sont illustrées à la figure A.7. La première est pendant $]t_1, t_2[$, la seconde pendant $]t_5, t_6[$, et la troisième pendant $]t_7, t_8[$. Les durées de ces MOAS, appelées **durées des MOAS par évènements**, sont simplement la durée de ces intervalles de temps, soit, respectivement, $t_2 - t_1$, $t_6 - t_5$, et $t_8 - t_7$.

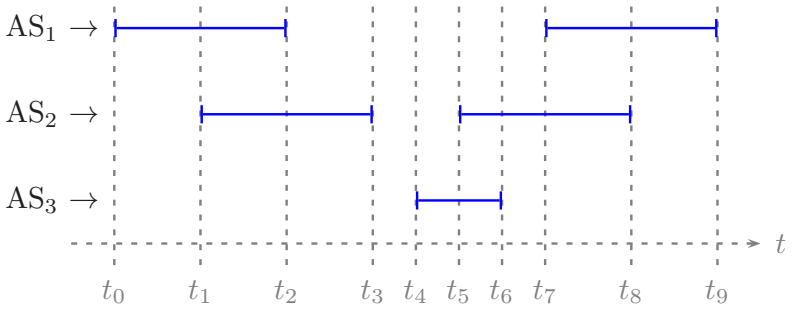


Figure A.7: Exemple d'annones BGP pour un préfixe p

Les résultats de l'application de cette métrique aux données de routage BGP collectées par [RIPE RIS] (colleteur : Amsterdam/rrc00) pendant l'entièreté de l'année 2012 sont disponible dans la table A.1. De manière similaire aux résultats de [Zhao et al. 2001 ; Chin 2007], la moyenne de la durée des MOAS (par évènements) est de 48 jours. Cependant, le coefficient de variation de la distribution est de 1.88, indiquant une forte variabilité. Et, en effet, comme remarqué par [Zhao et al. 2001], la médiane est de seulement 26h, ce qui nous indique qu'il y a une grande quantité d'évènements de (très) courte durée. La figure A.8 révèle la distribution complète de la durée des MOAS par évènement. On peut y voir que la moitié des évènements durent moins qu'un jour.

	μ	q_{50}/μ	q_{50}
Tous les MOAS en 2012	48j	1.88	26h

Table A.1: Durée des MOAS par évènements au long de l'année 2012

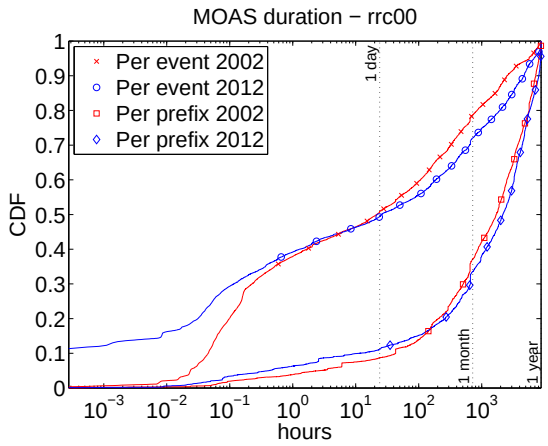


Figure A.8: Durée des MOAS

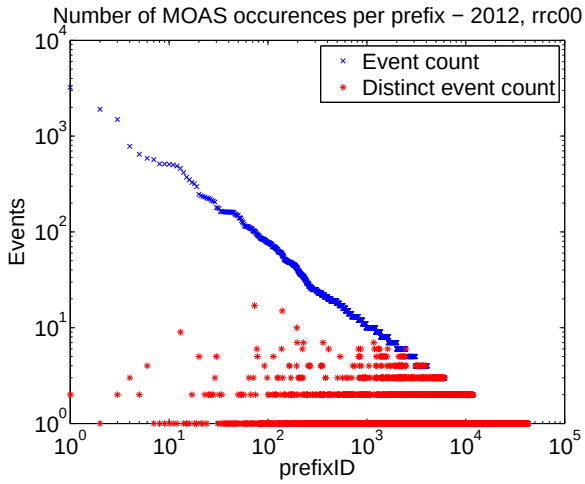


Figure A.9: Nombre de MOAS par préfixe en 2012

En revenant sur la situation illustrée à la figure A.7, on remarque que, des trois évènements MOAS, deux sont identiques. En effet, le MOAS de p pendant $]t_1, t_2[$, ainsi que celui pendant $]t_7, t_8[$ sont causés par les deux mêmes AS : AS_1 et AS_2. On dit que deux MOAS sont **distincts** si les AS qui les causent sont différents. Ainsi, pour le préfixe p de la figure A.7, il y a trois MOAS, dont seulement deux sont distincts. De plus, la **durée de MOAS par préfixe** est la somme de la durée des évènements MOAS pour ce préfixe. Ainsi, pour le préfixe p à la figure A.7, cette durée est $t_2 - t_1 + t_6 - t_5 + t_8 - t_7$. La figure A.8 montre la distribution de la durée des MOAS par préfixes. Contrairement à la courbe obtenue plus tôt, nous voyons qu'ici, le nombre de MOAS par préfixes plus court qu'un jour sont de l'ordre de 10%. Cela suggère que certains préfixes ont beaucoup d'évènements MOAS de courte durée.

La figure A.9 indique le nombre de MOAS pour un préfixe (croix bleues), ainsi que le nombre de MOAS *distincts* pour ce même préfixe (étoiles rouges). On voit en effet que pour approximativement un quart des préfixes présentant un MOAS, le nombre d'évènements distincts est entre 10 et 1000 fois plus petit que le nombre d'évènements. Par conséquent, **les évènements MOAS de courte durée ne peuvent pas être attribués à des erreurs de configuration**, comme dit par [Zhao et al. 2001]. Et ce parce que s'il s'agissait d'erreurs de configuration, les préfixes affectés ne seraient pas tout le temps les mêmes, et il n'y aurait donc pas une telle différence entre les fonctions de répartition représentées à la figure A.8. Au final, considérer les MOAS en tant que *groupe d'évènements liés*

à un préfixe, et pas en tant qu'évènements indépendants, nous permet de *refléter la configuration d'un réseau*, et ce indépendamment de la variabilité de la perception de ce choix inhérente à la localisation du point de collecte de données de routage.

A.2.1 Taxonomie des MOAS

En analysant la topologie au niveau AS des préfixes MOAS, nous avons mis en avant un ensemble de schémas topologiques de MOAS que nous présentons ici.

Le premier schéma, que nous appelons **MOAS pairs** (*peering MOAS*), est illustré à la figure A.10. Dans ce schéma, le propriétaire du préfixe, AS10, annonce son préfixe sur Internet. Son FAI (ou, de manière plus général, un voisin direct), AS20, l'annonce simultanément, ce qui crée le MOAS. Au final, nous considérons ce cas de figure comme un **faux** MOAS parce qu'il n'y a pas de bénéfice pour le propriétaire d'avoir son préfixe annoncé par son pair direct. En effet, si le pair arrête d'annoncer le préfixe, il n'y a pas de perte de connectivité, et donc pas de différence au niveau du réseau. La proportion de ce schéma dans tous les MOAS est d'approximativement 75%.

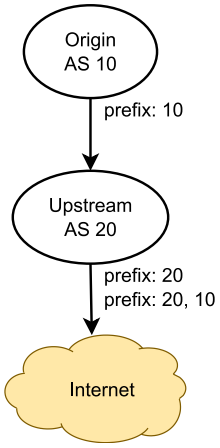


Figure A.10: Premier schéma MOAS : le MOAS pair (*peering MOAS*)

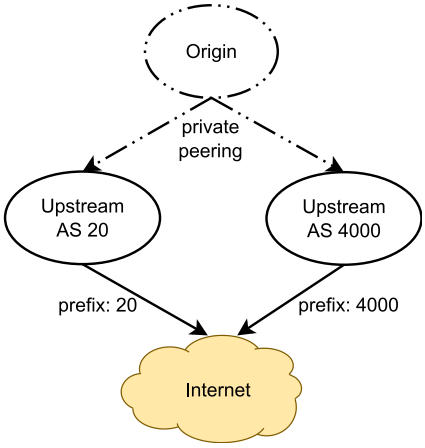


Figure A.11: Second schéma MOAS : le MOAS classique (*classical MOAS*)

Le second schéma, le **MOAS classique** (*classical MOAS*), est illustré à la figure A.11. Dans ce cas ci, il y a plusieurs chemins d'AS *distincts* menant vers l'origine du préfixe. Ce schéma représente l'utilisation classique faite du MOAS, comme décrite par [Zhao et al. 2001 ; Chin 2007]. Le nombre de cas de MOAS respectant ce schéma varie entre 25 et 30%. Ce phénomène est assez remarquable car le schéma correspondant à la compréhension usuelle associée aux MOAS ne correspond, au final, qu'à moins d'un tier de tous les cas de MOAS.

Le troisième et dernier schéma, **MOAS de-même** (*me-too MOAS*), est illustré à la figure A.12. Il combine les deux autres schémas en un seul. Sur le côté gauche de la figure A.12, on voit un MOAS pair, sur le premier niveau, on voit un MOAS classique. La proportion de ce cas varie entre 3 et 5% du nombre de MOAS total.

La figure A.13 montre que les occurrences, par préfixes, de ces schémas de MOAS sont principalement de longue durée, et donc représentent bien des états de configurations stables des réseaux BGP.

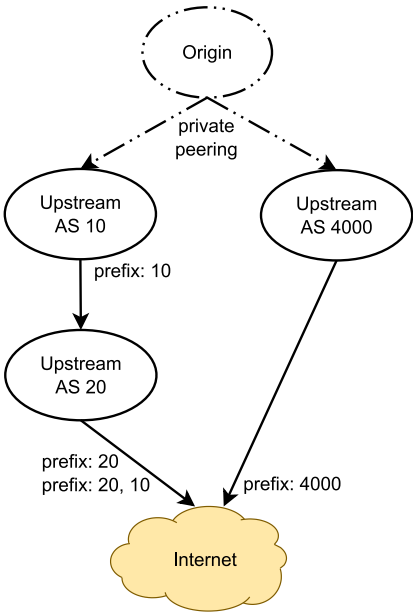


Figure A.12: Troisième schéma MOAS : le MOAS de-même (*me-too MOAS*)

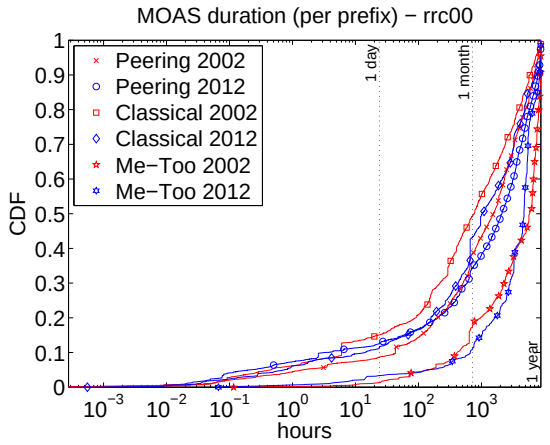


Figure A.13: Durée par préfixe des schémas MOAS

A.2.2 Elimination des faux positifs dûs aux MOAS

Dans cette section, nous étendons la taxonomie des MOAS présentée à la section A.2.1 afin de construire un système de filtres qui permettent d'éliminer un grand nombre d'alertes dues aux MOAS. Dans ce but, nous allons considérer chaque schéma de MOAS dans un contexte d'attaque par détournement afin d'estimer le risque posé par ce schéma au niveau de la sécurité inter-domaine.

Premièrement, considérons le cas des MOAS pairs, comme illustré à la figure A.10. Un fournisseur d'accès (AS20) est toujours sur le chemin entre son client (AS10) et Internet. En d'autres termes, AS20 peut surveiller et modifier le trafic à destination de AS10 sans avoir besoin d'effectuer une attaque par détournement. Pour cette raison, nous éliminons les alertes dues aux MOAS fournisseurs-clients (*provider-customer MOASesASes*). Ce raisonnement peut être étendu dans le cas de tout réseau pair. En admettant qu'AS20 et AS10 soient de simple pairs – ce qui signifie, par exemple, qu'il y a un autre lien entre AS10 et Internet que celui passant par AS20 dans la figure A.10 – l'efficacité de l'attaque par détournement sera limitée par les réseaux voisins du pair. En d'autres termes, l'efficacité de l'attaque serait très faible. Pour ces raisons, nous estimons que toutes les alertes résultant de MOAS pairs peuvent être considérées comme des faux positifs.

Un raisonnement similaire au niveau de la topologie ne peut malheureusement pas être fait pour les deux autres schémas de MOAS. Cependant, un résultat de la littérature, [Karlin et al. 2006], nous informe que 24 heures sont suffisantes pour qu'un opérateur réseau décide d'activement s'opposer à une annonce BGP erronée. Dans le même temps, la figure A.13 montre que la majorité des évènements MOAS sont de longue durée. Ainsi, nous éliminons les alertes MOAS pour les évènements suffisamment longs.

D'une manière générale, notre méthode s'appuie sur les techniques standard de routage pour créer une classification qui nous permet d'éliminer une certaine quantité d'évènements certainement bénins de notre analyse. En moyenne, nous observons une **réduction de 80% des cas de MOAS à analyser** comparé aux techniques existantes, par exemple [Lad et al. 2006]. Tous les cas de MOAS restants à analyser sont soit des MOAS classiques, soit des MOAS de-même.

A.2.3 Analyse des MOAS suspects

Dans la section A.2.2, nous avons présenté une série d'heuristiques qui nous permettent d'éliminer un grand nombre d'alertes dues à des MOAS. La figure A.14 présente le système que nous avons mis en place afin d'analyser les MOAS restants. En résumé, nous combinons des données de routage BGP, des données collectées de manière passive depuis des pièges à pourriels (*spamtraps*), des données NetFlow collectées sur un large réseau académique, et les données d'enregistrement des IRR (Internet Routing Registries).

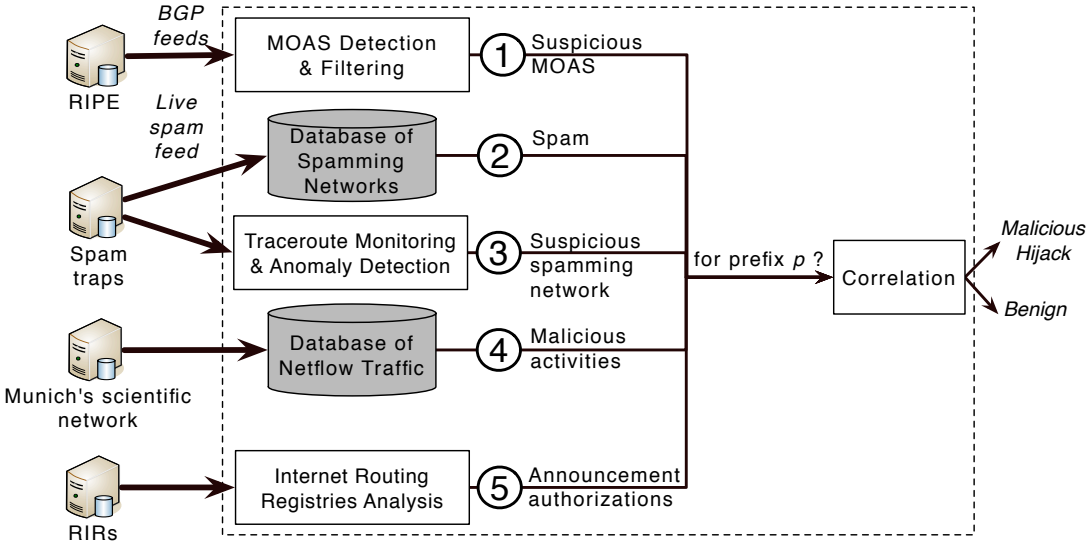


Figure A.14: Architecture du système d'analyse de MOAS

Afin de localiser les évènements BGP qui posent un problème sécuritaire, nous corrélons la sortie des heuristiques de la section A.2.2 (① dans la figure A.14) avec des données de pourriels ②, ainsi que des traceroutes effectués vers les réseaux envoyant ces pourriels ③. Nous utilisons aussi les traces applicatives laissées par ces réseaux ④ sur le réseau scientifique de Munich. Et, enfin, nous utilisons aussi les données d'enregistrement des IRR pour confirmer (ou infirmer) nos conclusions ⑤. Les sources de données de pourriels (② et ③) proviennent de nos partenaires à Symantec Research Labs. Les traces réseaux ④ sont collectées par nos partenaires de la Techniche Universität München sur le Münchner Wissenschaftsnetz, un réseau comprenant plus de 80 000 machines et dont le volume de trafic mensuel s'approche du pétaoctet.

A.2.4 Etude de cas : le cas bulgare

Dans cette section, nous présentons le résultat de l'utilisation du système présenté à la section A.2.3 pendant le mois de février 2013. Le 3 février 2013, notre système nous a informé d'un incident en Bulgarie : un certain nombre de MOAS étaient observés en même temps qu'un nombre important de pourriels reçus. Nous présentons ici notre analyse.

A.2.4.1 Première analyse

Lors de notre première analyse, nous utilisons une approche similaire à celle employée dans la littérature pour analyser les attaques par détournement. A des fins didactiques, nous avons divisé cette analyse en 4 phases distinctes, chacune correspondant à 4 comportements BGP différents. Ces phases sont illustrées dans le haut de la figure A.15.

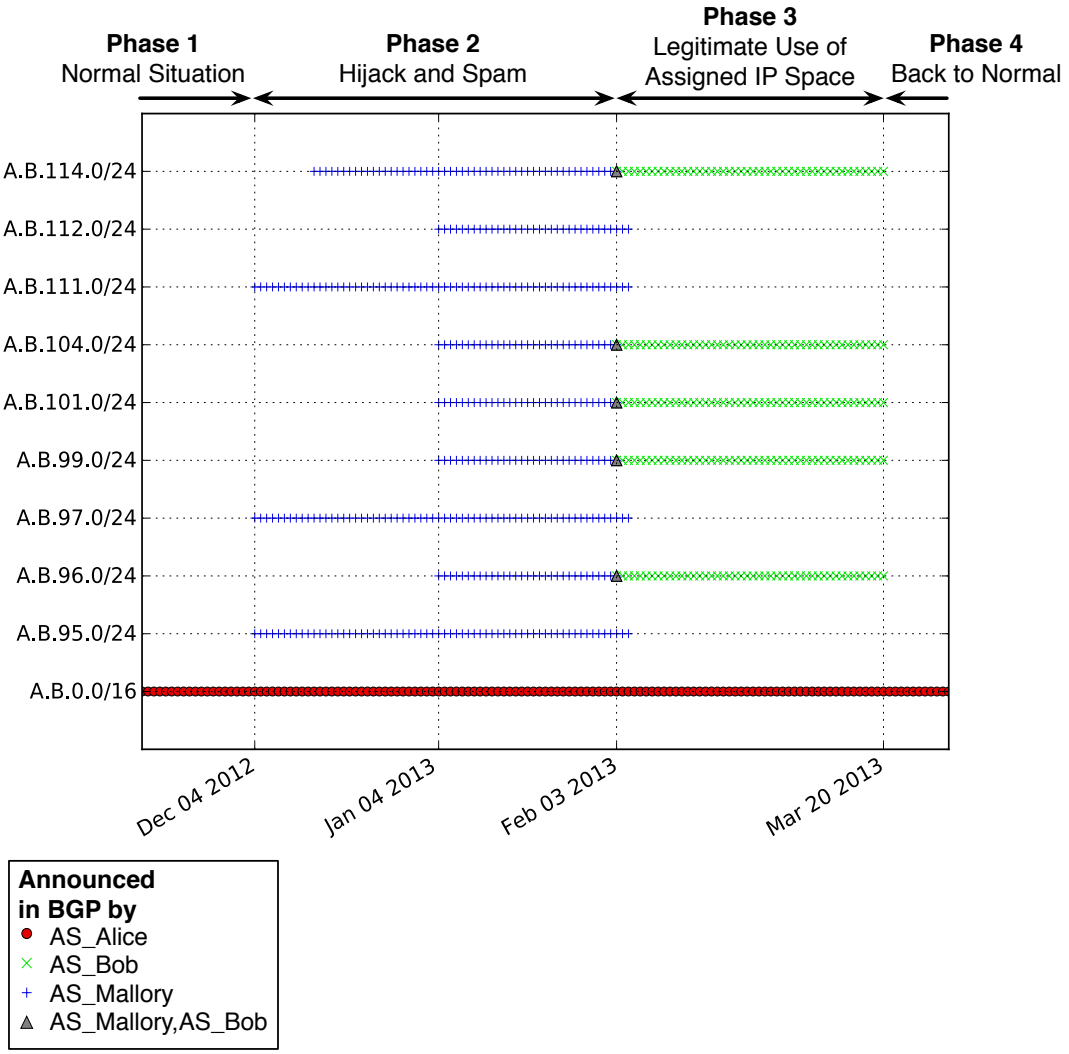


Figure A.15: Annonces BGP pour le cas bulgare

Phase 1 : situation normale
Depuis 2008, le préfixe A.B.0.0/16 est annoncé par Alice, un petit FAI bulgare, connu pour ses services d'hébergement. Aucun préfixe plus spécifique n'est annoncé.

Phase 2 : détournement et pourriels
Le 4 décembre 2012, Mallory annonce neuf préfixes plus spécifiques (/24) que celui d'Alice, alors qu'Alice continue à annoncer son /16. Les informations d'enregistrement de Mallory nous informent qu'il est, probablement, un fournisseur de serveurs virtuels, lui aussi situé en Bulgarie. Nous n'avons cependant pas été en mesure de trouver son site web.

La figure A.16 montre le pourriel reçu par Symantec.cloud depuis des adresses IP incluses dans les préfixes annoncés par Mallory. Cette figure montre aussi les adresses IP de ces préfixes contenues dans les listes noires de pourriels fournies par [Uceprotect]. Comme illustré, il existe une parfaite corrélation entre les annonces BGP, le pourriel, et la liste noire. De plus, le pourriel reçu ne contient pas d'éléments indiquant qu'il a été envoyé par un robot à pourriel (*spambot*), ce qui signifie que les machines émettant ce courrier non désirable ont probablement été installées par les spammeurs.

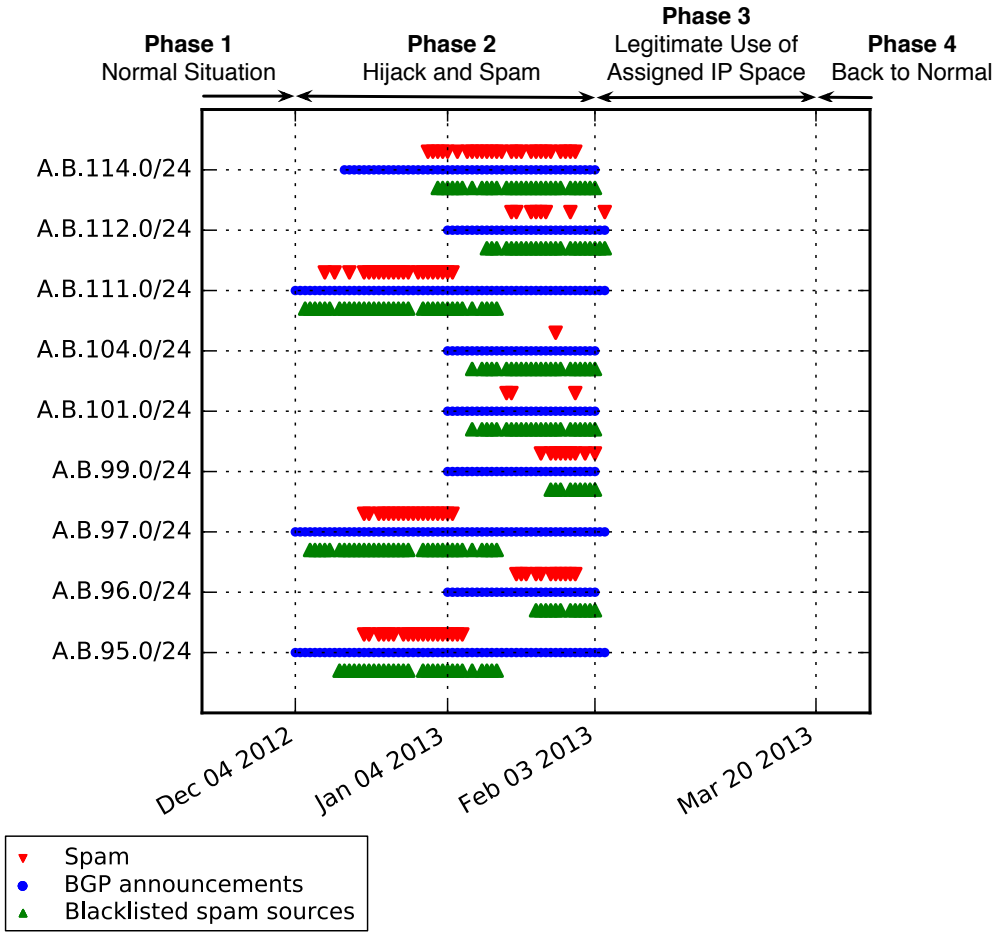


Figure A.16: Pourriel reçu et IP dans les listes noires des réseaux mentionnés

Les liens web inclus dans ces messages contenaient des noms de domaines qui se situaient, majoritairement, dans les préfixes incriminés. De cela, nous concluons que le pourriel était un outil pour promouvoir des sites d'arnaques hébergés sur cette infrastructure.

Les traces réseaux analysées entre décembre 2012 et mars 2013 montrent que la majorité du trafic provenant de ces réseaux étaient du SMTP, du DNS, et du HTTP. Le trafic était bi-directionnel, indiquant que des machines du MWZ échangeaient des données avec des machines des réseaux incriminés. La figure A.17 illustre ces traces.

Phase 3 : utilisation légitime de l'espace IP

Le 3 février 2013, Bob commence à annoncer cinq des neufs préfixes annoncés par Mallory, résultant en un MOAS qui dure quelques heures avant que Mallory ne disparaisse. Encore une fois, pendant cette phase, Alice continue à annoncer son /16 (figure A.15). La plupart des robots à pourriels

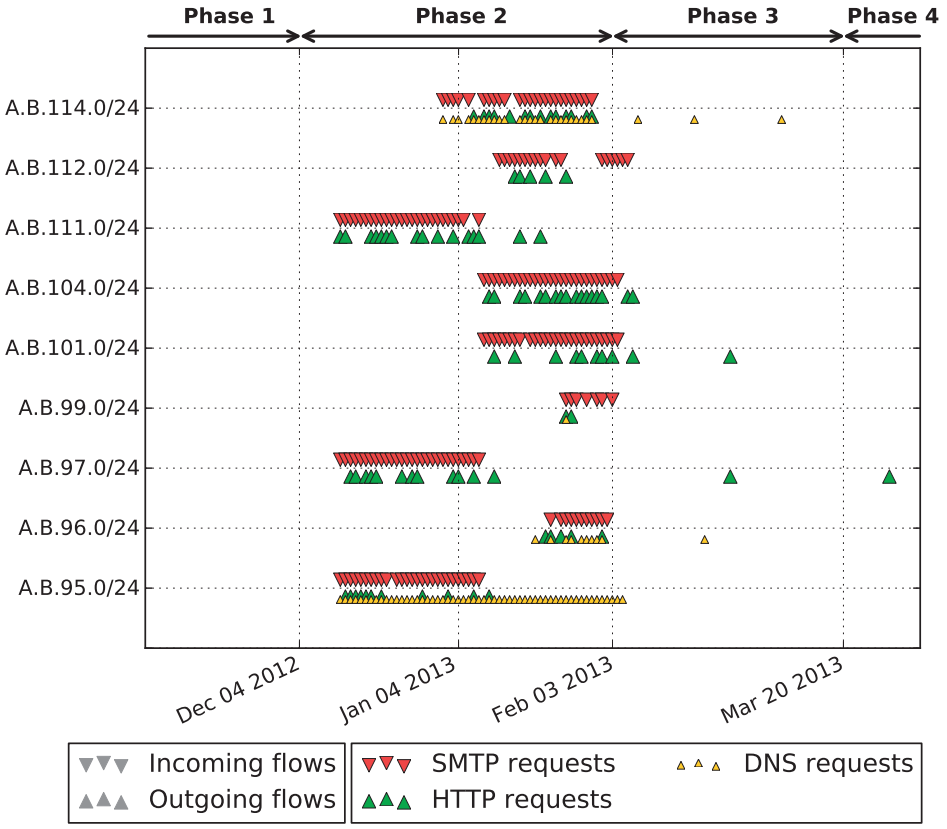


Figure A.17: Trace réseau des préfixes incriminés dans le cas bulgare

qui répondaient aux sondes de traceroutes pendant la phase 2 devinrent, eux aussi, subitement inatteignables, indiquant qu'il y a donc un réel changement topologique.

Selon son propre site web, Bob est un consultant IT se situant dans le même pays que Alice et Mallory. Son ASN est activement utilisé dans BGP depuis 2008, et tous les /24 d'Alice annoncés par Bob utilisent Alice comme FAI. La figure A.18 illustre la topologie BGP des différents acteurs.

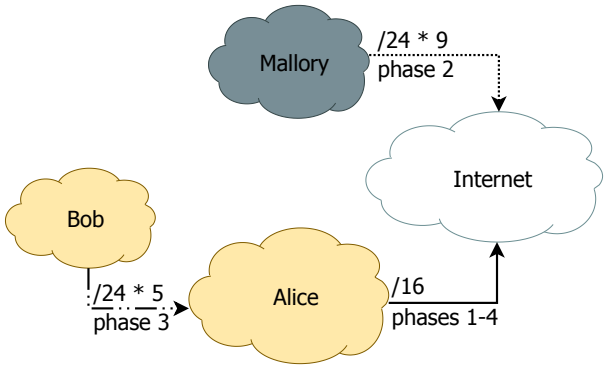


Figure A.18: Topologie dérivée de BGP

Au commencement de la phase 3, toute activité malveillante provenant de ces réseaux stoppe. Ceci semble indiquer que Bob a reçu les cinq préfixes de manière régulière.

Phase 4 : retour à la normale
Le 20 mars 2013, Bob arrête d'annoncer les cinq sous-préfixes d'Alice. La situation initiale, la phase 1, est donc de retour : seule Alice annonce A.B.0.0/16.

A.2.4.2 Seconde analyse

Sur base de cette analyse, les approches de [Ramachandran et al. 2006 ; Hu et al. 2007] auraient conclu une instance de détournement de préfixe avec intentions malveillantes. De fait, tous les indices présentés, ainsi que la corrélation forte existant entre le plan de contrôle BGP et le plan de données suggèrent qu'il s'agit bien d'une attaque délibérée contre Alice.

Malgré tous ces indices, nous avons décidé de continuer notre enquête, et nous avons mis en évidence une série d'éléments indiquant qu'il ne s'agit peut-être pas d'une attaque par détournement. En analysant plus d'un an d'archives IRR, nous avons constaté qu'Alice maintenait, de manière consciencieuse, ses objets de type `route`. La figure A.19 indique les origines associées à chaque préfixe. On y voit que, lors des phases 2 et 3, Alice a délégué les préfixes incriminés soit à Mallory (phase 2), soit à Bob (phase 3). Donc, suivant les données d'enregistrement de RIPE, et en supposant que ces données ne peuvent être modifiées que par Alice elle-même, Alice a bien autorisé ces deux autres entités à utiliser ces préfixes.

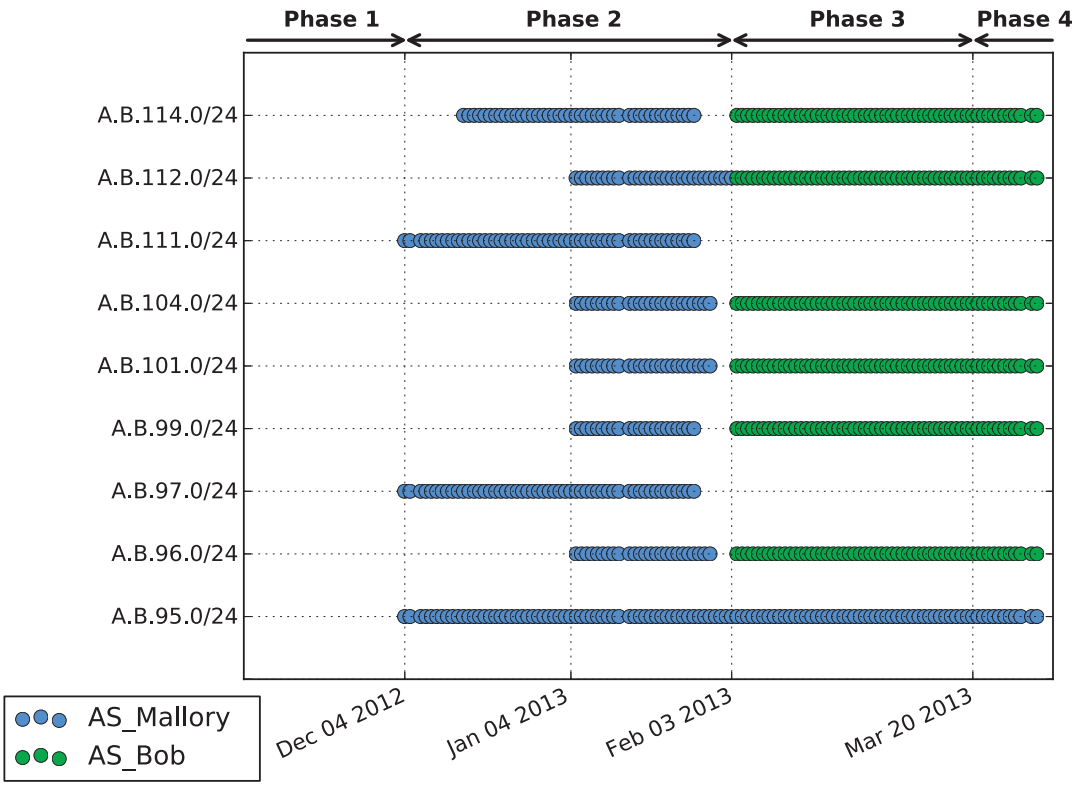


Figure A.19: RIPE IRR `route` objects for reported prefixes

Nous avons aussi contacté le réseau en amont de Mallory. Ce dernier nous a informé que Mallory avait effectivement demandé à annoncer un certain nombre de préfixes qu'il avait loué, mais le contrat avec Mallory a été cassé suite à de nombreuses plaintes.

En conclusion, nous pouvons dire que Mallory a effectivement envoyé du pourriel depuis les réseaux incriminés. Cependant, il est impossible de vérifier que Mallory a bien procédé à une attaque par détournement. Nos données montrent qu'il peut aussi bien s'agir d'un banal abus d'espace IP légitimement alloué.

A.2.5 Conclusion

Dans cette section, nous avons étudié les annonces simultanées, c.-à-d. les préfixes à origines multiples (**MOAS**). Premièrement, nous avons analysé et classifié ces préfixes en investiguant les raisons pour lesquelles ces préfixes sont annoncés. En nous basant sur cette classification, nous avons pu **réduire le champ de recherche d'attaques par détournement de l'ordre de 80%**. Deuxièmement, nous avons proposé un **système** nous **permettant d'analyser les cas suspects restants**. Ce système a été partiellement conçu sur les corollaires apportés par les travaux de [Ramachandran et al. 2006 ; Hu et al. 2007 ; Vervier et al. 2013 ; Schlamp et al. 2013]. Nous avons appliqué ce système sur un cas réel : le *cas bulgare*. Notre première analyse (section A.2.4.1), similaire à celles faites dans ces précédent travaux, nous indique que cet évènement réseau est effectivement le résultat d'une attaque par détournement intentionnelle. Cependant, notre seconde analyse (section A.2.4.2) utilise un ensemble de données qui nous permet d'élargir notre champ de vision, mais, en même temps, qui semble réfuter la thèse de l'attaque par détournement. Bien qu'il nous soit impossible de sélectionner une raison ou l'autre, nous avons montré qu'il est absolument nécessaire de considérer des données complémentaires afin de ne pas biaiser le résultat d'une analyse.

A.3 Le recouvrement de préfixes

Dans cette section, nous nous concentrons sur les recouvrements de préfixes, c.-à-d. sur des préfixes distincts qui annoncent les mêmes plages d'adresses IP. Les *sous-préfixes* sont la conséquence naturelle de la sous-allocation de préfixes. En effet, tout propriétaire d'espace IP peut choisir de diviser son espace IP en plusieurs morceaux de taille moindre. Etant donné que les paquets IP sont toujours acheminés vers le *préfixe le plus spécifique*, c.-à-d. vers le plus petit préfixe IP annoncé comprenant l'adresse de destination du paquet, une division de l'espace IP peut être le résultat d'ingénierie, par exemple pour assurer que des serveurs hors-site soient accessibles depuis le réseau global. En parallèle, l'attaque contre Spamhaus [Toonk 2013] a démontré que les sous-préfixes pouvaient être d'une redoutable efficacité afin d'opérer une attaque en déni de service. De plus, les conséquences peuvent être tout aussi désastreuses lorsqu'elles sont le résultat d'une mauvaise configuration d'un routeur [RIPE NCC 2008a].

Pour ces raisons, dans la première partie de cette section, nous décrivons et clarifions la façon dont ces préfixes sont utilisés. Pour ce faire, la méthode naïve consisterait à de comparer toutes les paires de préfixes qui se recouvrent. Considérons les trois préfixes a/8, a.b/16, et a.b.c/24. L'étude des trois paires de ces préfixes est-elle à propos ? Si le /8 appartient à un FAI, le /16 est probablement loué à un client du FAI, et c'est ce client qui a décidé de créer le /24. La comparaison entre le /8 et le /24 n'est donc pas pertinente. A l'inverse, si le /8 n'est pas un FAI, les deux autres préfixes ne sont pas le résultat d'une allocation, mais d'ingénierie. Dans ce cas-là, ils devraient être comparés tous les deux au /8, et pas entre eux. En d'autres termes, en comparant n'importe quelle paire de préfixes qui se recouvrent, on ignore la façon dont ces espaces IP sont assignés. Un bloc IP est assigné par un RIR (*Regional Internet Registry*) à une organisation. Cette organisation peut, à son tour, disposer de son espace IP de la manière qui lui plaît. En conséquence, la comparaison entre

deux préfixes quelconques se recouvrant ne tient pas compte du fait que différentes entités peuvent gérer les préfixes. Pour remédier à ce problème, nous utilisons les données des bases de données IRR (*Internet Routing Registry*) afin de grouper les différents préfixes annoncés dans BGP en **familles de préfixes**. Chaque préfixe inclus dans une famille tombe sous la tutelle du même administrateur, et la comparaison peut donc être faite sans ambiguïté.

La seconde partie de cette section est dédiée à l'étude des attaques par recouvrement de préfixe, et, plus précisément, le cas des sous-MOAS, c.-à-d. lorsque le préfixe plus spécifique est annoncé par un autre AS que l'AS d'origine du préfixe moins spécifique. Nous y proposons un système permettant de *valider* les annonces BGP sous-MOAS en combinant les informations récoltées dans trois sources de données distinctes. Premièrement, nous appliquons les recommandations que nous avons formulées suite à l'étude du *cas bulgare* (section A.2.4) en utilisant les IRR dans le but de trouver une possible relation entre deux préfixes se recouvrant, et, donc, d'éliminer les faux positifs. Ensuite, nous considérons la topologie au niveau AS des préfixes pour les comparer. Enfin, nous utilisons un balayage de l'espace IPv4 afin de constituer une base de données contenant les clés publiques SSL/TLS extraites d'une poignée de main avec un serveur HTTPS. Grâce à ce système, nous arrivons à valider la moitié des sous-MOAS dans l'espace IP couvert par le prototype.

A.3.1 Analyse du recouvrement de préfixes BGP

Dans cette section, nous montrons comment grouper les préfixes BGP en *familles de préfixes*, composées d'un *père* de famille, d'*enfants*, et de *sous-familles*.

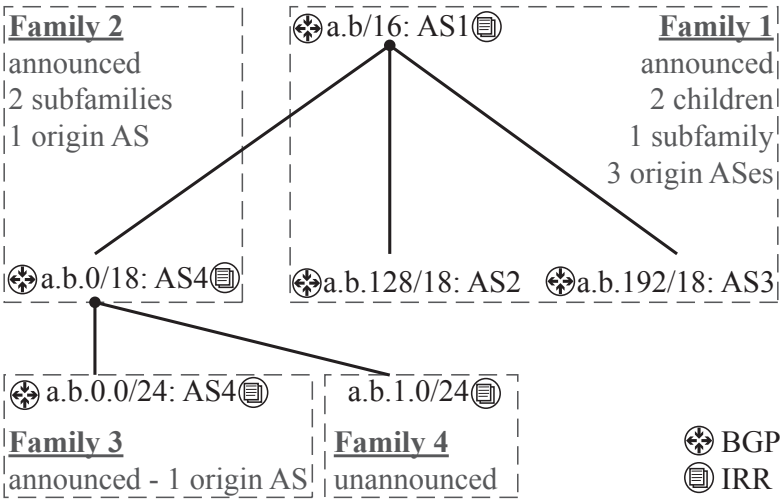


Figure A.20: Exemple de constitution de familles et de sous-familles

Tout préfixe inclus dans les bases IRR est *toujours* un **père de famille**. Il y a donc autant de familles que d'objets `inetnum` présents dans les IRR. Nous en comptons plus de $8,3 \cdot 10^6$. Puisque la plupart de ces préfixes se recouvrent, certains pères de familles incluent d'autres pères de familles. Cette situation mène aux sous-familles. Une **sous-famille** est une famille dont le père est complètement recouvert par l'espace IP d'un autre père de famille.

A titre d'illustration, la figure A.20 représente 4 familles, les préfixes a.b/16, a.b.0/18, a.b.0.0/24, et a.b.1.0/24 étant tous inclus dans les bases IRR. En conséquence, ces 4 préfixes sont donc des

pères de familles. Cependant, certains de ces préfixes se recouvrent. Ainsi, la famille 2 est une sous-famille de la famille 1 parce que le père de la famille 2 est plus spécifique que le père de la famille 1. De manière similaire, les familles 3 et 4 sont des sous-familles de la famille 2 ; mais ni la famille 3 et ni la famille 4 ne sont des sous-familles de la famille 1 parce que la famille 2 les "cache". Cette façon d'opérer s'explique par le fait que a.b.0/18 a été délégué à une autre entité que la famille 1, et ce, par la famille 1 (selon l'entrée IRR). La famille 2 est donc responsable de toute division de cet espace IP.

Une fois que les familles sont créées, elles sont peuplées avec les quelque 629k préfixes BGP. Un préfixe annoncé dans BGP est soit un père de famille, soit un préfixe plus spécifique qu'un père de famille. Dans le premier cas, il n'y a rien à faire : le père de famille est déjà inclus dans la famille qu'il définit. Dans le deuxième cas, le préfixe est ajouté à la famille en tant qu'enfant. Un **enfant** est un préfixe annoncé dans BGP, qui est plus spécifique que le père d'une famille, mais qui n'est pas déclaré dans les bases IRR comme ayant été délégué à une autre organisation. Par conséquent, le préfixe enfant est administré par la même organisation que le père de sa famille.

A.3.1.1 BGP et bases IRR

Dans cette section, nous comparons les préfixes annoncés dans BGP et les préfixes déclarés dans les bases IRR.

La première différence entre les deux sources de préfixes est leur nombre : il y a 12 fois plus de préfixes déclarés dans les bases IRR que de préfixes annoncés dans BGP. En comparant la distribution du nombre de préfixes suivant la longueur de leur masque, on constate que la distribution est fort similaire pour les /24 ou plus grands. Pour les préfixes plus petits, il y a au moins 100 fois plus d'entrées IRR que de préfixes annoncés dans BGP. Autrement dit, seulement 1% des préfixes IRR plus spécifiques qu'un /24 est annoncé dans BGP. Ce phénomène peut s'expliquer par deux raisons. D'une part, les bonnes pratiques BGP recommandent de ne pas propager les préfixes IPv4 plus petits que /24 [Hu et al. 2007]. D'autre part, les entrées IRR ne sont pas restreintes aux préfixes BGP : toute assignation de bloc IP peut être déclarée dans une base IRR. Ainsi, un serveur dédié peut avoir une entrée IRR pour son adresse IP (c.-à-d. un /32) liant, par exemple, cette adresse IP et une adresse postale de contact. Pour ces préfixes, il y a une différence entre le propriétaire du préfixe, et l'entité responsable de l'administration BGP nécessaire pour ce préfixe. Cette entité est, en général, le FAI du propriétaire, qui lui assure une connectivité Internet.

Regardons maintenant la taille relative des enfants et des sous-familles à l'intérieur d'une famille. La figure A.21 indique la distribution de la taille du masque des enfants selon la taille du masque du père de famille. Les abscisses indiquent la longueur du masque du père de famille ; les ordonnées indiquent la longueur du masque du préfixe enfant. La courbe est l'histogramme de la distribution : plus la ligne en une coordonnée est épaisse, plus il y a d'enfants de cette taille. Comme on peut le voir, indépendamment de la taille du père de famille, la majorité des enfants ont un masque de longueur 24. Ce résultat est contre-intuitif. On se serait, en effet, attendu à ce que des familles plus larges divisent leurs espaces IP en zones plus larges ; mais la pénurie d'adresses IPv4 peut expliquer que les FAIs préfèrent distribuer des préfixes de petite taille afin de maximiser leur utilisation.

La figure A.22 donne la taille du masque des pères des sous-familles d'une famille selon la taille du père de cette famille. Ici, on voit que la majorité de la distribution se situe autour des /29, indépendamment de la taille du père de famille. La popularité de ce type d'assignation IP peut s'expliquer par le fait qu'un préfixe /29, qui contient 6 adresses IP utilisables, est suffisant pour

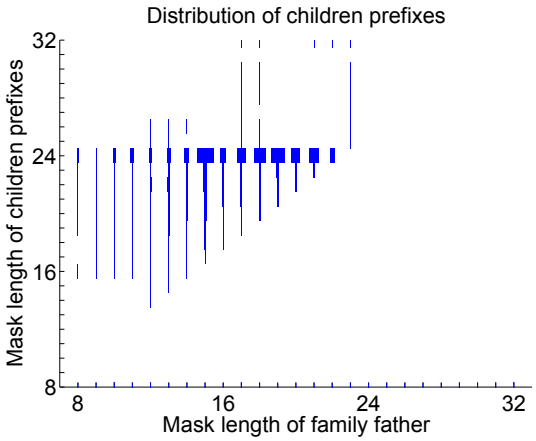


Figure A.21: Distribution de la taille du masque des enfants suivant la taille du masque du père de famille

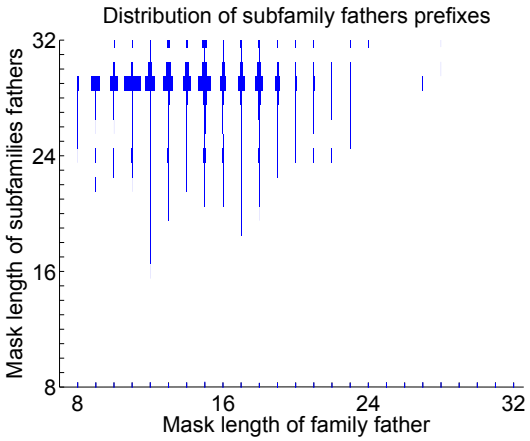


Figure A.22: Distribution de la taille du masque des pères des sous-familles suivant la taille du masque du père de famille

les petites et moyennes entreprises (PME) : quelques adresses IP publiques pour des serveurs et passerelles NAT. Les FAIs locaux (tier-3) étant généralement utilisés par ces PME pour accéder à Internet peut conduire à une prédominance naturelle de ces assignations.

A.3.1.2 Enfants et sous-familles

Dans cette section, nous nous concentrons sur le nombre d'enfants et de sous-familles d'une famille, et regardons l'espace du père occupé par ces préfixes.

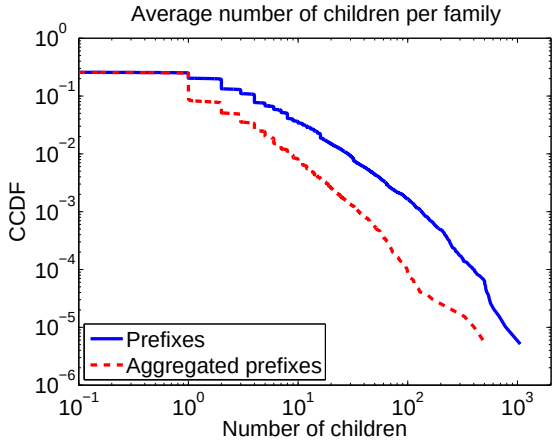


Figure A.23: Nombre d'enfants par famille

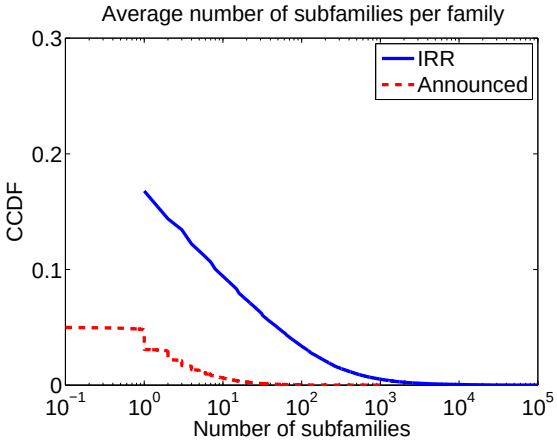


Figure A.24: Nombre de sous-familles par famille

La figure A.23 trace le nombre d'enfants par famille (trait plein) ainsi que le nombre *agrégé* d'enfants par famille (trait discontinu). Procéder à une agrégation sur les préfixes enfants nous permet de grouper ensemble les préfixes enfants qui occupent une plage IP voisine. Autrement dit, cette métrique nous indique le nombre d'espace IP continu dédié aux enfants, et pas le nombre d'enfants en lui-même. Seulement 25% des familles ont, en moyenne, au moins un enfant. De plus, la probabilité d'avoir un grand nombre d'enfants est fort basse. Pour 16% des familles, l'espace IP dédié aux enfants est contigu. Il y a donc une préférence nette pour assigner une partie de l'espace

IP de la famille aux enfants, ce qui évite une fragmentation importante de l'espace IP (et donc une configuration de routeur plus complexe).

La figure A.24 trace le nombre de sous-familles par famille. 83% des familles n'ont pas de sous-familles, ce qui s'explique par le fait qu'un très grand nombre de familles sont des /29, et donc difficiles à diviser.

Les figures A.23 et A.24 montrent que la majorité des familles n'ont ni enfant, ni sous-famille. En détails, 4% des familles ont des enfants et des sous-familles ; 22% des familles ont uniquement des enfants ; 1% des familles a uniquement des sous-familles ; et 73% des familles n'ont ni l'un ni l'autre. En d'autres termes, 73% des familles annoncent dans BGP uniquement le préfixe qui leur a été assigné, et ne l'ont pas sous-divisié.

Pour les 27% de familles qui ont soit un enfant, soit une sous-famille, soit les deux, la figure A.25 indique le ratio de recouvrement interne à la famille. Il s'agit du nombre d'adresses IP comprises dans les préfixes enfants ou dans les pères des sous-familles divisé par le nombre total d'adresses IP du père de famille. Pour 80% des familles, il n'y a pas de recouvrement avec les sous-familles, ce qui s'explique par le faible nombre de familles qui ont des sous-familles. Par opposition, dans 45% des cas, les enfants recouvrent totalement le père de famille. La figure A.25 indique aussi la somme du ratio de recouvrement par les enfants et du ratio de recouvrement par les sous-familles. Dans certains cas, ce ratio dépasse 1, ce qui signifie que, pour 3% des familles, les préfixes enfants et les préfixes des sous-familles recouvrent l'espace IP du père de famille plus d'une fois, et, donc, se recouvrent aussi eux-mêmes.

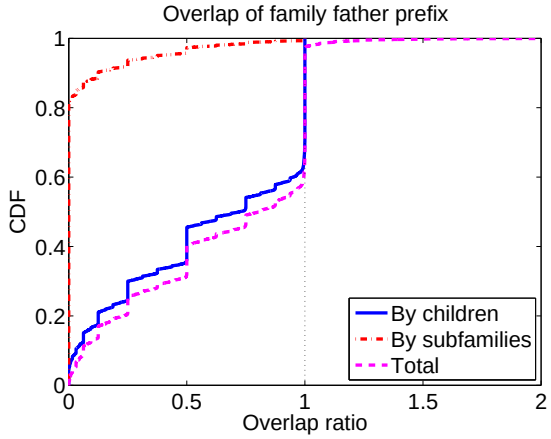


Figure A.25: Recouvrement au sein d'une famille

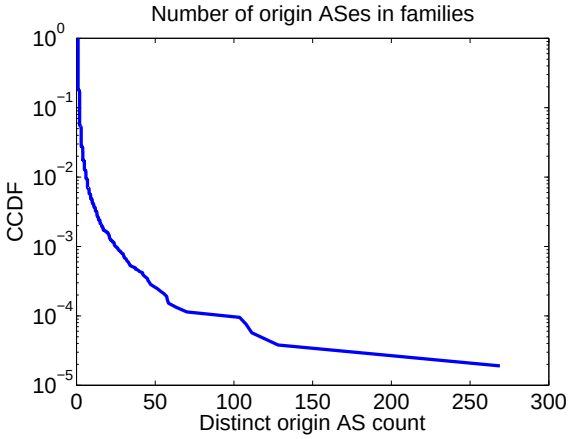


Figure A.26: Nombre d'AS d'origines par famille

A.3.1.3 Topologie au niveau AS

Cette section discute de la topologie niveau AS des familles de préfixes et de leurs enfants.

La figure A.26 trace le nombre d'AS d'origine dans une famille. Il s'agit du nombre d'AS distincts qui annoncent un préfixe inclus dans la famille. 81% des familles n'ont qu'un AS d'origine, 10% en ont deux, 3% en ont trois, et le reste en ont quatre ou plus. Cependant, il faut garder à l'esprit qu'un AS est une abstraction inhérente au protocole BGP, et pas forcément une infrastructure réelle localisée à un seul endroit. Par exemple, les préfixes de Cogent, un FAI dorsal (tier 1), sont annoncés par le seul AS174, mais sont physiquement situés à plusieurs endroits distincts.

Pour 42% des familles avec enfants, le père et au moins un enfant sont annoncés via BGP. Dans 87% des cas, ces préfixes ont le même AS d'origine ; dans 10% l'AS origine de l'enfant se situe en aval de l'AS d'origine du père ; et, dans 3% des cas, l'AS d'origine de l'enfant se situe en amont de l'AS d'origine du père.

A.3.2 Validations des annonces sous-MOAS

Jusqu'à présent, nous avons étudié le recouvrement de préfixes. Cette section s'attelle à valider les annonces de recouvrement, en particulier le cas où les préfixes se recouvrant sont annoncés depuis des AS d'origines distincts. Nous y présentons un système utilisant les relations existantes entre les divers objets inclus dans les bases IRR, un filtre basé sur les relations des topologies entre les divers AS d'origine, et une base de données de certificats SSL/TLS.

A.3.2.1 Architecture

L'architecture du système est représentée à la figure A.27. A partir d'un sous-MOAS (① dans la figure A.27), c.-à-d. de deux AS d'origines distincts, d'un préfixe moins spécifique et d'un préfixe plus spécifique, on cherche une relation administrative entre ces divers éléments (② dans la figure A.27) dans les bases IRR. Cette relation administrative existe si les divers éléments sont la propriété du même compte utilisateur de la base IRR. En même temps, on considère la topologie AS entre les origines (③ dans la figure A.27). Une relation fournisseur-client entre l'AS d'origine du sur-préfixe et l'AS d'origine du sous-préfixe implique que l'origine du sur-préfixe a décidé de ne pas filtrer l'annonce du sous-préfixe et qu'elle est donc légitime. Finalement, un balayage de l'espace IPv4 est utilisé pour collecter les clés publiques associées au service HTTPS. Lors de l'apparition d'un nouveau sous-MOAS, cette clé est utilisée afin de vérifier qu'un serveur joignable avant le sous-MOAS l'est toujours après, et que sa clé publique n'a pas changé. Si elle n'a pas changé, le sous-MOAS est estimé légitime.

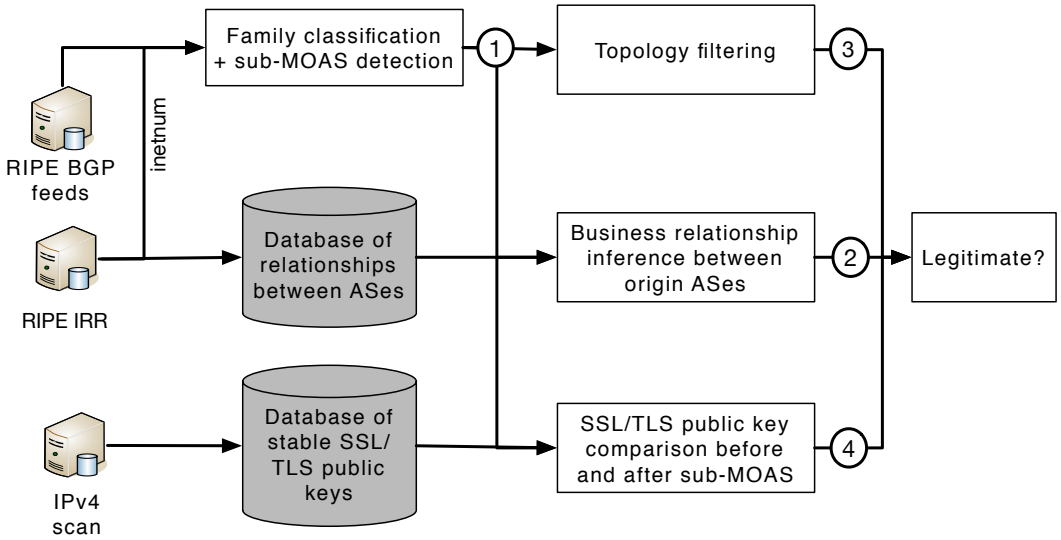


Figure A.27: Architecture du système de validation des annonces BGP sous-MOAS

A.3.2.2 Résultats

Nous avons utilisé notre système entre le 2 juin et le 12 juin 2014. Durant cette période nous avons pu valider 46.5% de tous les sous-MOAS grâce aux filtres représentés dans la figure A.27. Ce résultat est d'autant plus remarquable que, pour l'instant, seule la partie RIPE (c.-à-d. l'espace IP européen) des bases IRR a été implémentée dans le système. Cette partie des bases IRR ne couvre qu'approximativement 60% des cas de sous-MOAS. Notons que les autres filtres, c.-à-d. le filtre au niveau topologique et le filtre cryptographique, sont, eux, applicables à l'ensemble des cas. En particulier, le filtre cryptographique nous permet de valider 85% des sous-MOAS dans lesquels nous avons détecté un serveur HTTPS.

En d'autre termes, ces résultats sont encourageants et nous permettent de valider un nombre assez important de cas, bien que plus de 40% de l'espace IPv4 globalement alloué ne soit pas encore pris en compte dans notre implémentation. Par ailleurs, nous démontrons que les techniques actives de test sont très efficaces. De plus, notre base de données de clés publiques est disponible grâce à un balayage qui se révèle peu coûteux pour les systèmes distants puisqu'il n'est nécessaire que d'établir une connexion SSL/TLS, c.-à-d. que nous n'allons pas plus loin que la poignée de main.

A.3.3 Conclusion

Dans la première partie de cette section, nous avons défini une façon d'utiliser les données d'enregistrement des préfixes IP introduits dans les bases de données IRR comme source sémantique afin de grouper les préfixes annoncés dans BGP en plusieurs familles à l'intérieur desquelles une comparaison du recouvrement d'espace IP peut être faite sans ambiguïté. Nous avons montré que les bases IRR contiennent beaucoup plus de préfixes que les tables de routage BGP, particulièrement pour les préfixes dont le masque est plus long que 24. En particulier, nous avons vu que seul 2,32% des familles incluses dans les IRR sont visibles depuis BGP. Nous avons attribué cette observation au fait que les entrées IRR peuvent refléter n'importe quelle assignation IP, même celles faites en dehors BGP. De plus, un grand nombre d'entrées sont insérées et modifiées par les FAIs, ce qui est une preuve que l'information stockée dans les bases IRR ne peut pas être considérée comme globalement inutile. Nous avons montré que 74% des familles annoncées dans BGP n'ont pas d'enfant, ce qui est en accord avec les bonnes pratiques BGP. 15% des familles n'annoncent pas leur père de famille dans BGP, ce qui implique que ces familles présentent de l'espace dormant, et donc vulnérable à en croire [Vervier et al. 2015].

Dans la seconde partie de cette section, nous avons considéré les sous-MOAS et avons présenté un prototype capable de valider près de la moitié des sous-MOAS annoncés dans BGP. Pour cela, nous avons utilisé des règles combinant les informations administratives comprises dans les bases IRR, des informations liées à la topologie des AS d'origines, et des clés publiques associées au service HTTPS actif dans les réseaux considérés.

A.4 L'espace noir IP

Jusqu'à présent, la méthodologie que nous avons suivi consistait à isoler des évènements de routage suspects, et, ensuite, à chercher si une raison pouvait les expliquer, par exemple, en utilisant les IRR, où les assignations de préfixes sont détaillées, ou en utilisant des informations relatives au trafic

ou applications présentes sur ces réseaux, dans le but de trouver la réalité de terrain associée à l'évènement de routage. Dans cette section, nous allons inverser cette approche. Nous allons partir d'un cas de réalité de terrain suspect, et, ensuite, nous allons isoler les préfixes qui y sont associés.

Cette réalité de terrain est l'**espace noir IP**. Cet espace est composé des préfixes IPv4 *annoncés* dans BGP, mais qui n'ont *pas été alloués*, et qui n'ont *pas été réservés* pour une utilisation particulière (comme, par exemple, les zones d'espace IP privé). Puisqu'ils n'ont pas été alloués, ces préfixes ne devraient pas être annoncés. En même temps, cet espace noir est plus facile à détourner parce qu'il n'est surveillé ni par des opérateurs réseaux, ni par des algorithmes. Par exemple, [Thomas 2001] subit une attaque de DDos (Distributed Denial of Service (*déni de service distribué*)) attribuée à l'usurpation d'adresses IP (*IP spoofing*). Plus tard, [Feamster et al. 2004] suppose que cet espace est abusé mais n'en offre aucune preuve.

A.4.1 Calcul de l'espace noir

L'allocation de l'espace IP à une organisation se fait sur plusieurs niveaux. Le premier niveau est l'IANA (Internet Assigned Numbers Authority) qui distribue de larges zones IP aux RIRs (Regional Internet Registries). Pour l'espace IPv4, l'IANA distribue des préfixes dits "classe A", c.-à-d. des préfixes de masques de longueur 8. Ensuite les RIRs divisent cet espace et le distribuent aux LIRs (Local Internet Registries), suivant les besoins de ces derniers. La grandeur de l'espace alloué est alors variable. Les LIRs sont, en général, des FAIs ou des sociétés qui vont utiliser cet espace directement. S'il s'agit d'un FAI, l'espace alloué à un LIR peut être divisé et alloué aux clients de ce FAI.

En conséquence, l'espace IP non alloué se trouve, lui aussi, sur plusieurs niveaux. Il ne suffit pas de regarder quels préfixes ont été assignés par l'IANA, comme fait par [Feamster et al. 2004]. De par leur vocation, RIRs et LIRs disposent tous deux d'espaces IP qu'ils n'ont pas encore alloués à leurs clients. Donc, afin de trouver les préfixes de l'espace noir, il faut connaître quelles parties de chaque zone n'ont pas été allouées.

Pour cela, nous utilisons les IRRs (Internet Routing Registries), qui contiennent les informations d'allocation des préfixes : nous traitons chaque préfixe ayant une entrée `inetnum` comme alloué. Afin de compléter cette source d'information, nous utilisons les fichiers statistiques publiés chaque jour par tous les RIRs. Ces fichiers détaillent l'affectation faite des zones IP (et numéros d'AS) gérées par les RIRs et classifient les préfixes en quatre états. Les deux premiers sont les états ALLOCATED et ASSIGNED. Ils signifient que la zone IP spécifiée a été déléguée à un LIR et peut être utilisée. Les préfixes AVAILABLE n'ont pas encore été délégués, et donc *ne devraient pas* être annoncés. Finalement les préfixes RESERVED sont dans un état transitoire, et ne devraient plus (ou pas encore) être annoncés. Notons qu'il est utile de considérer les deux sources de données : les bases RIRs nous informent aussi des allocations au niveau des LIRs. De plus, il y a des inconsistances entre les deux sources [RADI], que nous éliminons afin de ne pas introduire d'erreurs dans notre méthode.

Une fois les zones IP non allouées connues, nous cherchons tout préfixe en faisant partie dans les tables de routage BGP.

A.4.2 Analyse de l'espace noir

Dans cette section, nous présentons l'analyse des préfixes constituant l'espace noir sur une période de sept mois, entre septembre 2014 et mars 2015.

A.4.2.1 Prévalence et persistance

La figure A.28 indique le nombre d'adresses IP incluses dans l'espace noir, pour chaque jour de la période d'étude. On y voit que l'espace noir varie entre l'équivalent de 10^{-2} et 10^{-1} équivalent d'un préfixe /8, en d'autres termes, la taille de l'espace noir est entre l'équivalent d'un /10 et d'un /15. Des maximums locaux sont observés en octobre 2014 et janvier 2015. Suivant les définitions de [Mahajan et al. 2002], nous avons attribué ces évènements à des fuites de routes : il s'agit d'une faible visibilité, des routes sortant d'un numéro d'AS privé, et atteignables via un chemin d'AS unique.

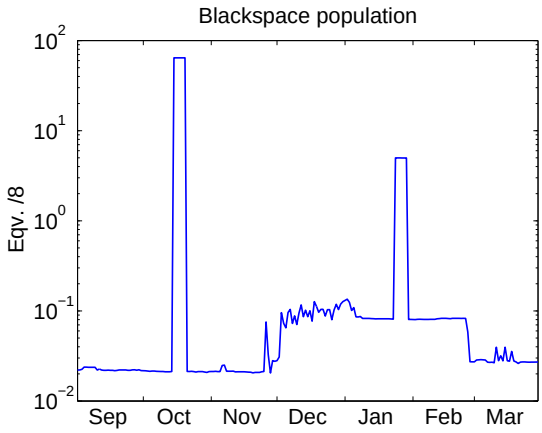


Figure A.28: Nombre d'adresses IP dans l'espace noir entre le 1er septembre 2014 et le 31 mars 2015

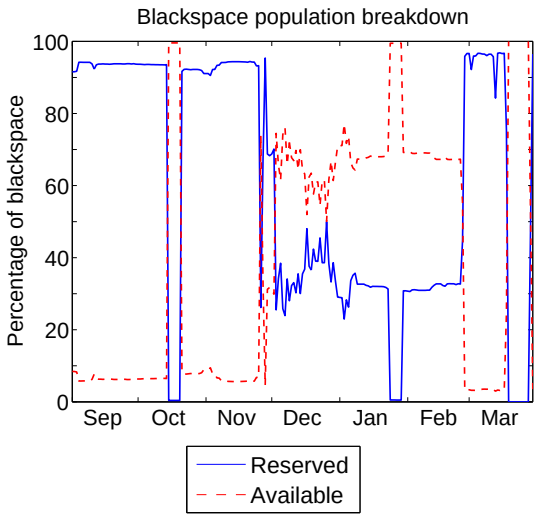


Figure A.29: Proportion quotidienne des adresses RESERVED et AVAILABLE dans l'espace noir entre le 1er septembre 2014 et le 31 mars 2015

La figure A.29 indique la proportion des adresses IP formant l'espace noir selon leur état dans les fichiers statistiques des RIRs. Fuites de routes exclues, la majorité des préfixes de l'espace noir sont des ressources RESERVED, c.-à-d. laissées de côtés par les RIRs parce qu'elles ne peuvent être allouées pour l'instant.

La figure A.30 indique le nombre de jours successifs où un préfixe est inclus dans l'espace noir. La ligne complète est la fonction de répartition de cette durée pour tous les préfixes, y compris les évènements transitoires observés lors des fuites de routes. La ligne pointillée est la fonction de répartition de cette durée, excluant ces évènements transitoires. Hors fuites de routes, la moitié des préfixes de l'espace noir y restent pendant un minimum de 12 jours ; seulement moins de 28% y sont pour moins d'un jour. La variation des préfixes constituant l'espace noir est tracée à la figure A.31. La courbe représente la valeur de l'index de Jaccard entre l'espace noir de deux jours successifs. En excluant les fuites de routes, on voit que la variation quotidienne est assez faible : bien en dessous de 10%.

A.4.2.2 Caractéristiques BGP

Lorsqu'un préfixe quitte l'espace noir, cela peut être soit parce qu'il n'est plus annoncé dans BGP, soit parce qu'une entrée dans un IRR indique son affectation, ou soit parce qu'un RIR a indiqué la zone IP comme allouée. La figure A.32 indique la proportion de chacune de ces conditions pour

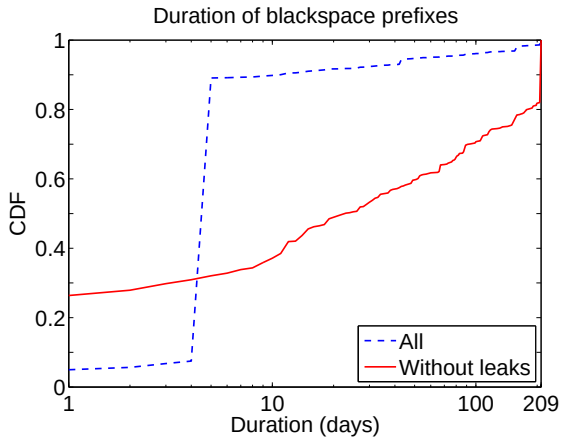


Figure A.30: Persistance des préfixes de l'espace noir

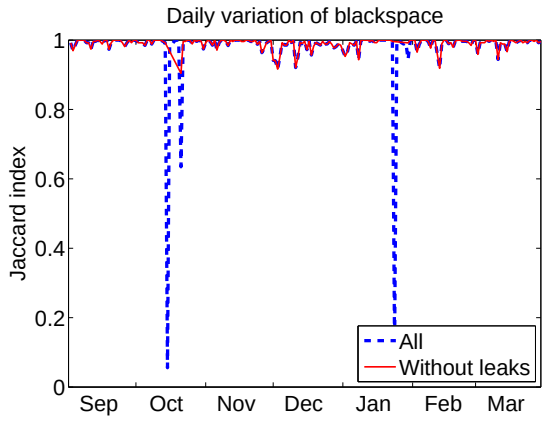


Figure A.31: Variation quotidienne des préfixes dans l'espace noir

les préfixes ayant quitté l'espace noir durant la période d'observation. Dans près de 45% des cas, le préfixe sort de l'espace noir parce que son annonce BGP s'est arrêtée. Dans le reste des cas, le préfixe semble avoir été alloué de manière régulière. En d'autres termes, il y a, en moyenne, quotidiennement l'équivalent d'un préfixe /11 annoncé dans BGP sans bonne raison ; et donc, une grande zone IP potentiellement malveillante.

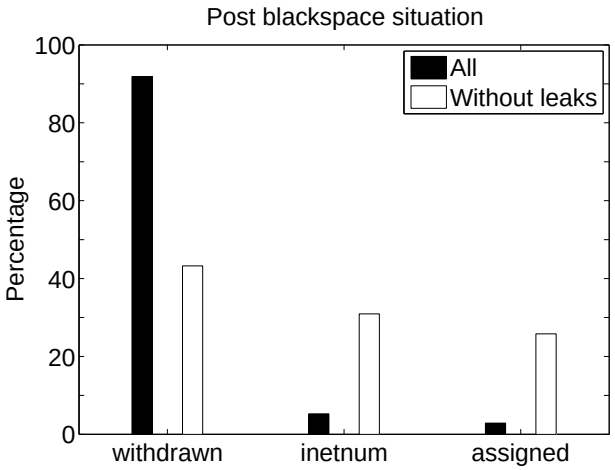


Figure A.32: Etat du préfixe lorsqu'il quitte l'espace noir

Comme tous les préfixes BGP, les préfixes de l'espace noir sont annoncés par un AS. Tout comme les préfixes, cet AS peut être, ou non, alloué. La figure A.33 indique le pourcentage quotidien de chaque état de l'AS d'origine des préfixes dans l'espace noir. Comme mentionné précédemment, les évènements liés aux fuites de routes sont annoncés par des AS privés. Les AS ALLOCATED, ASSIGNED, et RESERVED sont chacuns responsables d'approximativement 30% des annonces, et les AVAILABLE pour les 10% restants.

La figure A.34 indique la distribution du pourcentage de préfixes dans l'espace noir annoncé par un AS selon son état, pour les AS annonçant au moins un préfixe dans l'espace noir. On y voit que la métrique groupe les AS ALLOCATED et ASSIGNED ensemble d'un côté, et les AS AVAILABLE et

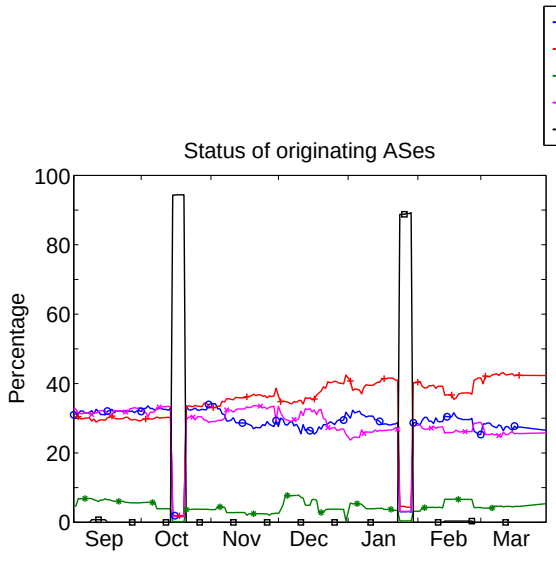


Figure A.33: Etat de l'AS d'origine des préfixes de l'espace noir

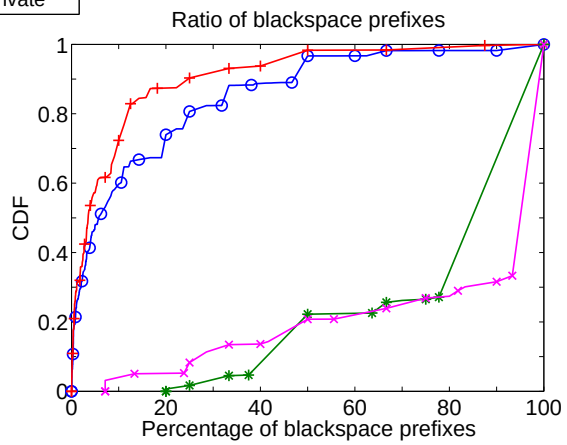


Figure A.34: Distribution du pourcentage relatif des préfixes dans l'espace noir suivant l'état de son AS d'origine

RESERVED de l'autre. En effet, en proportion, les AS ALLOCATED et ASSIGNED qui annoncent un ou des préfixe(s) de l'espace noir le font en faible quantité. Dans 90% des cas, moins de 1% des préfixes annoncés par ces AS sont dans l'espace noir. Le contraste est fort avec les AS RESERVED et AVAILABLE, où 70% de ces AS annoncent *uniquement* des préfixes de l'espace noir.

Si l'AS d'origine d'un préfixe dans l'espace noir est alloué, il est possible de regarder les informations d'enregistrement de cet AS. Ainsi, nous avons trouvé 185 noms pour les AS ALLOCATED et ASSIGNED qui annoncent des préfixes de l'espace noir, et avons cherché leurs informations de contact. Ces noms correspondent à un certain nombre de compagnies de tous domaines : FAIs, centres de données/fournisseurs de services dans le nuage, fabricants de matériel informatique, points d'échange, firmes de publicités, compagnies aériennes, banques, firmes d'ingénierie civile, transporteurs, commerces, hôpitaux, consultants militaires, ... Ces résultats pour les AS ALLOCATED et ASSIGNED semblent suggérer une utilisation non malveillante des préfixes de l'espace noir.

A.4.2.3 Plan de données et couche applicative

Comme nous l'avons vu à la section A.4.2.2 les AS à l'origine d'un ou plusieurs préfixe(s) de l'espace noir dans l'état ALLOCATED et ASSIGNED paraissent le faire sans intention malveillante. Mais, à la figure A.34, nous avons vu qu'un nombre important de préfixes de l'espace noir sont annoncés par des AS AVAILABLE et RESERVED. Afin d'éclairer la situation de ces préfixes, nous allons les analyser en détails au niveau du plan de données et au niveau des applications.

La première étape consiste à isoler les adresses IP joignables dans ces préfixes. A cette fin, nous utilisons [ZMap] pour envoyer des paquets à destination chaque adresse IP de ces réseaux, une fois par jour pendant février et mars 2015, depuis une machine située dans l'AS3215 (Orange). Nous envoyons des paquets TCP SYN sur les ports 21 (FTP), 22 (SSH), 25 (SMTP), 80 (HTTP), 137 (NetBios), 179 (BGP), et 443 (HTTPS), et regardons si nous recevons des paquets SYN/ACK. La figure A.35 indique le nombre de réponses reçues suite à ce balayage. On y voit que le nombre

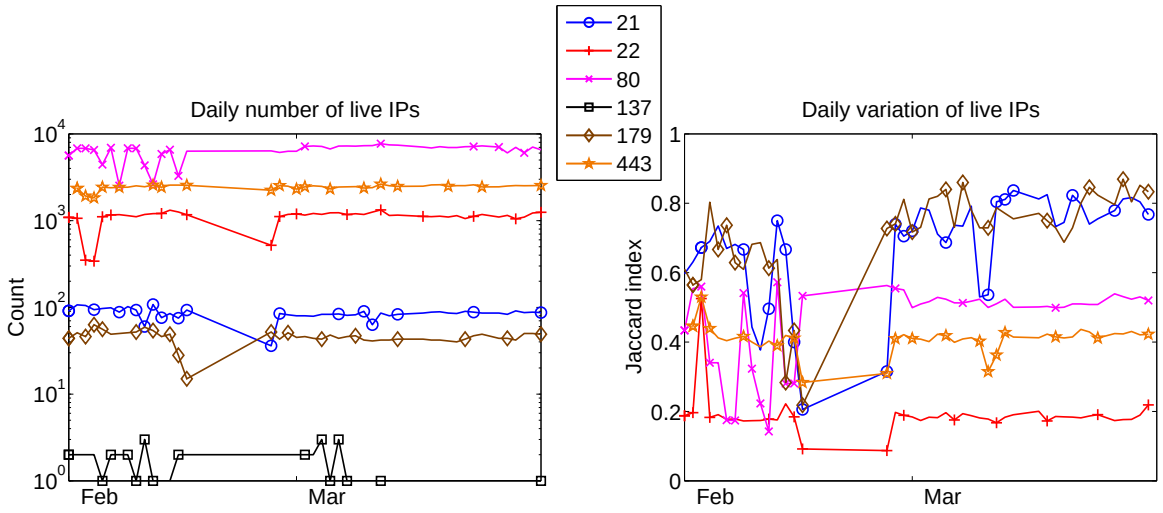


Figure A.35: Nombre quotidien de paquets SYN/ACK reçus de l'espace noir

Figure A.36: Variation quotidienne des adresses IP répondant au balayage de l'espace noir

de serveurs web est assez élevé : entre 6000 et 8000 machines tous les jours. Ensuite, les serveurs HTTPS, avec à peu près 2500 machines ; les serveurs SSH (1000 machines), et le reste.

La figure A.36 trace l'index de Jaccard de la différence quotidienne dans les adresses IP de l'espace noir répondant à notre balayage. On y voit que la variation est assez grande. Ceci est en contraste important avec la figure A.31 qui montrait que les réseaux constituant l'espace noir ne varient pas beaucoup. En conclusion, **les réseaux constituant l'espace noir sont activement configurés, et ne sont pas abandonnés**.

URLs, sites web, et noms de domaine

Nous venons de voir que près de 10 000 serveurs web sont accessibles dans l'espace noir. Afin de savoir quels genres de sites y sont hébergés, nous récupérons la page servie grâce à la requête `GET / HTTP/1.0`. En supplément, nous récupérons les pages associées aux erreurs HTTP 400 et 404 qui, parfois, retournent des informations supplémentaires associées au serveur.

De cette façon, nous collectons, chaque jour, des milliers de pages web disponibles dans l'espace noir. Etant donné leur nombre, nous ne pouvons pas les consulter une à une. En conséquent, nous utilisons un algorithme d'apprentissage non supervisé afin de grouper les pages web selon la similarité de leur code source brut. Nous obtenons quotidiennement entre 60 et 80 groupes distincts. Ces groupes contiennent des pages d'accès à divers services, comme les configurations de routeurs, applications d'e-mail, bureaux à distance ; des sites de petits commerces et entreprises ; des forums ; ...

Grâce à une base de données DNS passive, nous avons extrait 1428 FQDNs (*Fully Qualified Domain Names*, c.-à-d. des noms d'hôtes complets) correspondant à des adresses IP contenues dans l'espace noir. Ces 1428 hôtes sont distribués dans 556 domaines. Ensuite, nous avons composé une liste noire sur base de [PSBL] et de [VirusTotal] afin de savoir si des sites d'arnaque, de hameçonnage, ou de maliciel sont associés avec ces domaines. Notre liste blanche est composée des 10k domaines les plus populaires selon [Alexa]. Le résultat de la corrélation indique 5 domaines bénins, 516 non-classifiés, et 35 domaines (contenant 222 FQDNs) malveillants. Nous avons donc ici la preuve que l'espace noir contient de l'activité bénigne, mais aussi des criminels du cyber-espace.

Adresses IP malveillantes

Nous avons obtenu d'une société spécialisée dans la vente de solutions anti-virus une liste d'adresses IP, utilisée dans le monde réel, indiquant l'existence d'activités malveillantes. Ces adresses IP sont classifiées en quatre catégories : logiciels publicitaires (*adware*), hameçonnages, arnaques, et autres activités malveillantes. En faisant la corrélation avec les préfixes contenus dans l'espace noir, nous obtenons 4 réseaux associés aux arnaques, et 42 réseaux associés au hameçonnage.

Campagnes de spam

Nous avons aussi caractérisé le pourriel émanant de l'espace noir. En utilisant une liste noir composée de la SBL et la liste DROP de [Spamhaus], la [PSBL], la [WPBL], et la liste [Uceprotect], nous avons trouvé 206 404 adresses IP envoyant du pourriel depuis l'espace noir, dans 58 préfixe différents. La figure A.37 montre la forte corrélation existant entre les annonces BGP et l'activité de pourriel, pour une partie des préfixes corrélés.

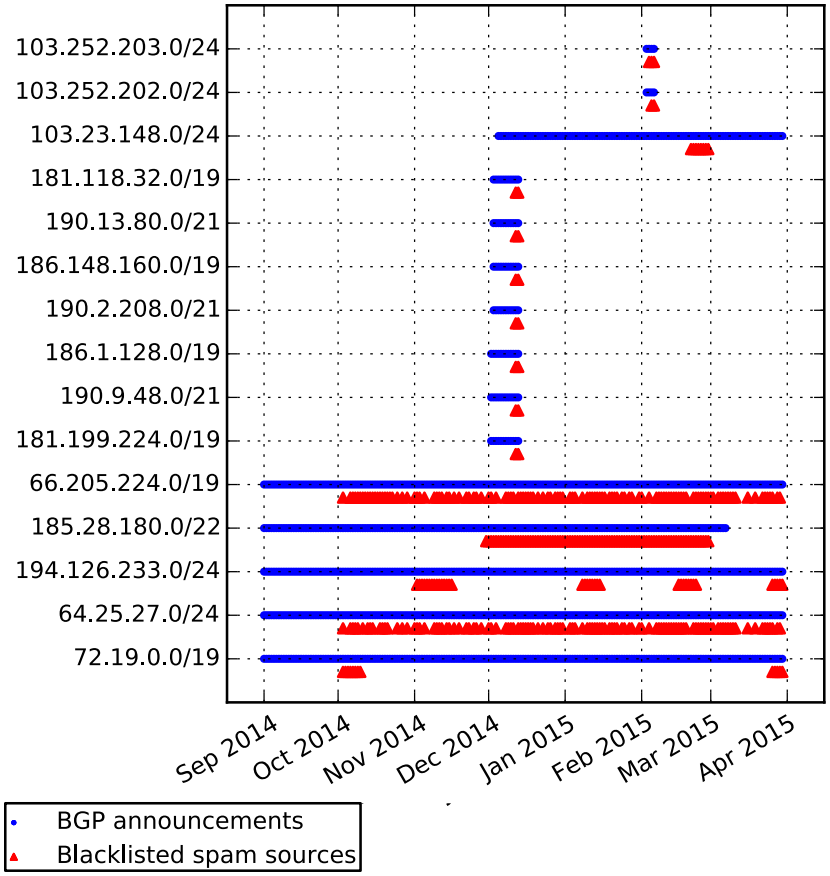


Figure A.37: Annonces BGP et sources de spam dans la liste noire pour le préfixe associé dans l'espace noir

Etude de cas

Lors de l'étude du pourriel émanant de l'espace noir, nous avons corrélé les préfixes de l'espace noir avec les préfixes détectés par SpamTracer [Vervier et al. 2015]. Cette corrélation nous a mené à 82 préfixes IP distincts, tous émanant d'un même AS d'origine : AS59790, enregistré en Allemagne. En y regardant de plus près, cet AS annonçait un total de 476 préfixes de l'espace noir entre le 17

octobre 2014 et le 8 janvier 2015. Tous ces préfixes ont été détournés de l'espace non alloué détenu par AfriNIC et ont, donc, été utilisés afin d'envoyer du pourriel.

A.4.3 Conclusion

Dans cette section, nous avons étudié l'espace IP noir, composé des préfixes IP annoncés dans BGP mais non alloués. Nous avons présenté une méthodologie permettant de calculer avec précision la composition de cet espace. Nous avons analysé les caractéristiques BGP des préfixes le composant, et avons montré que, quotidiennement, il a la taille de l'équivalent d'un préfixe /10. Bien que la moitié de cette zone IP semble être utilisée de manière bénigne, nous avons montré qu'une certaine quantité d'activité malveillante existe dans l'autre moitié. Notamment, nous avons montré que des préfixes de l'espace noir sont détournés pour héberger des sites de hameçonnage et d'arnaques en ligne, et envoyer du pourriel.

A.5 Conclusions et perspectives

Dans cette dissertation, nous avons étudié le phénomène du détournement de préfixes. Nous avons d'abord montré pourquoi les outils existants, même ceux de dernière génération, ne sont pas aptes à étudier ce phénomène : ils souffrent d'un taux de faux positifs trop élevé. De plus, ils sont principalement voués à être utilisés par les propriétaires des préfixes à surveiller. Ainsi, les faux positifs peuvent être facilement éliminés suite à la connaissance du comportement attendu du préfixe. Malheureusement pour nous, en tant qu'observateurs extérieurs, il n'est pas possible d'éliminer les faux positifs. Il faut donc analyser chaque situation, une à une, afin d'essayer de trouver la réalité de terrain. Afin d'analyser les situations, il est nécessaire de disposer d'un certain nombre de bases de données qui donnent des informations relatives aux réseaux observés. Les informations utiles comprennent des données sémantiques, telles que l'identité du ou des propriétaire(s) des réseaux, le comportement des réseaux au moment où l'alerte a été générée, et, si possible, les données nécessaires pour établir qu'un changement de comportement a eu lieu. Bien sûr, consulter ces informations nécessite beaucoup de temps. En conclusion, il faut, premièrement, localiser les bases de données utiles à la recherche de la réalité de terrain ; et, ensuite, réduire le nombre de cas à étudier afin de rendre la tâche possible.

Afin de trouver la réalité de terrain, nous avons utilisé les IRRs (Internet Routing Registries) qui contiennent, notamment, les informations d'enregistrement des préfixes IP et de numéros d'AS. De plus, grâce à nos partenaires de recherche, nous avons accès à des bases de données contenant des informations de sécurité relative au pourriels, arnaques, et maliciels ; et à des bases de données contenant des traces applicatives des réseaux.

Afin de réduire le nombre d'alertes à analyser, nous avons procédé à des analyses détaillées des pratiques BGP dans le but d'isoler un certain nombre de pratiques standards. L'analyse de ces pratiques, ainsi que leur impact possible au niveau du routage BGP nous permet d'écarter un nombre important de faux positifs.

Nous avons appliqué cette méthodologie à trois des méthodes possibles pour procéder au détournement de préfixe : les préfixes MOAS, les préfixes en recouvrement, et, en particulier, les sous-MOAS, et, enfin, l'espace noir IP, comprenant les préfixes publiquement annoncés sur BGP mais non alloués. Dans tous les cas, nous avons analysé les pratiques standards, ce qui nous permet d'écarter un nombre

important de faux positifs par rapport aux outils de l'état de l'art. Ensuite, nous avons proposé un système nous permettant une analyse concluante des cas restants. Nous avons montré la façon dont notre approche permet d'arriver à une conclusion finale avec une certitude plus grande qu'auparavant. Nous avons aussi mis au jour un grand nombre de cas où les attaques par détournement ont été utilisées comme la première étape d'une attaque plus complexe servant à héberger des sites web de hameçonnage, arnaque en ligne, ou hébergeant du maliciel, ou servant à envoyer du pourriciel.

Le travail apporté dans cette dissertation peut être étendu en considérant d'autres méthodes de détournement, en particulier, les attaques par homme au milieu. De plus, une exploration plus poussée de l'espace IPv6 est nécessaire ; mais, à l'heure actuelle, il n'existe pas de source de données de sécurité liées à l'IPv6.

Bibliography

[Address Management Hierarchy]
APNIC, *Understanding address management hierarchy*, http://www.apnic.net/services/manage-resources/address-management-objectives/management-hierarchy (accessed in August 2015).

[Address Space Registry]
Internet Assigned Numbers Authority (IANA), *IPv4 Address Space Registry*, http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml (accessed in August 2015).

[Alexa]
Alexa, *Actionable analytics for the web*, http://www.alexa.com/ (accessed in August 2015).

[Alperovitch 2010]
D. Alperovitch, *April route hijack: Sifting through the confusion*, http://blogs.mcafee.com/mcafee-labs/april-route-hijack-sifting-through-the-confusion-2 (accessed in August 2011), 2010.

[APNIC 2008]
APNIC, *AS number change could affect internet routing from 1 January 2009*, http://www.apnic.net/publications/news/2008/as-number-change-could-affect-internet-routing-from-1-january-2009 (accessed in August 2011), 2008.

[APNIC Whois Guide]
——, *Using Whois: Quick Beginners Guide*, http://www.apnic.net/apnic-info/whois_search/using-whois/guide (accessed in August 2015).

[Argus]
Tsinghua University, CNPT Lab, *Argus: Realtime prefix hijacking alarms*, http://argus.csnet1.cs.tsinghua.edu.cn/ (accessed in August 2015).

[ARIN 2008]
ARIN, *AS number change on 1 January 2009*, https://www.arin.net/announcements/2008/07242008.html (accessed in August 2011), 2008.

[ARIN 2012]
——, *Extended Allocation and Assignment Report for RIRs*, https://www.arin.net/knowledge/statistics/nro_extended_stats_format.pdf (accessed in February 2015), 2012.

[Ballani *et al.* 2007]
H. Ballani, P. Francis, and X. Zhang, "A study of prefix hijacking and interception in the internet", in *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, 2007, pp. 265–276.

[Bellovin *et al.* 2001]
S. Bellovin, R. Bush, T. G. Griffin, and J. Rexford, *Slowing routing table growth by filtering based on address allocation policies*, https://www.cs.princeton.edu/~jrex/papers/filter.pdf (accessed in November 2010), 2001.

[Bellovin 2012]
S. Bellovin, *Dear ripe: please don't encourage phishing*, http://mailman.nanog.org/pipermail/nanog/2012-February/045062.html (accessed in March 2013), 2012.

[BGPmon]
Colorado State University, *BGP monitoring system: Bgpmon*, http://bgpmon.netsec.colostate.edu/ (accessed in November 2010).

[BGPmon.net]
OpenDNS, *Bgpmon*, http://www.bgpmon.net/ (accessed in August 2015).

[Biersack *et al.* 2012]
E. Biersack, Q. Jacquemart, F. Fischer, J. Fuchs, O. Thonnard, G. Theodoridis, D. Tzovaras, and P.-A. Vervier, "Visual analytics for BGP monitoring and prefix hijacking identification", *IEEE Network Magazine, Special Issue on Computer Network Visualization, Volume 26 #6*, Nov. 2012.

[Blunk 2011]
L. Blunk, *Routing registry tutorial*, Talk at NANOG51, Jan 30 - Feb 2, 2011, Miami, FL, USA, https://www.nanog.org/meetings/nanog51/presentations/Sunday/NANOG51.Talk34.NANOG51%20IRR%20Tutorial.pdf (accessed in August 2015), 2011.

[Bogon Reference]
Team Cymru, *The Bogon Reference*, http://www.team-cymru.org/bogon-reference.html (accessed in January 2015).

[Bono 1997]
V. J. Bono, *7007 explanation and apology*, http://www.merit.edu/mail.archives/nanog/1997-04/msg00444.html (accessed in November 2010), 1997.

[Bright 2013]
P. Bright, *When spammers go to war: Behind the Spamhaus DDoS*, http://arstechnica.com/security/2013/03/when-spammers-go-to-war-behind-the-spamhaus-ddos/1/ (accessed in August 2015), 2013.

[Bruno *et al.* 2014]
L. Bruno, M. Graziano, D. Balzarotti, and A. Francillon, "Through the looking-glass, and what eve found there", in *8th USENIX Workshop on Offensive Technologies (WOOT 14)*, San Diego, CA, USA, 2014.

[Bu *et al.* 2004]
T. Bu, L. Gao, and D. Towsley, "On characterizing BGP routing table growth", *Comput. Netw.*, vol. 45, no. 1, pp. 45–54, May 2004.

[Bush *et al.* 2007]
R. Bush, J. Hiebert, O. Maennel, M. Roughan, and S. Uhlig, "Testing the reachability of (new) address space", in *Proceedings of the 2007 SIGCOMM Workshop on Internet Network Management*, ser. INM '07, 2007, pp. 236–241.

[Bush *et al.* 2009]
R. Bush, O. Maennel, M. Roughan, and S. Uhlig, "Internet optometry: assessing the broken glasses in internet reachability", in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC '09, 2009, pp. 242–253.

[Butler *et al.* 2010]
K. Butler, T. R. Farley, P. McDaniel, and J. Rexford, "A survey of bgp security issues and solutions", *Proceedings of the IEEE*, vol. 98, no. 1, pp. 100–122, 2010.

[CAIDA Ark]
Center for Applied Internet Data Analysis (CAIDA), *Archipelago (Ark) measurement interface*, http://www.caida.org/projects/ark/ (accessed in August 2015).

[IASR]
——, *Inferred AS Relationships*, http://as-rank.caida.org/data/ (accessed in November 2014).

[Chen *et al.* 2009]
K. Chen, D. R. Choffnes, R. Potharaju, Y. Chen, F. E. Bustamante, D. Pei, and Y. Zhao, "Where the sidewalk ends: Extending the internet AS graph using traceroutes from P2P users", in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09, 2009, pp. 217–228.

[Chin 2007]
K.-W. Chin, "On the characteristics of BGP multiple origin as conflicts", in *The Australiasian Telecommunications Networks and Applications Conference (ATNAC)*, 2007.

[Cisco 2014]
Cisco, "BGP – origin AS validation", *CISCO IOS XE RELEASE 3S – BGP Configuration Guide*, Dec. 2014.

[Cittadini *et al.* 2010]
L. Cittadini, W. Muhlbauer, S. Uhlig, R. Bush, P. Francois, and O. Maennel, "Evolution of internet address space deaggregation: Myths and reality", *IEEE J.Sel. A. Commun.*, vol. 28, no. 8, pp. 1238–1249, Oct. 2010.

[CNET News 1997]
CNET News, *Router glitch cuts net access*, http://news.cnet.com/2100-1033-279235.html (accessed in Novembre 2010), 1997.

[Cova *et al.* 2010]
M. Cova, C. Kruegel, and G. Vigna, "Detection and analysis of drive-by-download attacks and malicious javascript code", in *Proceedings of the World Wide Web Conference (WWW)*, 2010.

[Feamster *et al.* 2004]
N. Feamster, J. Jung, and H. Balakrishnan, "An empirical study of "bogon" route advertisements", *Computer Communication Review*, vol. 35, no. 1, pp. 63–70, 2004.

[Freedman 1997]
A. Freedman, *7007: From the horse's mouth*, http://merit.edu/mail.archives/nanog/1997-04/msg00380.html (accessed in November 2010), 1997.

[Gao 2001]
L. Gao, "On inferring autonomous system relationships in the internet", *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 733–745, 2001.

[Goodell *et al.* 2003]
G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. D. McDaniel, and A. D. Rubin, "Working around BGP: An incremental approach to improving security and accuracy in interdomain routing", in *Proceedings of the Network and Distributed System Security Symposium (NDSS 2003)*, 2003.

[Greenberg 2014]
A. Greenberg, *Hacker redirects traffic from 19 internet providers to steal bitcoins*, http://www.wired.com/2014/08/isp-bitcoin-theft/ (accessed in April 2015), 2014.

[Gregori *et al.* 2012]
E. Gregori, A. Improta, L. Lenzini, L. Rossi, and L. Sani, "On the incompleteness of the AS-level graph: A novel methodology for BGP route collector placement", in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12, 2012, pp. 253–264.

[Halse 2012]
G. A. Halse, *Whois.afrinic.net leaks passwords*, https://lists.afrinic.net/pipermail/rpd/2012/002586.html (accessed in September 2013), 2012.

[Hanford 2013]
S. Hanford, *Chronology of a DDoS: SpamHaus*, http://blogs.cisco.com/security/chronology-of-a-ddos-spamhaus (accessed in April 2015), Mar. 2013.

[Hong *et al.* 2009]
S.-C. Hong, H.-T. Ju, and J. W. Hong, "IP prefix hijacking detection using idle scan", in *APNOMS'09: Proceedings of the 12th Asia-Pacific network operations and management conference on Management enabling the future internet for changing business and new computing services*, 2009, pp. 395–404.

[Hu *et al.* 2007]
X. Hu and Z. M. Mao, "Accurate real-time identification of IP prefix hijacking", in *SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy*, 2007, pp. 3–17.

[Huston 2014]
G. Huston, *BGP in 2013 – The Churn Report*, http://labs.apnic.net/blabs/?p=457 (accessed in October 2014), 2014.

[Huston *et al.* 2011]
G. Huston, M. Rossi, and G. Armitage, "Securing BGP – a literature survey", *IEEE Communications Surveys and Tutorials*, vol. 13, no. 2, pp. 199 –222, 2011.

[Internet Alert Registry]
University of New Mexico, *The internet alert registry*, http://iar.cs.unm.edu/ (accessed in Novembre 2010).

[iPlane]
University of Washington, *Iplane*, http://iplane.cs.washington.edu/ (accessed in Novembre 2010).

[Juniper 2013]
Juniper, *Configuring origin validation for BGP*, https://www.juniper.net/techpubs/en_US/junos12.2/topics/topic-map/bgp-origin-as-validation.html (accessed in August 2015), May 2013.

[Karlin *et al.* 2006]
J. Karlin, S. Forrest, and J. Rexford, "Pretty good BGP: Improving BGP by cautiously adopting routes", in *ICNP*, 2006, pp. 290–299.

[Kent *et al.* 2000]
S. Kent, C. Lynn, J. Mikkelson, and K. Seo, "Secure border gateway protocol (S-BGP)", *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 103–116, 2000.

[Khan *et al.* 2013]
A. Khan, H. Kim, T. Kwon, and Y. Choi, "A comparative study on IP prefixes and their origin ases in BGP and the IRR", *Computer Communication Review*, pp. 16–24, 2013.

[Kisteleki 2010]
R. Kisteleki, *Filtering after recent Chinese "BGP hijack" does not affect RIPE region*, http://labs.ripe.net/Members/kistel/content-recent-chinese-bgp-hijack-does-not-affect-ripe (accessed in October 2010), 2010.

[Kurose *et al.* 2010]
J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 5th ed. Pearson, 2010.

[Labovitz 2010a]
C. Labovitz, *China hijacks 15% of internet traffic?*, http://asert.arbornetworks.com/2010/11/china-hijacks-15-of-internet-traffic/ (accessed in November 2010), 2010.

[Labovitz 2010b]
——, *Additional discussion of the April China BGP hijack incident*, http://asert.arbornetworks.com/2010/11/additional-discussion-of-the-april-china-bgp-hijack-incident/ (accessed in November 2010), 2010.

[Lad *et al.* 2006]
M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, "PHAS: A prefix hijack alert system", in *USENIX Security Symposium*, 2006.

[Litke *et al.* 2014]
P. Litke and J. Stewart, *BGP Hijacking for Cryptocurrency Profit*, http://www.secureworks.com/cyber-threat-intelligence/threats/bgp-hijacking-for-cryptocurrency-profit/ (accessed in October 2014), 2014.

[Madory 2015]
D. Madory, *The vast world of fraudulent routing*, http://research.dyn.com/2015/01/vast-world-of-fraudulent-routing/ (accessed in May 2015), 2015.

[Mahajan *et al.* 2002]
R. Mahajan, D. Wetherall, and T. Anderson, "Understanding BGP misconfiguration", *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 3–16, Aug. 2002.

[McArthur *et al.* 2009]
C. McArthur and M. Guirguis, "Stealthy IP prefix hijacking: Don't bite off more than you can chew", in *GLOBECOM*, 2009, pp. 1–6.

[McMillan 2010]
R. McMillan, *A Chinese ISP momentarily hijacks the internet*, http://www.nytimes.com/external/idg/2010/04/08/08idg-a-chinese-isp-momentarily-hijacks-the-internet-33717.html (accessed in November 2010), 2010.

[NSF1]
National Science Foundation, *From modest beginnings*, http://www.nsf.gov/about/history/nsf0050/internet/modest.htm (accessed in August 2011).

[NSF2]
——, *NSF and the birth of the Internet*, http://www.nsf.gov/news/special_reports/nsfnet/textonly/80s.jsp (accessed in August 2011).

[PCH]
Packet Clearing House (PCH), *Packet clearing house*, http://www.pch.net/home/index.php (accessed in August 2015).

[PCH LG]
——, *Looking glass*, http://www.pch.net/resources/data.php?dir=/routing-tables/looking-glass (accessed in August 2015).

[Pilosov *et al.* 2008]
A. Pilosov and T. Kapela, *Stealing the internet: An internet-scale man in the middle attack*, Presentation at DEFCON16, http://www.defcon.org/images/defcon-16/dc16-presentations/defcon-16-pilosov-kapela.pdf (accessed in October 2010), 2008.

[PNR]
Internet Assigned Numbers Authority (IANA), *Service name and transport protocol port number registry*, https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt (accessed in August 2015).

[Popescu *et al.* 2005]
A. C. Popescu, B. J. Premore, and T. Underwood, *The anatomy of a leak: AS9121*, Presentation at NANOG34, http://www.renesys.com/tech/presentations/pdf/renesys-nanog34.pdf (accessed in October 2010), 2005.

[Potaroo]
G. Huston, *BGP reports*, http://bgp.potaroo.net/ (accessed in August 2015).

[Potaroo: Autnums]
——, *AS names*, http://bgp.potaroo.net/cidr/autnums.html (accessed in April 2015).

[PSBL]
Passive Spam Block List, http://psbl.org/.

[Qiu *et al.* 2007]
J. Qiu, L. Gao, S. Ranjan, and A. Nucci, "Detecting bogus BGP route information: Going beyond prefix hijacking", in *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007.*, 2007, pp. 381–390.

[RADI]
G. Huston, *RIR Resource Allocation Data Inconsistencies*, http://www.cidr-report.org/bogons/rir-data.html.

[Ramachandran *et al.* 2006]
A. Ramachandran and N. Feamster, "Understanding the network-level behavior of spammers", in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '06, 2006, pp. 291–302.

[RFC675]
V. Cerf, Y. Dalal, and C. Sunshine, *Specification of internet transmission control program*, RFC 675, Dec. 1974.

[RFC904]
D. Mills, *Exterior gateway protocol formal specification*, RFC 904, Apr. 1984.

[RFC1105]
K. Lougheed and Y. Rekhter, *Border gateway protocol (BGP)*, RFC 1105, Jun. 1989.

[RFC1163]
——, *Border gateway protocol (BGP)*, RFC 1163, Jun. 1990.

[RFC1267]
——, *Border gateway protocol 3 (BGP-3)*, RFC 1267, Oct. 1991.

[RFC1654]
Y. Rekhter and T. Li, *A border gateway protocol 4 (BGP-4)*, RFC 1654, Jul. 1994.

[RFC1771]
——, *A border gateway protocol 4 (BGP-4)*, RFC 1771, Mar. 1995.

[RFC1930]
J. Hawkinson and T. Bates, *Guidelines for creation, selection, and registration of an autonomous system (AS)*, RFC 1930, Mar. 1996.

[RFC2385]
A. Heffernan, *Protection of BGP sessions via the TCP MD5 signature option*, RFC 2385, Aug. 1998.

[RFC2622]
C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra, *Routing policy specification language (RPSL)*, RFC 2622, Jun. 1999.

[RFC4012]
L. Blunk, J. Damas, F. Parent, and A. Robachevsky, *Routing policy specification language next generation (RPSLng)*, RFC 4012, Mar. 2005.

[RFC4271]
Y. Rekhter, T. Li, and S. Hares, *A border gateway protocol 4 (BGP-4)*, RFC 4271, Jan. 2006.

[RFC4632]
V. Fuller and T. Li, *Classless inter-domain routing (CIDR): the internet address assignment and aggregation plan*, RFC 4632, Aug. 2006.

[RFC5103]
B. Trammell and E. Boschi, *Bidirectional flow export using IP flow information export (IPFIX)*, RFC 5103, Jan. 2008.

[RFC5396]
G. Huston and G. Michaelson, *Textual representation of autonomous system (AS) numbers*, RFC 5396, Dec. 2008.

[RFC5925]
J. Touch, A. Mankin, and R. Bonica, *The TCP authentication option*, RFC 5925, Jun. 2010.

[RFC6480]
M. Lepinski and S. Kent, *An infrastructure to support secure internet routing*, RFC 6480, Feb. 2012.

[RFC6481]
G. Huston, R. Loomans, and G. Michaelson, *A profile for resource certificate repository structure*, RFC 6481, Feb. 2012.

[RFC6793]
Q. Vohra and E. Chen, *BGP support for four-octet autonomous system (AS) number space*, RFC 6793, Dec. 2012.

[RFC6996]
J. Mitchell, *Autonomous system (AS) reservation for private use*, RFC 6996, Jul. 2013.

[RIPE LIRs FAQ]
RIPE NCC, *FAQ: Becoming a member*, https://www.ripe.net/lir-services/member-support/info/faqs/faq-joining (accessed in August 2015).

[RIPE NCC 2008a]
——, *YouTube hijacking: A RIPE NCC RIS case study*, http://www.ripe.net/internet-coordination/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study (accessed in October 2010), 2008.

[RIPE NCC 2008b]
——, *AS number change could affect internet routing from 1 January 2009*, http://www.ripe.net/internet-coordination/news/announcements/as-number-change-routing-january-2009 (accessed in December 2010), 2008.

[RIPE NCC 2011]
——, *Authentication methods used in the RIPE database*, https://labs.ripe.net/Members/kranjbar/authentication-methods-used-in-the-ripe-database (accessed in August 2015), 2011.

[RIPEDB Docs]
——, *RIPE database documentation*, https://www.ripe.net/manage-ips-and-asns/db/support/documentation/ripe-database-documentation (accessed in December 2010).

[RIPE RIS]
——, *Routing Information Service*, http://www.ripe.net/ris/.

[Roughan *et al.* 2011]
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", *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 9, pp. 1810–1821, 2011.

[RouteViews]
University of Oregon, *RouteViews project*, http://www.routeviews.org/ (accessed in August 2015).

[RPKI Dashboard]
SURFnet / Jac Kloots, *RPKI dashboard*, http://rpki.surfnet.nl/ (accessed in August 2015).

[Schlamp *et al.* 2013]
J. Schlamp, G. Carle, and E. Biersack, "A forensic case study on AS hijacking: The attacker's perspective", *SIGCOMM CCR*, pp. 5–12, 2013.

[Schlamp 2015]
J. Schlamp, "An Evaluation of Architectural Threats to Internet Routing", PhD thesis, Technische Universität München, Nov. 2015.

[Shaw 2013]
A. Shaw, *Spam? not spam? tracking a hijacked Spamhaus IP*, https://greenhost.nl/2013/03/21/spam-not-spam-tracking-hijacked-spamhaus-ip/ (accessed in August 2015), Mar. 2013.

[Shi *et al.* 2012]
X. Shi, Y. Xiang, Z. Wang, X. Yin, and J. Wu, "Detecting prefix hijackings in the internet with Argus", in *12th ACM SIGCOMM Internet Measurement Conference, IMC '12*, 2012.

[SIDR]
Internet Egineering Task Force, *Secure Inter-Domain Routing*, https://datatracker.ietf.org/wg/sidr/ (accessed in August 2015).

[Siganos *et al.* 2002]
G. Siganos and M. Faloutsos, "BGP routing: A study at large time scale", in *Proc. IEEE Global Internet*, 2002.

[Siganos *et al.* 2004]
——, "Analyzing BGP policies: Methodology and tool", in *INFOCOM 2004*, vol. 3, 2004, pp. 1640–1651.

[Siganos *et al.* 2007]
——, "Neighborhood watch for Internet routing: Can we improve the robustness of internet routing today?", in *IEEE INFOCOM*, 2007.

[Snijders 2013]
J. Snijders, *CB3ROB/SPAMHOUSE hijack as seen from RIPE RIS route collector 3 (Amsterdam)*, http://instituut.net/~job/cb3rob-spamhaus-hijack-21-mar-2013.txt (accessed in August 2015), Mar. 2013.

[Spamhaus]
Spamhaus, *The Spamhaus project*, http://www.spamhaus.org/ (accessed in August 2015).

[Spirin 2010a]
D. Spirin, *Prefix hijacking by Michael Lindsay via Internap*, http://mailman.nanog.org/pipermail/nanog/2011-August/039379.html (accessed in September 2011), 2011.

[Spirin 2010b]
——, *Prefix hijacking by Michael Lindsay via Internap*, http://mailman.nanog.org/pipermail/nanog/2011-August/039381.html (accessed in September 2011), 2011.

[Spirin 2010c]
——, *Prefix hijacking by Michael Lindsay via Internap*, http://mailman.nanog.org/pipermail/nanog/2011-August/039568.html (accessed in September 2011), 2011.

[Tahara *et al.* 2008]
M. Tahara, N. Tateishi, T. Oimatsu, and S. Majima, "A method to detect prefix hijacking by using ping tests", in *APNOMS '08: Proceedings of the 11th Asia-Pacific Symposium on Network Operations and Management*, 2008, pp. 390–398.

[Tanenbaum 2002]
A. Tanenbaum, *Computer Networks*, 4th. Prentice Hall Professional Technical Reference, 2002.

[Thomas 2001]
R. Thomas, *60 Days of Basic Naughtiness: Probes and Attacks Endured by an Active Web Site*, http://www.team-cymru.org/documents/60Days.ppt (accessed in January 2015), 2001.

[Toonk 2008]
A. Toonk, *4 bytes autonomous system (AS) numbers*, http://www.toonk.nl/blog/?p=345 (accessed in August 2011), 2008.

[Toonk 2009]
——, *Monitoring your prefixes with bgpmon*, Presentation at NANOG45. Video: http://www.bgpmon.net/screencast.php; slides: http://nanog.org/meetings/nanog45/presentations/Sunday/Toonk_bgpmon_N45.pdf (accessed in March 2011), 2009.

[Toonk 2010]
——, *Chinese ISP hijacks the internet*, http://bgpmon.net/blog/?p=282 (accessed in December 2010), 2010.

[Toonk 2013]
——, *Looking at the Spamhaus DDOS from a BGP perspective*, http://www.bgpmon.net/looking-at-the-spamhouse-ddos-from-a-bgp-perspective/ (accessed in October 2014), 2013.

[Toonk 2015]
——, *Recent BGP routing incidents - malicious or not*, Presentation at NANOG 63, 2015.

[Uceprotect]
Uceprotect Network, *Germany's first spam protection database*, http://www.uceprotect.net/ (accessed in August 2015).

[Underwood 2006]
T. Underwood, *Con-ed steals the 'net*, http://www.renesys.com/blog/2006/01/coned_steals_the_net.shtml (accessed in November 2010), 2006.

[PHAS]
C. S. University, *A prefix hijack alert system*, http://phas.netsec.colostate.edu/ (accessed in Novembre 2010).

[van Beijnum 2002]
I. van Beijnum, *BGP*. O'Reilly Media, Inc., 2002.

[van Oorschot *et al.* 2007]
P. van Oorschot, T. Wan, and E. Kranakis, "On interdomain routing security and pretty secure BGP (psBGP)", *ACM Trans. Inf. Syst. Secur.*, vol. 10, no. 3, Jul. 2007.

[Vervier *et al.* 2013]
P.-A. Vervier and O. Thonnard, "SpamTracer: How stealthy are spammers?", in *5th IEEE International Traffic Monitoring and Analysis Workshop (TMA 2013)*, Apr. 2013.

[Vervier 2014]
P.-A. Vervier, "Detection, analysis and mitigation of malicious BGP hijack attacks", PhD thesis, Télécom Paris-Tech, Dec. 2014.

[Vervier *et al.* 2015]
P.-A. Vervier, O. Thonnard, and M. Dacier, "Mind your blocks: On the stealthiness of malicious BGP hijacks", in *NDSS 2015, Network and Distributed System Security Symposium, 8-11 February 2015, San Diego, California, USA*, San Diego, CA, USA, Feb. 2015.

[VirusTotal]
VirusTotal, *Free online virus, malware and URL scanner*, https://www.virustotal.com/ (accessed in August 2015).

[Wählisch *et al.* 2012]
M. Wählisch, O. Maennel, and T. C. Schmidt, "Towards detecting BGP route hijacking using the RPKI", *ACM SIGCOMM CCR*, vol. 42, no. 4, pp. 103–104, 2012.

[Wepawet]
Wepawet, *Wepawet*, http://wepawet.cs.ucsb.edu (accessed in May 2015).

[White 2003]
R. White, "Securing BGP through secure origin BGP", *Internet Protocol Journal*, vol. 6, no. 3, 2003.

[Wolf 2010]
J. Wolf, *Pentagon says "aware" of China Internet rerouting*, http://www.reuters.com/article/idUSTRE6AI4HJ20101119?pageNumber=1 (accessed in November 2010), 2010.

[WPBL]
Weighted Private Block List, http://www.wpbl.info/.

[Xiang *et al.* 2011]
Y. Xiang, Z. Wang, X. Yin, and J. Wu, "Argus: An accurate and agile system to detecting IP prefix hijacking", in *19th IEEE International Conference on Network Protocols (ICNP 2011)*, 2011, pp. 43–48.

[Yan *et al.* 2009]
H. Yan, R. Olivera, K. Burnett, D. Matthews, L. Zhang, and D. Massey, *Bgpmon: A real-time, scalable, extensible monitoring system*, CATCH2009, http://bgpmon.netsec.colostate.edu/download/publications/catch09.pdf (accessed in October 2010), 2009.

[Zhang *et al.* 2008]
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 2008 conference on Data communication*, ser. SIGCOMM '08, 2008, pp. 327–338.

[Zhao *et al.* 2001]
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 1st ACM SIGCOMM Workshop on Internet Measurement*, ser. IMW '01, 2001, pp. 31–35.

[Zheng *et al.* 2007]
C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis, "A light-weight distributed scheme for detecting IP prefix hijacks in real-time", *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 277–288, 2007.

[Zimmermann 1995]
P. R. Zimmermann, *The Official PGP User's Guide*. 1995.

[ZMap]
Z. Durumeric, E. Wustrow, and J. A. Halderman, "Zmap: Fast Internet-wide scanning and its security applications", in *Proceedings of the 22nd USENIX Security Symposium*, Aug. 2013.

[Zmijewski 2010]
E. Zmijewski, *Accidentally importing censorship*, http://www.renesys.com/blog/2010/03/fouling-the-global-nest.shtml (accessed in December 2010), 2010.