Detection of BGP Hijacking Using TTL Analysis

Tamir Carmeli

Detection of BGP Hijacking Using TTL Analysis

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science

Tamir Carmeli

Submitted to the Senate
of the Technion — Israel Institute of Technology
Shevat 5778 Haifa February 2018

This research was carried out under the supervision of Prof. Reuven Cohen, in the Faculty of Computer Science.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Prof. Reuven Cohen, for his excellent guidance. Reuven's guidance was patient, kind and inspiring. I feel privileged for having had the opportunity to work with him and to learn from him.
I would also like to thank my family and friends, for encouraging me, and for caring.

The Generous Financial Help Of The Technion Is Gratefully Acknowledged.

Contents

List of Figures

List of Tables

Abstract

The Border Gateway Protocol (BGP) plays an important role in the Internet infrastructure. However, it was developed in the 1980s with limited concern for security. In particular, it lacks authentication, which makes it vulnerable to the so-called prefix hijacking attack. In this attack, a malicious or compromised BGP router announces a route to an IP prefix it does not own. Consequently, packets destined to this prefix are actually forwarded to the attacker. A special case of this attack, known as interception attack, is when the attacker manages to forward the hijacked traffic to the intended destination. Interception attacks have been publicly documented since 2013, when a Belarusian ISP successfully intercepted traffic whose original route should have never left North America. In this thesis we study the effect of prefix interception on the TTL (Time To Live) value of hijacked IP packets as observed by their real destinations, with the aim of detecting whether a sudden TTL increase can be attributed to prefix interception or to a legitimate link failure. We first analyze how interception attacks and link failures change the TTL from the perspective of the packet receiver, and then study additional TTL-related effects of the prefix interception attack. Using these observations, we propose a detection method for the attack and use simulations to evaluate its performance.

Abbreviations and Notations

AS	:	Autonomous System
ASN	:	Autonomous System Number
BGP	:	Border Gateway Protocol
CAIDA	:	Center for Applied Internet Data Analysis
IDS	:	Intrusion Detection System
IP	:	Internet Protocol
ISP	:	Internet Service Provider
IXP	:	Internet Exchange Point
MOAS	:	Multiple Origin AS
NA	:	Not Applicable
OSPF	:	Open Shortest Path First
POP	:	Points of Presence
RIP	:	Routing Information Protocol
RIPE	:	Reseaux IP Europeens
SSFNet	:	Scalable Simulation Framework Network
STD	:	Standard Deviation
TTL	:	Time To Live
Tier-1	:	A network that can reach any network on the Internet without purchasing IP transit.

Chapter 1

Introduction

The Border Gateway Protocol (BGP) is the interdomain routing protocol used by Autonomous Systems (ASes) in the Internet to advertise IP prefix ownership, and it lacks the option to authenticate the ownership of prefixes advertised by routers to their peers. This makes the BGP vulnerable to the IP prefix hijacking attack. In this attack, an AS pretends to own an IP prefix that it actually does not own, and its BGP router announces a route to this prefix [45]. Such announcements, also known as bogus BGP messages, influence other BGP routers to route the traffic destined to the hijacked prefix to the attacker BGP router. The attacker can perform malicious activities on the hijacked packets, and may then choose to forward these packets back to the legitimate owner of the prefix (the victim). This may prevent the victim from detecting that these packets have been compromised.

A simple example of such an attack is demonstrated in Figure 1.1. In this figure, arrows represent links between ASes (inter-AS links), thick arrows represent links chosen by the BGP to route traffic to AS-100, and thin arrows represent links on non-preferred routes. The direction of an arrow indicates AS relationships; e.g., AS-400 is a provider of AS-300. In this example we assume that routes are selected by local preference policy. In Figure 1.1(a), traffic is initially routed from AS-300 to AS-100. Suppose that AS-300 equally prefers to route via AS-400 or via AS-200, and that AS-400 is the ISP of AS-300. Thus, symmetry is broken by the shortest AS-path, and the chosen route is [AS-300, AS-400, AS-500, AS-100].

Now, suppose that AS-200 initiates an attack to hijack the traffic destined for AS-100. Figure 1.1(b) describes the routing after AS-200 sends AS-300 a BGP announcement reporting that $\left[\text{AS-200 AS-100}\right]$. AS-300 prefers the new route since it is one hop shorter than the previously chosen one [AS-400, AS-500, AS-100], and it forwards to AS-200 rather than to AS-400 the traffic destined for AS-100. This attack is likely to increase the hop count of packets sent from AS-300 to AS-100, since such packets are now forwarded along a route that consists of 4 rather than 3 ASes.

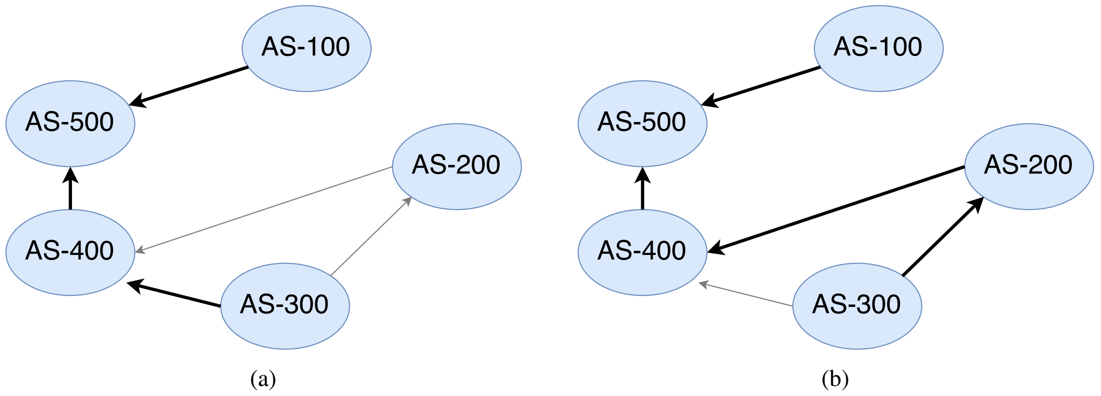

Figure 1.1: Traffic interception.

There are many reports of prefix hijacking in the Internet. In 2008, for example, a Pakistani AS advertised a more specific prefix than the one owned and advertised by YouTube, in order to prevent Pakistani users from accessing the site [5]. Since the BGP prefers announcements to more specific prefixes, even if they have a longer route, this advertisement indeed prevented access to the YouTube servers. In fact, access to these servers was prevented not only for Pakistani users but for two-thirds of all Internet users. Another incident occurred in 2013, when a Belarusian ISP hijacked American IP prefixes, causing traffic to be diverted from Mexico to Belarus [8]. This traffic was intercepted by the attacking ISP, and then forwarded back to the US. A similar attack was conducted by an Icelandic ISP, which intercepted traffic from North America and returned it to the legitimate destination in the US.

A BGP hijacking attack over Bitcoin may also used to disconnect nodes or slow down block propagation so Bitcoin users and merchants lose money [15]. A real world example occurred in 2014, when a BGP hijacking attack intercepted traffic from Bitcoin miners to a mining pool server. This attack was used to steal $83,000 in cryptocurrency [12]. In 2015, confidential documents of a company called “Hacking Team” were leaked, revealing their involvement in BGP hijacking [2].

Prefix hijacking has two different flavors:

- Blackholing: The hijacked traffic reaches the hijacking AS, and never continues to its true destination. This attack is useful mainly for denial of service. It is also useful for an AS that wants to take control over IP addresses that belong to another AS.
- Interception (man in the middle): The hijacked traffic reaches the hijacking AS and is then forwarded to the true destination using a route that was not polluted by the attack. The hijacking AS can intercept or modify the traffic, without the knowledge of the legitimate sender and receiver. Although most Internet traffic today is encrypted [6], some is not, and even interception of encrypted traffic can be harmful because it may reveal traffic metadata [3], [26].

In this thesis we develop a scheme by which an AS can determine whether the traffic it receives was intercepted by a prefix hijack. We first intended to identify prefix interception by analyzing the TTL field in the received packets to detect any sudden increase in the number of hops they have traversed along their route. However, we discovered that it is very difficult to distinguish whether this increase was due to prefix interception or to a legitimate route change after a failure. We therefore analyzed other traffic properties, looking for the one that can be uniquely associated with prefix interception rather than with topological changes. With respect to a specific destination AS, we found that the number of source ASes whose traffic is intercepted by a prefix interception attack is significantly higher than the number of source ASes whose traffic is affected by a link failure. We then use this property to develop an algorithm that can accurately distinguish between prefix interception and link failure.

The rest of this thesis is organized as follows. Section 2 reviews related work. Section 3 compares the effect of prefix interception and link failure on the route length. Section 4 studies additional factors that affect the route length, proposes a method to distinguish between prefix interception and link failure based on the study, and analyzes the results of the method. Section 5 details how the simulations used to study prefix interception and link failure were conducted. Section 6 concludes the work.

Chapter 2

Related Work

Improving BGP security is a widely researched area, and previous works employ several different approaches. In [20] and [40], machine learning and data mining algorithms are used for detecting various anomalies in BGP logs collected from publicly available BGP record collectors. Some of these anomalies are prefix hijacks, BGP table leaks, and link failures. However, the use of public BGP collectors has several limitations. First, the collectors are only periodically updated. For example, RouteViews [11] is updated only every 15 minutes. Thus, systems that use collectors are usually not able to detect network anomalies in real time. Second, collectors collect BGP logs from a small set of BGP routers, and therefore have only a partial view of routing events.

In [18], several public BGP collectors are used to detect prefix hijacking for prefixes owned by a given AS. After a hijacking is detected, de-aggregated sub-prefixes of the hijacked prefix are automatically sent. For example, if 21.3.2.0/23 has been hijacked, 21.3.2.0/24 and 21.3.3.0/24 are automatically announced. The proposed solution is evaluated using PEERING [9], a system that makes real BGP announcements possible using its own ASNs and prefixes.

SBGP [31], BGPSec [29], psBGP [41] and soBGP [42] use cryptography to improve BGP security and prevent route manipulation attacks. These extensions cryptographically sign and validate the AS path in BGP messages. Since they all require changing the implementation of existing routers, they cannot be fully deployed in the Internet. In [34], the authors indicate that partial deployment of these extensions will actually degrade BGP security. One reason is that partial deployment creates a tradeoff between connectivity and security, which might be exploited to cause ASes to favor short and insecure routes over long and secure ones. In [19], the authors suggest a cryptographic solution that improves BGP security even under partial deployment. This work extends RPKI [7] to authenticate only the last AS link in an AS path, so that short and partially secure routes can be selected.

In [28], [38] and [45], active probing is used to detect prefix hijacking. These works use probing programs, such as Ping or Traceroute, to send packets to IP addresses suspected as hijacked. Then, anomalies in the response messages, which might result from prefix hijacking, are detected.

Examples of such anomalies are missing hosts or hosts that change their response to the probing programs. However, since prefix hijacking leading to interception causes traffic to be forwarded to the legitimate prefix owner, responses to the described probing methods will show no anomalies. Hence, this method can be used to detect prefix blackholing only, where probing packets are routed to a different network. Another drawback of active probing is that it adds extra traffic to the network.

In [46], another active technique is proposed and implemented: monitors are placed at vantage points in the Internet. These monitors constantly probe the location of network prefixes, searching for inconsistencies. If a significant change were to be detected, an alert would be issued to the legitimate prefix owner.

The work described in [33] combines an active technique and control plane analysis to detect prefix interception. It runs traceroute measurements from vantage points to a set of IP prefixes and monitors which ASes use to route the packets. Once it discovers than an AS on the route has become a "hotspot" (meaning that many more traceroutes route through it than in the normal state) it is suspected as the interception attacker and is searched in control plane records, e.g., RouteViews [11], for anomalies.

In [32], prefix hijacking caused by Multiple Origin AS (MOAS) scenarios is detected. This attack is a special case of prefix hijacking, where a BGP router sends a BGP announcement regarding a prefix whose origin is different from the one reported in a previous BGP update for the same prefix. Consequently, there are at least two announced routes for the same prefix, each indicating a different destination AS.

In [44], prefix hijacking attack that exploits BGP AS-PATH prepending is discussed. In such an attack, routers deliberately add their own AS number multiple times to a route they announce. This makes the route less attractive, and it usually becomes useful only as backup for the shorter main route. Malicious BGP routers exploit existing prepended routes by shortening and re-announcing them, making them as attractive as the main routes, and enabling the malicious router to intercept their traffic.

Both [32] and [44] detect a specific pattern of prefix interception by matching a signature of the pattern against recent BGP announcements. A limitation of these methods is that a match for a specific form of prefix hijacking is usually not a good match for other forms. Specifically, the method described in [32] detects routes leading to a “wrong” destination, but does not detect prepended routes that were shortened by malicious routers, since the destination AS of such routes is correct. In a similar manner, the method described in [44] detects a match for shortened prepended routes, but does not detect non-prepended routes that lead to a “wrong” destination AS.

In [23], the authors analyze topological properties of ASes in order to determine how a prefix hijacking attack is propagated to a large number of ASes. The authors use simulations to determine the likelihood of an AS to become a victim or an attacker. This work reports that the ability to successfully conduct a prefix hijack is mainly a function of distance: victims located far from Tier-1 ASes and attackers located close to Tier-1 ASes are more likely to be part of a successful attack.

In [16], the authors analyze RouteViews records to estimate the probability of prefixes belonging to Tier-1 ASes to be hijacked. They then define a simple signature of a hijacking scenario. The proposed signature is compared to the outcome of probing packets, such as those obtained using Traceroute. The authors state that the signature does not deal well with prefixes whose origin AS uses traffic engineering. Another drawback of this method is that it performs IP-to-AS mappings based on BGP routing tables, a technique known to be error prone [35].

Chapter 3

Comparing the Effect of Prefix Interception and a Link Failure on the Route Length

Two previous studies of prefix interception [8], [14] show that such an event is likely to increase the hop count of the intercepted packets by up to 100%, since instead of being routed on their original route, these packets are first routed on a polluted route to the attacker, and then on a non-polluted route to the correct destination. However, network anomalies other than prefix interception, such as a failure of an inter-AS link or a change in BGP routing policy, might also increase the hop count of packets. In [39] it is found that a change in the routing policy of an AS is usually due to swapping the local preference between two neighboring ASes. The effect of such a change on the length of the route to the destination is similar to the effect of an inter-AS link failure, which forces an AS to route using another neighbor AS. Since a link failure does not cause packets to be routed "twice", but rather to be routed directly while bypassing the failed link, we expect that the hop count increase due to link failure would be less significant than the hop count increase due to prefix interception. As shown later, this hypothesis is not always correct.

Suppose that an AS sends a bogus announcement with respect to a prefix p of a victim AS, AS$_D$. We can divide the ASes in the Internet to those that adopt this announcement and those that do not. For every AS$_i$ in the first set, all traffic generated by AS$_i$ and destined to p is subject to misrouting and hop count increase. Hence, we propose to use a TTL-based scheme to detect significant increases in the hop count and assume such events can be attributed to a prefix interception attack. This scheme samples received packets, computes the number of hops each packet has traversed according to the packet's TTL field, e.g., using the method described in [30], and maintains a separate counter for measuring the average number of hops for every [originating AS, destination subnet] combination. The originating AS is an AS in the Internet that sends traffic to AS$_D$. The destination subnet is a local subnet of AS$_D$ to which packets from the originating AS are destined.

For example, consider Figure 3.1 and suppose that the prefix 11.11.22.0/24 of AS-100, referred to as Prefix2, is intercepted by AS-200 in the same way described in Figure 1.1. Suppose that AS-100 has another prefix 11.11.11.0/24, referred to as Prefix1, which is not intercepted. AS-100 maintains a hop counter for each of the following combinations: [AS-200, Prefix1], [AS-200, Prefix2], [AS-300, Prefix1], [AS-300, Prefix2] and so on. After Prefix2 is intercepted by AS-200, the counter of [AS-300, Prefix2] is expected to show a significant increase since the traffic originated by AS-300 to 11.11.22.0/24 is routed twice: first towards AS-200 and then over a non-polluted route to AS-100.

The above technique does not require AS-100 to monitor all the packets it receives. Since all the packets from a given source node traverse the same route, it would be enough to monitor only a sampling of the packets, e.g., only 0.1%.

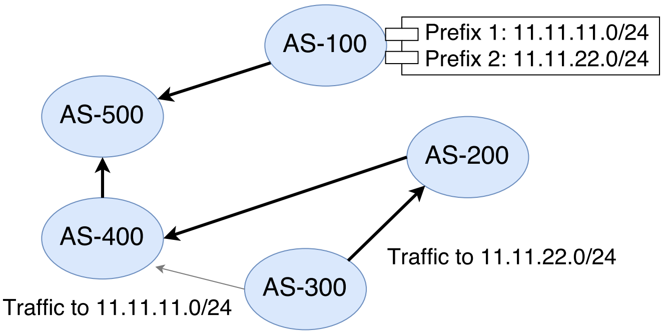

Figure 3.1: Interception of only one prefix of AS-100.

However, as already noted, the hop count may increase also due to inter-AS link failures. Thus, the main problem is to distinguish between hop count increase due to a link failure and hop count increase due to prefix interception. Since real data of packets recorded during prefix interception events is not publicly available, we study this problem using the popular SSFNet simulator [13]. Network topologies from CAIDA's datasources [4] are used as input to the simulator. SSFNet has a feature that allows an attacker AS to broadcast bogus BGP announcements regarding a given AS victim. However, this feature does not necessarily produce a non-polluted route towards the victim when a polluted route is injected into the network, because each bogus BGP announcement is sent to all the BGP neighbors of the attacker AS. This prevents the routing path from the attacker to the correct destination (the victim) from remaining non-polluted, which is a necessary condition for traffic interception. We therefore changed SSFNet such that it would be possible to send a bogus announcement only to a portion of the neighbors. This allows us to partition the network into polluted and non-polluted ASes with respect to a certain bogus announcement, thereby making prefix interception possible. Further details regarding the simulation setup are provided in Section 5.

Table 3.1 shows simulation results of traffic interception vs. link failures for three different destination ASes. Consider first Table 3.1(a), for which the destination AS (the victim) is AS-

39026. The first row in the table shows the average hop count from each source AS as measured during normal operation. The second row shows the average hop count measured during prefix interception attacks on a destination prefix of AS-39026. The third row shows the average hop counts measured during inter-AS link failure scenarios. In these scenarios, links on the AS paths from each source AS to AS-39026 are taken down. For example, we see in Table 3.1(a) that the hop count of packets sent from AS-3267 to AS-39026 increases, on the average, by 90.71%, due to prefix interception. In contrast, the average hop count increase of the same traffic flows due to a link failure is significantly smaller: 14.28%.

	Destination AS: AS-39026				
Source ASes	AS-20640	AS-6849	AS-3267	AS-1267	AS-8928
Normal hop count	8	9	7	10	8
Average hop count due to prefix interception	12.53	14.92	13.35	12.91	12.76
Average hop count due to link failure	10.67	11	8	9.5	8
Hop count increase due to prefix interception	56.63%	65.78%	90.71%	29.1%	59.5%
Hop count increase due to link failure	33.38%	22.22%	14.28%	-5%	0%

(a)

	Destination AS: AS-41176				
Source ASes	AS-36352	AS-8220	AS-21592	AS-3292	AS-48159
Normal hop count	11	10	18	9	12
Average hop count due to prefix interception	16.45	14.47	19.6	13.69	16.72
Average hop count due to link failure	10	12.67	16.33	11.67	16
Hop count increase due to prefix interception	49.55%	44.7%	8.89%	52.11%	38.5%
Hop count increase due to link failure	-9.09%	26.7%	-9.28%	29.67	33.33%

(b)

	Destination AS: AS-12925				
Source ASes	AS-8419	AS-8447	AS-8928	AS-12989	AS-43554
Normal hop count	13	8	11	10	12
Average hop count due to prefix interception	15.32	13.09	14.07	14.97	15.9
Average hop count due to link failure	13.4	9.67	14.25	11.23	12.75
Hop count increase due to prefix interception	17.85%	63.63%	27.91%	49.7%	32.5%
Hop count increase due to link failure	3.08%	20.88%	29.55%	12.3	6.25

(c)

Table 3.1: Comparison of hop count increase rates during simulated prefix interception and link failure scenarios.

There are, however, exceptions to this rule. For example, in Table 3.1(c) we see that the average hop count increase of packets sent from AS-8928 to AS-12925 after prefix interception is similar to the increase after a link failure. Cases such as these necessitate the use of additional factors indicative of prefix hijacking to help us distinguish more precisely between the two root causes.

Chapter 4

Searching for Other Factors that Affect Hop Count Increase

In [23], the authors mention factors that affect the magnitude of prefix hijacking, i.e., the number of ASes that become polluted. One such factor, which we refer to as $\mathbf{F_1}$, is the depth of the victim AS in the Internet graph. The depth is defined as the length of the AS path from the victim AS to its closest Tier-1 AS. It is found in [23] that the magnitude of prefix hijacking is larger when the victim ASes are located deeper in the AS graph. Another related factor studied in [23] is the difference between the depth of the attacker AS and the depth of the victim AS. Specifically, the magnitude of an attack is found to be larger when the above difference is larger. We call this factor $\mathbf{F_2}$.

Consider a source and a destination located in AS_S and AS AS_D, respectively. In addition to F_1 and F_2, we focus on two other factors we believe relevant for distinguishing between the two root causes of hop count increase:

- $\mathbf{F_3}$: The AS path length from AS_S to AS_D. We expect to see a larger increase in the hop count of packets that are routed from AS_S to AS_S when the AS path length, i.e., number of intermediate ASes, is shorter. To see the reason, consider Figure 4.1. In this figure, AS_{S1} and AS_{S2} route to AS_D over AS paths whose lengths are x_1 and x_2, respectively. Suppose that AS_M carries out a prefix interception attack on the prefixes of AS_D. Due to this attack, both source ASes have their packets routed first to AS_M and then to AS_D rather than directly to AS_D. Let the number of AS hops between AS_M and AS_D be z. For S_1, the AS path length increases due to prefix interception by $y_1 + z - x_1$, whereas for S_2 it increases by $y_2 + z - x_2$. If AS_M is an arbitrary AS, then y_1, y_2 and z are equal to the average number of hops in the Internet, denoted $\overline{d}$. Thus, the expected AS path increases for S_1 and S_2 are $2\overline{d} - x_1$ and $2\overline{d} - x_2$, respectively. If $x_1 < x_2$, then $2\overline{d} - x_1 > 2\overline{d} - x_2$.

- $\mathbf{F_4}$: Similarly to F_3, we expect to see a larger increase in the number of hops when the original length, in number of hops, is shorter. Thus, we define as F_4 the number of hops from the

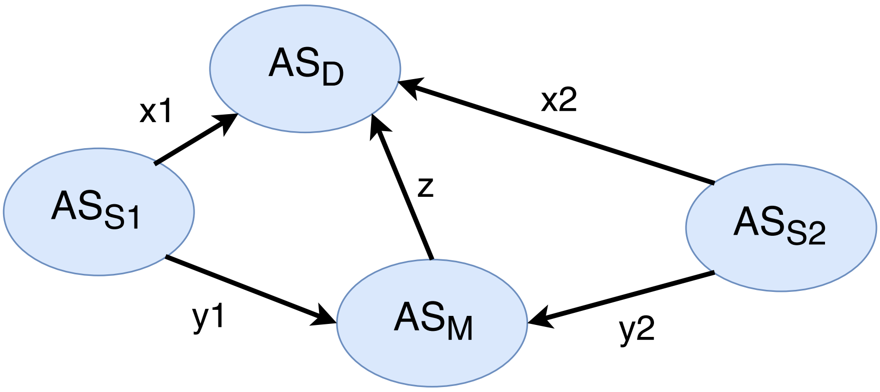

Figure 4.1: Hop count increases differently between source ASes during prefix interception.

source node to the destination node.

We first analyze the effect of F_1-F_4 on the TTL by computing the Pearson Correlation Coefficient between the increase in hop count during simulated prefix interception and link fail scenarios and the values of these factors as observed by several destination ASes. Consider a destination AS_D for which we conduct n simulated scenarios of prefix interception or of link failures. For each factor F_k, $k = 1 \dots 4$, the Pearson Coefficient ρ between F_k and the hop count increase during prefix interception, denoted $\delta(HC)$, is defined as:

$$\rho(F_k, \delta(HC)) = \frac{Cov(F_k, \delta(HC))}{Var(F_k)Var(\delta(HC))} = \frac{\sum_{i=1}^{n}(F_{ki} - \overline{F_k}) \cdot \delta(HC_i))}{\sqrt{\sum_{i=1}^{n}(F_{ki} - \overline{X})^2} \cdot \sqrt{\sum_{i=1}^{n} \delta(HC_i)^2}}, \tag{4.1}$$

where $\delta(HC_i)$ is the hop count increase in the i-th scenario, and $\overline{F_k}$ is the mean value of F_k across the n simulated scenarios of prefix interception for destination AS_D. The same computation is performed for each F_k for link fail scenarios. The Pearson Correlation Coefficient measures the linear correlation between two variables. Its values range between 1 and (-1), where 1 indicates a total positive linear correlation, 0 indicates no linear correlation, and (-1) indicates a total negative linear correlation. We are interested in finding a factor F_k for which $\rho(F_k, \delta(HC))$ is close to 1 or to (-1).

Table 4.1 shows the computed correlations for 5 different destination ASes. The first three destination ASes are part of a simulated Internet comprised of 205 ASes, and the next two destination ASes are part of a simulated Internet comprised of 245 ASes. For each destination AS, ρ values computed for prefix interception scenarios are listed in the left column, and ρ values computed for link failure scenarios are listed in the right column. “NA” (not applicable) is used for destination AS-21183 because every source AS of AS-21183 has the same path length to AS-21183, meaning that the Pearson Correlation Coefficient is undefined. For example, we see that the Pearson Correlation Coefficient of F_2 with respect to hop count increase during prefix interception has both positive values (0.18 in AS-12925) and negative values (-0.19 in

AS-21182); hence it is inconsistent with respect to hop count increase during prefix interception. For F_4, we see a moderate negative correlation with hop count increase during prefix interception for all destination ASes.

Destination AS	AS-12925 (Internet of 205 ASes)		AS-21183 (Internet of 205 ASes)		AS-39026 (Internet of 205 ASes)		AS-41176 (Internet of 245 ASes)		AS-41225 (Internet of 245 ASes)	
	Interception	Link Failure	Interception	Link Failure	Interception	Link Failure	Interception	Link Failure	Interception	Link Failure
F_1: AS path length from source AS to closest Tier-1 AS	-0.175	-0.64	0.19	-0.24	-0.34	-0.77	-0.22	-0.06	0.2	-0.34
F_2: Difference of AS path lengths to closest Tier-1 AS	0.18	0.64	-0.19	0.24	0.33	0.77	0.22	0.06	-0.2	0.34
F_3: AS path length from source AS to destination AS	-0.36	-0.41	NA	NA	-0.39	0.18	-0.34	-0.26	-0.3	-0.34
F_4: Route length from source node to destination node	-0.31	-0.38	-0.45	0.75	-0.26	0.61	-0.36	-0.28	-0.39	-0.34

Table 4.1: The Pearson Correlation Coefficient between the hop count increase and F_1-F_4 for prefix interception and for link failure events.

From Table 4.1 we learn that F_3 and F_4 are the factors whose ρ values are closest to (-1) or to 1 for prefix interception scenarios. However, in order for AS_D to use F_3 to determine whether a sudden hop count increase should be attributed to prefix interception or to a link failure, AS_D needs to know the length of the AS path from AS_S to itself. Unfortunately, AS_D does not have this information. F_1 and F_2 are useless for the same reason. F_4, however, requires AS_D to know the hop count distance from AS_S to itself, which can be deduced from the TTL of packets received by AS_D. Since F_4 correlates well with hop count increase during prefix interception but not so well with link failures, we now study whether it can be used to distinguish between the two.

To analyze F_4, in Figure 4.2 we plot the dependency between the hop count increase and the initial hop count. In this figure, each (x, y) coordinate represents a simulated scenario in which the hop distance between the source and destination ASes is x, and it increases by y due to an interception attack or due to a link failure. The scenarios given in this figure are simulated using five different destination ASes. An interception attack scenario is marked by a circle and a link failure is marked by a triangle. The size of a circle or a triangle is proportional to its frequency. For instance, we see that interception attacks for which the initial route length is 7 and the hop count increase is 5 are more frequent than those for which the initial route length is 16 and the hop count increase is 15. Negative y values appear for routes whose hop count decreases, which is possible since BGP does not necessarily choose the shortest paths; hence, it is possible, though unlikely, that a route length decreases after an attack or a failure.

From Figure 4.2 we learn that in most cases, the hop count of packets sent by a source AS that is affected by either prefix interception or a link failure, increases or decreases by 2 or more. Moreover, we learn that most prefix interception scenarios cause a greater increase in hop count than do link failure scenarios, and that the hop count increase due to prefix interception increases only slightly with the increase in the initial hop count distance. While we cannot determine a clear threshold to distinguish between prefix interception and link fail scenarios, we determine a "best-effort threshold", defined as:

$$T(AS_S, AS_D) = \begin{cases} 3, & \text{for } \mathit{initial\_hop\_count}(AS_S, AS_D) \leq 10 \\ 2, & \text{for } \mathit{initial\_hop\_count}(AS_S, AS_D) > 10, \end{cases} \tag{4.2}$$

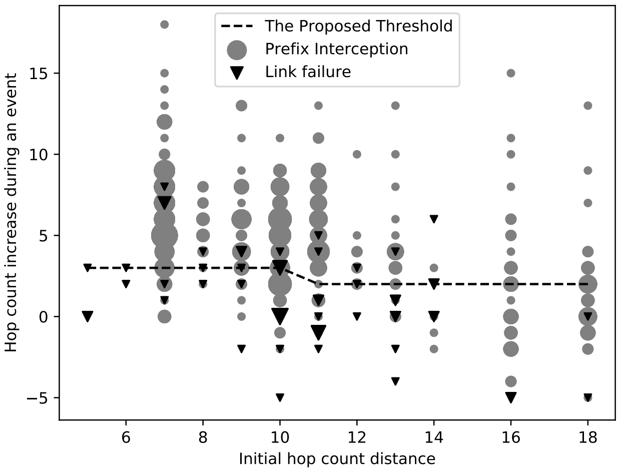

Figure 4.2: Analysis of F_4: Hop count increase over different initial hop count distances.

where $\mathit{initial\_hop\_count}(AS_S, AS_D)$ is the initial hop count of packets whose source AS is AS_S and destination AS is AS_D. The best-effort threshold is indicated in Figure 4.2 by the dashed line. According to this line, when the initial hop count is ≤ 10, an increase of 3 or more hops can usually be attributed to prefix interception, and an increase of 2 or less hops can usually be attributed to a link failure. When the initial hop count is > 10, an increase of 2 or more hops can usually be attributed to prefix interception, and an increase of 1 or less hops can usually be attributed to a link failure.

We now propose another factor, referred to as $\mathbf{F_5}$. F_5 is defined as the number of source ASes whose traffic is affected by the interception attack or link failure. We expect this number to be different for interception attacks and link failures because during an interception attack, the attacker announces an attractive new path, which is expected to be adopted by all the ASes that are close to the attacker and by some remote ASes as well. In contrast, an AS that withdraws one of its paths because of a failure in one of its inter-AS links is likely to affect the routing of close ASes but very unlikely to affect the routing of remote ones.

Table 4.2 shows the number of source ASes affected by prefix interception scenarios versus link failure scenarios. The results in each row are for a single destination AS. The first column indicates the number of source ASes that send traffic to the considered destination AS. The second and third columns indicate the mean and STD of the number of source ASes whose traffic is affected by prefix interception. The last two columns indicate the same statistics for link failures. For example, consider the statistics of a destination AS that receives traffic from 46 source ASes. For this AS, the average number of source ASes whose traffic is affected during prefix interception is 4.05, whereas the average number of source ASes whose traffic is affected by a link failure is only 2.75.

	Prefix interception		Link failure	
Number of Source ASes	Mean number of source ASes whose traffic is affected	STD of the number of affected ASes	Mean number of source ASes whose traffic is affected	STD of the number of affected ASes
6	1.61	1.19	1.14	0.34
8	1.75	1.43	1.69	1.26
13	2.03	1.9	1.81	2.48
40	3.78	5.1	3	5.48
46	4.05	7.59	2.75	4.97
56	6.28	7.56	2.39	3.88
99	9.45	22.17	2.61	5.62

Table 4.2: Number of source ASes whose traffic is affected by prefix interception vs. link failure.

In Figure 4.3 we plot the dependency between the number of source ASes and the mean number of source ASes whose traffic is affected by either prefix interception or link failure. The data in this figure is taken from Table 4.2. Prefix interception scenarios are marked by circles and link failure scenarios are marked by triangles. The solid line is the linear regression computed for prefix interception scenarios. The dotted line is a constant approximation of the linear regression computed for link failure scenarios. From Figure 4.3 we learn that when the number of source ASes increases, the number of source ASes whose traffic is affected by prefix interception increases as well. In contrast, the number of source ASes whose traffic is affected by link failure is a near-constant whose value is approximately 2.2. Moreover, we see that F_5 distinguishes well between the two root causes when the number of source ASes grows beyond 40.

Let *linear_regression_interception(n)* be the linear regression of the average number of affected source ASes during prefix interception (the solid line in Figure 4.3). The dashed line distinguishes between prefix interception and link failure scenarios when the number of source ASes is ≥ 40. This line is defined by the following function:

$$f(n) = \frac{linear\_regression\_interception(n) + 2.2}{3}. \tag{4.3}$$

The value “3” in the denominator was chosen empirically. Consider an AS that receives traffic from n source ASes. If the number of source ASes is is ≥ 40, F_5 will be used, and if the number of affected source ASes is $\geq f(n)$, the hop count increase is likely due to prefix interception. Recall that a destination AS considers a source AS as “being affected” if the hop count average computed for packets received from this source AS increases or decreases by 2 or more.

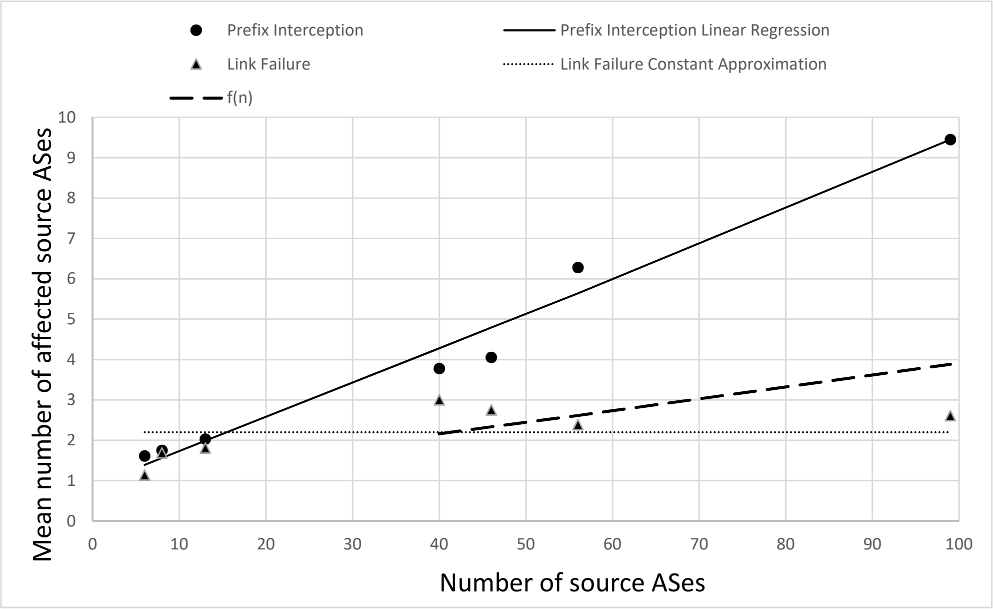

Figure 4.3: The mean number of source ASes whose traffic becomes affected by prefix interception vs. link failure.

If there are less than 40 source ASes, the destination AS can use F_4 in the following way. If the hop count of the packets received from a source AS is greater than $T(\mathrm{AS}_S, \mathrm{AS}_D)$, as defined by Eq. 4.2, the source AS is marked as “F_4-suspect”. Since we learned that F_4 does not clearly distinguish between prefix interception and link failure events, the destination AS counts the number of “F_4-suspect” source ASes. If this number is at least 50% of the total number of affected ASes, the destination AS flags for a possible prefix interception.

Consider a destination AS_D that wants to detect a prefix interception attack on its prefixes. AS_D traces the hop count of the packets it receives and maintains a moving average of the hop count for the sampled packets for each [originating AS, destination subset] pair. During every time period, AS_D checks all of its source ASes and marks those whose hop count increases by at least 2 as “affected." In addition, each affected AS whose hop count not only increases by 2 but also crosses the threshold $T(\mathrm{AS}_S, \mathrm{AS}_D)$ is also marked as “F_4-suspect”. If the number of source ASes from which traffic has been received during the last test period is ≥ 40, then the considered destination AS raises a flag for possible prefix interception if the number of affected ASes is greater than $f(n)$. If the number of source ASes from which traffic has been received during the last test period is greater than 10 and less than 40, then the considered destination AS raises a flag for possible prefix interception if at least 50% of the affected source ASes are also marked as “F_4-suspect”.

Algorithm 1. Executed by AS_D for detecting possible prefix interception

1. For each AS_i in the Internet and for every subnet prefix p_j of AS_D: HopCountList[AS$_i$, p_j] ← empty list.

2. Consider a sampled packet received at time t whose source AS is AS_i and destination subnet is p_j. Let H be the number of hops the packet has traversed, as observed from the packet's TTL. Then, add (H,t) to HopCountList[AS$_i$, p_j].

3. During every time interval τ, perform the following steps for every HopCountList[AS_i, p_j]:

   (a) Remove all the records that are older than $c \cdot \tau$ seconds (c is a constant integer).

   (b) Let k be the current interval. Then, set Average$_{i,j,k}(\tau)$ to be the average of the hop counts in HopCountList[AS_i, p_j].

   (c) If Average$_{i,j,k}(\tau)-$Average$_{i,j,k-1}(\tau) \geq 2$ or Average$_{i,j,k}(\tau)-$Average$_{i,j,k-1}(\tau) \leq -2$, mark AS$_i$ as affected .

   (d) If Average$_{i,j,k}(\tau)$ − Average$_{i,j,k-1}(\tau)$ is greater than the threshold $T(AS_i, AS_D)$, defined by Eq. 4.2, mark AS_i as F_4-suspect.

4. For each prefix p_j of AS_D:

   (a) If the number of source ASes is ≥ 40 and the number of affected ASes is ≥ $f(n)$, alert for prefix interception.

   (b) If the number of source ASes is > 10 and is also ≤ 40, and at least 50% of the affected source ASes are marked as F_4-suspect, alert for prefix interception.

We now evaluate the ability of Algorithm 1 to distinguish between prefix interception and link failure scenarios when it is executed by 7 destination ASes: AS-13355, AS-19043, AS-23192, AS-43887, AS-5538, AS-2607 and AS-378. We simulate 1459 prefix interception scenarios and 337 link failure scenarios. Note that there are more possible interception scenarios than link failure scenarios for a given destination AS because every AS in the network is a potential attacker in an interception scenario, where only a few ASes are positioned on the routing path between two ASes. All the simulation scenarios performed for this test are new. The results are given in Table 4.3(a). In this table, the first row indicates the destination AS. The second row indicates the number of source ASes that send traffic to the considered destination AS. Successful identification of a prefix interception scenario is considered a true positive. Misidentification of a link failure as a prefix interception is considered a false positive. Successful identification of a link failure is considered a true negative. Finally, failure to detect a prefix interception is considered a false negative. The interception detection rate is defined as $\frac{\sum \text{True Positive}}{\sum \text{True Positive}+\sum \text{False Negative}}$, i.e., the percentage of prefix interception events that are correctly identified. The link failure detection rate is defined as $\frac{\sum \text{True Negative}}{\sum \text{True Negative}+\sum \text{False Positive}}$, i.e., the percentage of correctly identified non-interception events.

Destination AS	13355	19043	23192	43887	5538	2607	378
Number of source ASes	35	38	40	43	51	65	84
True Positives	112	91	61	35	50	52	43
False Positives	12	6	5	8	8	10	10
True Negatives	26	30	29	48	45	45	55
False Negatives	104	98	95	198	87	217	216
Interception Detection Rate	51.86%	48.15%	39.1%	15.02%	36.5%	19.33%	16.6%
Link Failure Detection Rate	68.42%	83.33%	85.29%	85.71%	84.91%	81.82%	84.62%

(a) All tested scenarios.

Destination AS	13355	19043	23192	43887	5538	2607	378
Number of source ASes	35	38	40	43	51	65	84
True Positives	15	12	61	31	46	52	43
False Positives	2	1	4	8	6	10	10
True Negatives	0	0	0	0	0	18	0
False Negatives	8	10	1	7	6	12	15
Interception Detection Rate	65.22%	54.55%	98.39%	81.58%	88.46%	81.25%	74.14%
Link Failure Detection Rate	0%	0%	0%	0%	0%	69.29%	0%

(b) Scenarios for which the real number of affected source ASes is > 2.

Table 4.3: Evaluation of Algorithm 1.

Table 4.3(a) shows that Algorithm 1 only rarely identifies non-interception events as interception events. Unfortunately, the average interception detection rate of Algorithm 1 is low, ranging from 15.02% to 51.86%. This means that the algorithm only rarely detects prefix interception scenarios. However, for attacks with more than 2 affected ASes, Table 4.3(b) shows a significant increase in the interception detection rate, ranging from 55% to 98%. Moreover, when the number of source ASes is greater or equal to 40, the detection rate ranges from 74% to 98%. Since there are almost 60,000 ASes in the Internet, most are likely to have more than 40 concurrent source ASes.

We also note that attacks for which the real number of affected ASes is ≤ 2 have a relatively small magnitude [23]. From the attacker's perspective, conducting a prefix interception attack with a magnitude of 2 or less polluted ASes is difficult or even impossible. Consider an attacker that aims to intercept traffic sent from AS$_S$ to AS$_D$. In order to avoid being detected by our algorithm, the attacker must choose an attacker AS, AS_M, from which a bogus BGP announcement is sent such that:

1. There exists an AS path from AS$_M$ to AS$_S$ whose AS length is less than or equal to 2. Namely, AS$_M$ must be "close" to AS$_S$. Otherwise, the polluted route from AS$_S$ to AS$_M$ already contains more than 2 affected ASes.

2. AS$_S$ will accept the bogus announcement. This requires the routing policies enforced by the ASes on the AS path from AS$_S$ to AS$_M$ to accept the bogus announcement.

3. No more than one AS other than AS$_S$ will accept the bogus announcement. This is

non-trivial since AS$_S$ may forward the bogus announcement to its neighbors according to its BGP route export policy.

4. A non-polluted route remains between AS$_M$ and AS$_D$.

We believe that finding an AS that matches all these conditions is infeasible.

Chapter 5

Simulation Setup

To study the effect of prefix interception attacks and link failures on the hop count of affected packets, we hoped to use real traffic recorded during real prefix interception attacks or link failures. Unfortunately, we found no publicly available data recorded during prefix interception attacks. We therefore decided to use the SSFNet [13] network simulator.

We used CAIDA's datasets [4] as input for the simulator. CAIDA uses publicly available routing information datasets in order to infer the links and the relationships between ASes according to the propagation of routes from Tier-1 ASes to the rest of the Internet. We randomly chose a subset of the Tier-1 ASes as the topology seed, and then added layers of neighboring ASes until the desired topology size was obtained. The number of nodes within an AS was determined by the number of prefixes this AS owns, according to real BGP records [4], since more prefixes usually mean more nodes. The nodes exchange packets with randomly chosen nodes. Thus, an AS that owns 2 IPv4 /24 prefixes receives traffic from more source nodes than an AS that owns only 1 IPv4 /24 prefix, since the first AS has more destination nodes than the second.

The BGP routing rules of our ASes conform to the no-valley property and local preference policy [22]. Namely, a BGP router prefers a customer route over a peer route, and a peer route over a provider route. When a BGP router is notified about a new route to a given prefix by one of its peers or one of its providers, it propagates this route to its customers. If it is notified about a new route by a customer, it propagates this route to its peers and providers.

As indicated earlier, we determine the hop count each packet traverses using the TTL field and the algorithm proposed in [30]. Thus, our simulation needs to determine how to update the TTL when a packet crosses an AS. In [25], the internal structure of 8 Tier-1 ASes is studied. The authors report that these ASes decrement the TTL between 4 to 7 times (hops). In order to gain more information about the number of routers a packet needs to traverse in each AS, we used RIPE's Atlas [10], which is a framework of more than 10,000 nodes in the Internet. We analyzed the results of 450 traceroutes preformed by Atlas nodes using IP to ASN mapping [36].

Packets that traverse a source or a destination AS are routed through a single border router. However, packets that traverse a transit AS cross the AS from one border router to another border router of the same AS, or they are sent to the next AS directly by the first border router. Thus, in Table 5.1, which presents the TTL reduction rules used in our simulations, we distinguish between Transit ASes (1st row) and source/destination ASes (2nd row). Since Tier-1 ASes have a unique internal structure, e.g., they apply special techniques to reduce the number of routers [25], [43], we further distinguish between Tier-1 ASes and non Tier-1 ASes. This distinction is made according to Caida AS ranking [4].

We see in Table 5.1 that the TTL reduction of Tier-1 Transit ASes is 2 hops less than the TTL reduction reported by [25]. This difference can be attributed to new technologies, such as MPLS, deployed by Tier-1 ASes since [25] was published. When a packet crosses an MPLS network, its TTL is reduced only by 1 regardless of the number of MPLS switches it traverses. This might also explain why the number of routers traversed by a packet in the source or destination ASes is, surprisingly, larger than the number of routers traversed in a transit AS.

	Tier-1 Average Hop Count	Tier-1 STD	Non Tier-1 Average Hop Count	Non Tier-1 STD
Transit ASes	3.565	2.588	1.915	1.227
Source or Destination ASes	1.481	1.067	2.25	2.486

Table 5.1: TTL reduction rules in our simulations.

The technique we use for traceroute results analysis has some flaws, as reported by [27]. However, we are not interested in the exact TTL reduction in every AS, but rather in the total TTL reduction of a path, as observed by a destination AS. It is found in [24] that 65% of Internet paths are of 7 to 12 hops. In our simulations, we ended up with paths whose average length is 9.64, with an STD of 2.91. This implies that the average hop distance in our simulations is nearly the same as in [24], even though our simulations use far fewer ASes than exist in the Internet. This is probably because the Internet is comprised of a “core” set of ASes, with additional ASes connected to this core. Since our simulated Internet contains enough core ASes, and this core is responsible for the majority of hops a packet traverses in its route, the average hop distance in our simulations is similar to that in the Internet.

In our simulations, an event is counted as a prefix interception if the following conditions are met:

1. There is at least one polluted AS, namely, an AS that changes its routing tables according to a bogus route.
2. The hijacking AS, which is the one that announces a bogus route and receives traffic from polluted ASes, is able to route the hijacked traffic to the legitimate destination using a

non-polluted route.

The SSFNet simulator enables an AS to send bogus announcements to all of its neighbors. However, such an event is likely to pollute all neighboring ASes, making it very easy to simulate a prefix hijacking attack, but very difficult to simulate a prefix interception attack. Thus, as part of this work, we modify SSFNet to send a bogus BGP announcement to specific neighbors. An AS can claim ownership of a prefix that belongs to another AS by announcing the exact same prefix, a less specific prefix, or a more specific prefix. An analysis in [16] shows that announcing a more specific prefix would pollute every AS in the Internet, thereby creating a prefix hijacking attack but not a prefix interception attack. The authors also find that announcing a less specific prefix while the legitimate owner does not withdraw its prefixes will have no effect, i.e., prefix interception will not take place. Hence, in order to simulate interception attacks, the attacker announces exactly the same prefixes already announced by the owner AS. In each prefix interception scenario, a bogus announcement is sent to a specific neighbor so that a non-polluted route may still exist.

In [23], the authors find that ASes far from Tier-1 are the most vulnerable to prefix interception. Hence, these ASes are used as destination ASes in our simulations. In order to create as many prefix interception scenarios as possible while using a reasonable amount of computing resources, almost half of all possible prefix interception scenarios are computed for every destination AS.

The SSFNet simulator can simulate a failure in an inter-AS link or in an intra-AS link. In [21], the authors find that regardless of which link fails, at least one BGP message reports the failure 35% of the times. The authors conclude that when a BGP message is sent after a link failure, it is equally probable that either one of the links has failed. Thus, we can simulate link failures as both types. However, since simulating an inter-AS link failure requires less computing resources than simulating an intra-AS link failure, we choose to simulate link failures at AS boundaries. It is also found in [21] that link failures are not unique to a small set of specific "problematic" links, but are rather a general property of Internet paths. Thus, for a given source and destination ASes, we simulate a link failure for every inter-AS link that exists on their AS path.

Chapter 6

Conclusions

In this work we developed a scheme to be used by an AS to determine whether the traffic it receives was intercepted using prefix hijacking. When we began our study, we hoped to identify prefix interception by analyzing the TTL field in the received packets and detecting a sudden increase in the number of hops they traversed along their route. However, we discovered that it is very difficult to distinguish whether the cause of the increase was prefix interception or a legitimate route change after a failure. Further analysis showed that the number of source ASes whose traffic is affected by prefix interception is significantly higher from the number of source ASes whose traffic is affected by a link failure. Using this observation, we proposed a detection method for the considered attack. Using simulations, we found that the proposed method identifies prefix interception attacks in most of the cases, and that this method has a negligible error rate.

The method we proposed is sensitive to the selection of the detection threshold $T(\text{AS}_i, \text{AS}_D)$ defined by Eq. 4.2 and to the value of $f(n)$ defined by Eq. 4.3. The values given in our study have been optimized for the Internet as simulated in our experiments. We do not claim that these thresholds will always be good for distinguishing between a prefix interception and a link failure. However, we showed that by choosing the thresholds correctly, an AS can detect most of the interception attacks on its prefixes.

Since the method we proposed relies on hop count measurement, a sophisticated attacker may modify the TTL value in the packets it intercepts. In [46], the authors discuss two such modifications. First, an attacker can mimic the location of the source AS before the attack. However, this requires the attacker to add to the TTL field of the intercepted packets the number of hops along the original route minus the number of hops on the intercepted route. This requires the attacker to know, in real time, the original route of each packet it intercepts. Second, an attacker can change the TTL values of intercepted packets in a way that does not necessarily mimic the location of the source. However, such a change can be detected using other algorithms, such as the one proposed by [46].

Our simulation setup assumes a simplistic Internet routing model. In this method, an AS is represented by a single node, and the links between ASes form a directed AS graph. According to [37], this model does not support complex, real-world AS relationships. For example, an IXP (Internet exchange point) is a peering point to which multiple ASes are connected. A special BGP feature, called BGP Community Attribute [1], is used by ASes that are connected to IXPs to announce their route export policies. Our simulation model does not include this feature. Another example of a BGP feature not included in our model is route filtering [17]. This feature is used by an AS to explicitly reject routes that might otherwise be preferred by the local preference policies defined in our model.

Bibliography

[1] BGP communities attribute RFC. https://tools.ietf.org/html/rfc1997/.

[2] BGPmon network solutions. https://www.bgpmon.net/how-hacking-team-helped-italian-special-operations-\group-with-bgp-routing-hijack/.

[3] Breaking TOR can bring in the big bucks. http://www.forbes.com/sites/thomasbrewster/2015/11/12/earn-money-breaking-tor/.

[4] CAIDA datasets. http://www.caida.org/data/as-relationships/.

[5] Dyn research. https://dyn.com/blog/pakistan-hijacks-youtube-1/.

[6] Google transparency report. https://www.google.com/transparencyreport/https/.

[7] An infrastructure to support secure Internet routing. https://buildbot.tools.ietf.org/html/rfc6480.

[8] The new threat: Targeted Internet traffic misdirection. https://dyn.com/blog/mitm-internet-hijacking/.

[9] PEERING the BGP testbed. https://peering.usc.edu/.

[10] RIPE Atlas. https://atlas.ripe.net/.

[11] RouteViews. http://www.routeviews.org/.

[12] SecureWorks, early warning system. https://www.secureworks.com/research/bgp-hijacking-for-cryptocurrency-profit.

[13] SSFNET simulator. http://www.ssfnet.org/homePage.html.

[14] UK traffic diverted through Ukraine. https://dyn.com/blog/uk-traffic-diverted-ukraine/.

[15] Maria Apostolaki, Aviv Zohar, and Laurent Vanbever. Hijacking bitcoin: Routing attacks on cryptocurrencies. *arXiv preprint arXiv:1605.07524*, 2016.

[16] Hitesh Ballani, Paul Francis, and Xinyang Zhang. A study of prefix hijacking and interception in the Internet. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 265–276. ACM, 2007.

[17] Matthew Caesar and Jennifer Rexford. BGP routing policies in ISP networks. *IEEE Network*, 19(6):5–11, 2005.

[18] Gavriil Chaviaras, Petros Gigis, Pavlos Sermpezis, and Xenofontas Dimitropoulos. ARTEMIS: Real-time detection and automatic mitigation for BGP prefix hijacking. In *Proceedings of the 2016 ACM SIGCOM Conference*, pages 625–626. ACM, 2016.

[19] Avichai Cohen, Yossi Gilad, Amir Herzberg, and Michael Schapira. Jumpstarting BGP security with path-end validation. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 342–355. ACM, 2016.

[20] Iñigo Ortiz de Urbina Cazenave, Erkan Köşlük, and Murat Can Ganiz. An anomaly detection framework for BGP. In *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium*, pages 107–111. IEEE, 2011.

[21] Nick Feamster, David G Andersen, Hari Balakrishnan, and M Frans Kaashoek. Measuring the effects of Internet path faults on reactive routing. In *ACM SIGMETRICS Performance Evaluation Review*, volume 31, pages 126–137. ACM, 2003.

[22] Lixin Gao. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking (ToN)*, 9(6):733–745, 2001.

[23] Joseph Gersch and Dan Massey. Characterizing vulnerability to IP hijack attempts. In *Technologies for Homeland Security (HST), 2013 IEEE International Conference*, pages 328–333. IEEE, 2013.

[24] Forough Golkar, Thomas Dreibholz, and Amund Kvalbein. Measuring and comparing Internet path stability in IPv4 and IPv6. In *Network of the Future (NOF), 2014 International Conference and Workshop*, pages 1–5. IEEE, 2014.

[25] Ramesh Govindan and Pavlin Radoslavov. An analysis of the internal structure of large autonomous systems. *University of Southern California, CS Dept., Tech. Rep*, pages 02–777, 2002.

[26] Benjamin Greschbach, Gunnar Kreitz, and Sonja Buchegger. The devil is in the metadata. new privacy challenges in decentralised online social networks. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference*, pages 333–339. IEEE, 2012.

[27] Hamed Haddadi, Miguel Rio, Gianluca Iannaccone, Andrew Moore, and Richard Mortier. Network topologies: Inference, modeling, and generation. *IEEE Communications Surveys & Tutorials*, 10(2), 2008.

[28] Xin Hu and Z Morley Mao. Accurate real-time identification of IP prefix hijacking. In *2007 IEEE Symposium on Security and Privacy (SP'07)*, pages 3–17. IEEE, 2007.

[29] Geoff Huston and Randy Bush. Securing BGP with BGPsec. In *The Internet Protocol Forum*, volume 14, 2011.

[30] Cheng Jin, Haining Wang, and Kang G Shin. Hop-count filtering: an effective defense against spoofed DDoS traffic. In *Proceedings of the 10th ACM Conference on Computer and Communications Security*, pages 30–41. ACM, 2003.

[31] Stephen Kent, Charles Lynn, and Karen Seo. Secure border gateway protocol (S-BGP). *IEEE Journal on Selected Areas in Communications*, 18.4:582–592, 2000.

[32] Mohit Lad, Daniel Massey, Dan Pei, Yiguo Wu, Beichuan Zhang, and Lixia Zhang. PHAS: A prefix hijack alert system. In *Usenix Security*, 2006.

[33] Song Li, Haixin Duan, Zhiliang Wang, Jinjin Liang, and Xing Li. An accurate distributed scheme for detection of prefix interception. *Science China Information Sciences*, 59(5):052105, 2016.

[34] Robert Lychev, Sharon Goldberg, and Michael Schapira. BGP security in partial deployment: Is the juice worth the squeeze? *ACM SIGCOMM Computer Communication Review*, 43(4):171–182, 2013.

[35] Zhuoqing Morley Mao, Jennifer Rexford, Jia Wang, and Randy H Katz. Towards an accurate AS-level traceroute tool. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 365–378. ACM, 2003.

[36] Pietro Marchetta, Walter de Donato, and Antonio Pescapé. Detecting third-party addresses in traceroute traces with IP timestamp option. In *International Conference on Passive and Active Network Measurement*, pages 21–30. Springer, 2013.

[37] Matthew Roughan, Walter Willinger, Olaf Maennel, Debbie Perouli, and Randy Bush. 10 lessons from 10 years of measuring and modeling the Internet's autonomous systems. *IEEE Journal on Selected Areas in Communications*, 29(9):1810–1821, 2011.

[38] Xingang Shi, Yang Xiang, Zhiliang Wang, Xia Yin, and Jianping Wu. Detecting prefix hijackings in the Internet with Argus. In *Proceedings of the 2012 ACM Internet Measurement Conference*, pages 15–28. ACM, 2012.

[39] Shen Su, Hongli Zhang, Binxing Fang, and Lin Ye. Quantifying AS-level routing policy changes. In *Communications (ICC), 2014 IEEE International Conference*, pages 1148–1153. IEEE, 2014.

[40] Soon Tee Teoh, Ke Zhang, Shih-Ming Tseng, Kwan-Liu Ma, and S Felix Wu. Combining visual and automated data mining for near-real-time anomaly detection and analysis in BGP. In *Proceedings of the 2004 ACM workshop on Visualization and Data Mining for Computer Security*, pages 35–44. ACM, 2004.

[41] Tao Wan, Evangelos Kranakis, and Paul C van Oorschot. Pretty secure BGP, psBGP. In *NDSS*, 2005.

[42] Russ White. Securing BGP through secure origin BGP (soBGP). *Business Communications Review*, 33(5):47–53, 2003.

[43] Mark Winther. Tier 1 ISPs: What they are and why they are important. *IDC White Paper, NTT Communications*, 2006.

[44] Ying Zhang and Makan Pourzandi. Studying impacts of prefix interception attack by exploring BGP AS-PATH prepending. In *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference*, pages 667–677. IEEE, 2012.

[45] Zheng Zhang, Ying Zhang, Y Charlie Hu, Z Morley Mao, and Randy Bush. ISpy: detecting IP prefix hijacking on my own. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 327–338. ACM, 2008.

[46] Changxi Zheng, Lusheng Ji, Dan Pei, Jia Wang, and Paul Francis. A light-weight distributed scheme for detecting IP prefix hijacks in real-time. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 277–288. ACM, 2007.

על ה-Hop Count של חבילות. הניתוח מבוצע מנקודת המבט של Source AS אשר דוגם חבילות הנשלחות אליו במטרה לזהות האם עלייה פתאומית ב-Hop Count נובעת מ-Prefix Interception או מנפילה לגיטימית בקו שידור שמחבר בין ASes כלשהם באינטרנט. אנו לומדים שאכן קיימים מקרי Prefix Interception שבהם העליה ב-Hop Count היא משמעותית יותר מאשר במקרי נפילה בקו שידור, אך קיימים מקרים בהם לא ניתן לסווג את הסיבה שגרמה לעלייה ב-Hop Count.

אנו חוקרים מדוע מסלולים מ-Source ASes שונים מתארכים בצורה שונה בתרחישי Prefix Interception ובתרחישי נפילה בקווי שידור. לשם כך אנו מגדירים מספר גורמים ומ-חשבים את הקורלציה בין כל גורם לבין התארכות המסלול. אנו לומדים שהגורם שעבורו הקורלציה היא הגדולה ביותר הוא אורך מסלול הניתוב המקורי מהמקור ליעד. אנו מוצאים שאם המקור קרוב ליעד השינוי ב-Hop count משמעותי יותר מאשר אם המקור רחוק מהיעד. בנוסף, אנו מוצאים שלא ניתן להבדיל בצורה טובה בין Prefix Interception לנפילה בקו שידור רק באמצעות ניתוח אורך המסלול מהמקור ליעד.

גורם נוסף שחקרנו הוא מספר ה-Source ASes אשר חבילות שנשלחות מהם מושפעות מ-Prefix Interception או מנפילת קו שידור. אנו מוצאים שמספר ה-Source ASes אשר מושפעים מהתקפת Prefix Interception עולה ככל שמספר ה-Source ASes עולה, בעוד שעבור נפילת קו שידור מספר ה-Source ASes המושפעים מתנהג כקבוע. בנוסף אנו לומדים שכאשר מספר ה-Source ASes גדול או שווה ל-40, מספר ה-Source ASes המושפעים מפריד בצורה טובה בין מקרי Prefix Interception למקרי נפילה בקו שידור. לבסוף, אנו מציעים אלגוריתם לזיהוי מקרי Prefix Interception המבוסס על הממצאים שלנו, ומעריכים את ביצועיו על פני תרחישים מדומים של Prefix Interception ושל נפילות קווי שידור. אנו מראים שהאלגוריתם מגלה Prefix Interception בשיעור שנע בין 55 ל-98 אחוז במקרים שבהם מספר ה-Source ASes המושפעים גדול מ-2. בנוסף, כאשר מספר ה-Source ASes גדול או שווה ל-40, שיעור גילוי ה-Prefix Interception של האלגוריתם נע בין 74 ל-98 אחוז.

תקציר

BGP (Border Gateway Protocol) הוא חלק חיוני בתשתית האינטרנט, אך הוא פותח בשנות ה-80 עם מודעות חלקית לנושאי אבטחה. חוסר האימות של הודעות הנשלחות בפרוטוקול הופך אותו לפגיע למגוון תופעות, ובפרט ל-BGP Prefix Hijacking. בהתקפה זו, נתב BGP זדוני או פגיע להשתלטות מרחוק מכריז על נתיב ל-IP Prefix שאינה בבעלותו. כתוצאה מכך, חבילות המיועדות ל-Prefix זה מועברות בפועל לנתב התוקף. מקרה מיוחד של התקפה כזו מתרחש כאשר התוקף מצליח להעביר את התעבורה שנחטפה בחזרה ליעדה המקורי. מקרה זה מכונה לעתים קרובות Prefix Interception. התקפות Prefix Interception תועדו בפומבי מאז 2013, כאשר AS (Autonomous System) מבלארוס הצליח ליירט תעבורה אשר מסלולה המקורי מעולם לא היה צריך לעזוב את צפון אמריקה.

במאמר המוזכר בחיבור נמצא שרוב השינויים שמתרחשים בהעדפות הניתוב של AS ליעד מסויים מתבטאים בהחלפת השכן של ה-AS דרכו מנותבות חבילות לעבר היעד, כך שהחבילות תנותבנה לעבר שכן אחר במקום השכן המקורי דרכו הן נותבו. כיוון שמקרה זה דומה למקרה בו הנתיב ליעד העובר דרך שכן מסוים מוחלף לעבור דרך שכן אחר לאחר שקו השידור אל השכן הראשון נפל, אנו מאמינים שישנן שתי סיבות עיקריות בלבד לעלייה ב-Hop Count (מספר הנתבים שעוברת חבילה): Prefix Interception ונפילת קו שידור. בעוד שבהתקפת Prefix Interception החבילות מנותבות "פעמיים": פעם אחת אל התוקף ופעם שנייה אל היעד המקורי שלהן, אנו מעריכים שבנפילת קו שידור, המסלול מתארך כך שהחבילות "עוקפות" את קו השידור שנפל. אנו מצפים ש"עקיפה" זו תוביל להתארכות נמוכה יותר מאשר התארכות שנוצרת כתוצאה מ-Prefix Interception .

על מנת לחקור את שתי התופעות אנו משתמשים בסימולטור הפופולרי SSFNet. אנו בונים אינטרנט מדומה באמצעות נתונים לגבי קשרים בין ASes אשר הוסקו על ידי גוף המחקר CAIDA. באינטרנט שאנו מדמים, מספר התחנות בכל AS פרופורציוני למספר ה-IP Prefixes שיש ל-AS במציאות, כך ש-ASes שיש להם יותר Prefixes צפויים לשלוח ולקבל יותר תעבורה. כדי לאפשר הרצת סימולציות של מספר גדול של ASes שינינו את הסימולטור כך שיאפשר לכל נתב באינטרנט המדומה להפחית משדה ה-TTL בחבילה ערך מסוים, שאינו בהכרח 1. בנוסף, חקרנו אילו ערכים אנחנו צריכים להפחית משדה ה-TTL כדי לדמות בצורה טובה את האינטרנט. כך אנחנו נמנעים מלדמות מבנה פנימי מורכב של AS וחוסכים משאבי ריצה, בהם אנחנו עושים שימוש כדי להגדיל את גודל האינטרנט המדומה. כמו כן, ביצענו שינוי נוסף בסימולטור אשר מאפשר לדמות תרחישי Prefix Interception.

העבודה מנתחת את ההשפעות של תרחישי Prefix Interception ושל תרחישי נפילה בקו שידור

המחקר בוצע בהנחייתו של פרופסור ראובן כהן, בפקולטה למדעי המחשב.

תודות

ברצוני להודות למנחה שלי, פרופ' ראובן כהן, עבור הנחייתו המעולה. הנחייתו של ראובן היתה אדיבה, סבלנית ומעוררת השראה. אני שמח שנפלה בחלקי הזכות לעבוד עמו וללמוד ממנו.
בנוסף, ברצוני להודות למשפחתי ולחברי, על התמיכה והאכפתיות.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

גילוי BGP Hijacking באמצעות ניתוח TTL

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
מדעי המחשב

תמיר כרמלי

הוגש לסנט הטכניון --- מכון טכנולוגי לישראל
שבט התשע"ח חיפה פברואר 2018

גילוי BGP Hijacking באמצעות ניתוח TTL

תמיר כרמלי