Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Networked Systems
ETH Zürich — seit 2015

BGP Verification without Specification

Semester Thesis

SA-2020-21

Author: Kirill Meisser

Advisor: Rüdiger Birkner

Supervisor: Prof. Dr. Laurent Vanbever

February 2020 to June 2020

Abstract

BGP started on three sheets of paper drawn in 1989 by Yakov Rekhter and Kirk Lougheed and has become one of the most important protocols in today's Internet. Unfortunately, at the time of its creation nobody imagined how large the Internet would become. The security aspect of it was just an afterthought. Fast forward to 2020 and we find ourselves in a world with thousands of Autonomous Systems that all work together on a trust basis through BGP. Yet, trusting someone you do not know can be a dangerous game sometimes, leading to BGP hijacks and leaks. These can cause outages and be used for 'man in the middle' attacks that can become costly both to consumers and the ISPs. Until this day no widely deployed solution exists to protect us from them. We are in dire need of a security mechanism as the world is becoming more and more dependent on the Internet infrastructure. In this thesis, we propose a model that protects a single AS from leaks and hijacks using a score based system that relies on past BGP data, enabling it to classify incoming announcements into four groups: benign, suspicious, malicious and new. We then prove its proper operation by evaluating it against three famous incidents and discuss its future potential.

Contents

Chapter 1

Introduction

The Border Gateway Protocol (BGP) is one of the most important protocols and makes the Internet as we know possible. It has been introduced in 1994 and ties together an ever-growing number of Autonomous Systems. Unfortunately as great as this protocol is, it lacks any form of security which makes it vulnerable to BGP hijacks and leaks, these in turn cause outages and painful troubleshooting which can become very costly for ISPs and inconvenience the users. In my thesis I try to tackle this problem by developing a system that automatically detects malicious announcements and blocks them before they can cause any harm. In addition, the system is easily deployable as it can be inserted into existing BGP sessions.

1.1 Motivation

BGP is a trust based protocol and does not have any security mechanisms in place. A big issue is that it does not offer AS path verification and authentication, this enables anyone with access to a BGP-enabled router to announce routes to address spaces that they do not own. When this occurs we talk of a BGP hijack. The consequences of a hijack can differ: in best-case scenario traffic just takes more time to arrive at its destination, it gets worse when the traffic is 'black-holed' and services cannot be accessed and in worst-case scenario traffic is redirected to fake websites which try to steal your credentials. The latter one is exactly what happened in April 2018 when a Russian provider announced a group of IP addresses that usually belong to Route53 Amazon DNS servers and when myetherwallet.com users tried to log in to their wallets they unknowingly were giving their information to a fake website that stole a total of 152'000$ of cryptocurrency [3].

Up to this day there were various attempts at reaching BGP security such as RPKI and BGPsec but both of them failed to be adopted by the ISPs, the main problem being that they require considerable effort to be deployed. BGP is in dire need of a security mechanism and that is the reason why I tried to tackle this problem by using a new approach.

1.2 Goal

The goal of this thesis was to create a system that aims to protect a single AS from accepting wrong, suspicious or unintended routes and halt the propagation of these. To this end we decided to create a statistical model using past BGP data to train the model and thus get an understanding of what normal behaviour looks like. Once the model is trained we want to be able to classify the new announcements into three categories: benign, suspicious or bad. An additional requirement was that the system has to have minimal deployment effort in order to potentially succeed. After

doing so the task was to verify its correct operation by evaluating the model on three real world cases: Pakistan's YouTube Leak (February 24th, 2008), Safe Host's Leak to China Telecom (June 6th, 2019) and Rostelecom's Hijack (April 1st, 2020).

1.3 Overview

I started my journey by reviewing different approaches that were made in the past such as: RPKI, BGPsec and DISCO. The descriptions of the approaches can be found in Chapter 2, alongside with a refresher on BGP. We then look at the data sources and tools that were used in order to train the model. Using these we analyze some recent leaks and hijacks to gain more insight. Additionally, we studied the behaviour of BGP announcements of different prefixes to see which information and features we can extract to help us in our predictions. In Chapter 4 the model description and implementation are introduced, together with the different scoring criteria used to label the new announcements.

We later evaluate our system and successfully demonstrate that it is able to detect all malicious announcements in three major hijacks while discussing the overall performance. Finally, in Chapter 6 we look at potential optimizations and further work that can be done in the future.

Chapter 2

The Current State of BGP

In this chapter we begin by introducing BGP and the two important security threats: leaks and hijacks. We look at some recent examples of these threats and see what makes some announcements suspicious. Finally we look at the past approaches, what level of security they bring and why they have not prospered.

2.1 Getting to know BGP

BGP is the protocol that is used between different ASes to exchange network reachability information about the *prefixes* they can reach, where prefixes are an aggregation of IP addresses grouped together by their most significant bits in common. It is a path-vector routing protocol, meaning that when an AS starts announcing its prefix it adds itself to the AS path and becomes the first AS in that path, also known as the origin. All the ASes that receive and pass on the announcement will then prepend themselves to the AS path. In doing so, an AS that receives an announcement can theoretically see which ASes will be traversed to reach the destination. This procedure also enables an AS to quickly detect if it is already contained in the path and thus discard it to prevent loops [6]. BGP is by no means an optimum seeking protocol, it lets ASes implement routing policies by setting *BGP communities* and by doing so manipulate the flow of internet traffic, this can lead to cases where all ASes are physically connected but where some may find themselves disconnected in the BGP world.

BGP lacks any sort of security mechanism for validating the correctness of the AS path that it receives in the announcements, which means that the authenticity and correctness of these are in no way guaranteed, this makes the protocol vulnerable to misconfiguration and attacks such as *leaks* and *hijacks*, these can cause outages and prevent certain services to be reached until the configuration has been fixed. A *BGP leak* is the phenomenon where an AS announces a disallowed address space by accident (e.g. typo), this can cause traffic to be diverted and in some cases be lost in a "black-hole" [8]. A *BGP hijack* on the other hand is an event where traffic is intentionally diverted from the legitimate address space owner by announcing a more specific prefix or making an announcement that has other favorable metrics (e.g. path length) making it more preferable to the original route [3]. Hijacks can be very elaborate, for example to avoid suspicion an AS may append the true originating AS to an otherwise bogus path making it much harder to detect. In Chapter 3 you will see how some hijacks and leaks manifest themselves in real life and how we can detect them by observing suspicious behaviour such as AS path origin change and more specific prefix announcement.

Prominent hijacks happened in the last few years such as the YouTube Pakistan incident (Febru-

ary 2008), where access to the website was denied to millions of users and last year's European traffic hijack by China Telecom (June 2019). BGP leaks can also pose a great threat as can be noted by Google's leak to Verizon (August 2017) that had devastating effects that have been witnessed in Japan in form of a total outage [9].

2.2 What has been done so far?

BGP security has been an issue for a while now and there have been different ideas and implementations which throughout the years tried to help solve this problem, yet their deployment has seen little to no growth. First RPKI was introduced to validate the origin of a prefix. That was not enough, so BGPsec was introduced that authenticates the entire AS path. Both of these initiatives required significant changes to the current setup. Hence, DISCO was introduced which traded in the legal ownership of prefixes in favor of *de facto* posession which can be verified automatically and required less changes to the setup. In the following, we discuss all three in greater detail, highlighting their respective advantages and disadvantages.

2.2.1 Resource Public Key Infrastructure (RPKI)

In RPKI [7] Regional Internet Registries (RIRs) issue certificates to address space holders, where the certificates are typically renewed each year. RPKI enables filtering on originating AS and prefix size. The owners of a particular address range generate Route Origination Authorizations (ROAs) and sign them with the issued certificate's key. This binds their address prefix to an AS number, thus giving that AS permission to originate the owned prefix. In doing so ASes which use RPKI check the origin and whether it corresponds to the owning entity, if so they are marked as *valid*, if not they are marked *invalid* and can be filtered out. [11]

- **Advantages:** Security is gained by checking whether the announcement has indeed the right originating AS for the given prefix. This protects an AS from hijacks that use a different originating AS.
- **Disadvantages:** Someone with bad intentions can still append the authorized originating AS to a fake announcement, the infrastructure cannot detect this since it does not check the AS path. RPKI cannot protect against attacks in which the original origin is kept in the AS path.

2.2.2 Border Gateway Protocol Security (BGPsec)

In this protocol routers need to decide together whether they will use BGPsec [5] or not. By using BGPsec the routers replace the AS PATH attribute with BGPsec Path, this allows them to certify the legitimacy of the path by using cryptographic signatures which are added by every AS in the path. BGPsec is a heavier protocol where routers do all the work, thus requiring more computationally capable routers. [12]

- **Advantages:** The legitimacy of the AS path is secured.
- **Disadvantages:** For BGPsec to work there needs to be an unbroken path of routers which are all capable of communicating through it, else all the security information is gone and normal BGP is used once again. In addition BGPsec requires faster hardware which means a bigger investment.

2.2.3 Decentralized Infrastructure for Securing and Certifying Origins (DISCO)

Is a proposed approach where legal ownership of prefixes is traded in favor of *de facto* possession. The three main players in DISCO [4] are: agents, registrars and repositories. Agents have the task to attach a public key to the announcements. Registrars on the other hand are responsible for the DISCO-certificates, one is handed out to an AS when a certain number of registrars observe announcements with the same public key for a specified time interval. Finally the repositories store and distribute the certificates.

- **Advantages:** Protects from common attacks, fully automated and does not need coordination with others to be able to work.
- **Disadvantages:** Cannot certify ownership of unannounced IP prefixes and while an attack is occurring.

Chapter 3

Analysis of BGP Behavior

Before we can proceed to the design of the model, we analyze how normal and abnormal BGP behavior looks like. This will help us gain knowledge on how malicious announcements differ from the benign ones and ultimately help us decide what features we can use to train our statistical model. In this chapter we present to you the tools that were used to collect the information that was then fed to the model. Then we will look at some hijacks and leaks that were recently recorded using BGPMon [14] and conclude by analyzing 3 week long streams where we observe patterns that repeat themselves across prefixes.

3.1 Methodology

In this section we briefly introduce route collectors and CAIDA's BGP Stream [2], the tools that were used to gather the BGP data. Thanks to these, access to BGP data is greatly simplified and the gained insights allow us to build a statistical model.

3.1.1 Route Collectors

In order to get a statistical model one first of all needs data, but the problem is that BGP data is not shared all that willingly because ASes implement policies and they prefer to keep them secret. Yet, thankfully route collectors have been installed around the globe by organisations such as RIPE (Réseaux IP Européens) and Route Views [13], these help monitor and study BGP announcements and are publicly available. Each route collector peers with many ASes through one or more peering routers. The routers forward the best known paths for each prefix as a BGP announcement to the collector. The peering ASes additionally send their full routing tables in form of Routing Information Base (RIB) dumps, these are sent regularly every 2 or 8 hours depending on which organization the route collector belongs to. Finally the collector stores the announcements and RIB dumps, making it available even after several years have passed. Below you can see the attributes that are stored for each entry in the route collector:

- **Time:** the time at which the announcement was registered.
- **Type:** the type of entry, can be either 'R'(ribs), 'A'(announcement) or 'W' (withdrawal).
- **Collector:** the collector id from which we gather the data.
- **Peer AS:** the AS in question that peers with the collector and provides the data.

- **Peer address:** address of the peering router.
- **Prefix:** the address space in question.
- **Next-Hop:** the next hop address that is used to reach the destination.
- **AS Path:** the path that leads to the announced address space.
- **Communities:** attribute that can be used for traffic engineering or dynamic routing policies.

3.1.2 CAIDA BGPStream

Having the route collectors at our disposal we now need a way to access it and that is where CAIDA's BGPStream comes into play, this API enables us to access the records of the various collectors and apply different filters such as the desired time frame, record type, peer AS, paths containing certain ASes and many more. Finally, we can take the desired stream and create a table containing the entries and use this data to perform our statistical analysis.

3.2 Characteristics of BGP Hijacks and Leaks

In the following we will observe some recent examples of hijacks and leaks that I have found using BGPMon, a service which provides information about the latest hijacks and leaks around the globe. Using this information we then relied on CAIDA's BGPStream to observe the announcements that were shared in that time period.

- **First Case:** On the 21st of March 2020 at 14:10:23 UTC, a possible BGP hijack was detected by BGPmon. Prefix 176.222.61.0/24, is normally announced by AS208306 LARSA, IQ but was hijacked by AS 60663. The gathered data can be found in table 3.1, here we can witness that at the beginning we receive the usual announcements from the expected origin AS 208306 then the hijack occurs and the origin AS changes to AS 60663. BGP does not check the authenticity of the originating AS and thus the announcements propagate. For this possible hijack we observed the BGPstream at AS 24482 which peers with collector "route-views.sg" in Singapore.
- **Second Case:** Beginning on 21st of March 2020 at 15:21:54 UTC, a possible leak was detected for prefix 91.242.168.0/23, normally announced by AS199119 OOO-BARSTEL-AS, RU, the prefix was leaked by AS9002 RETN-AS, EU to AS3320 DTAG, DE. The announcements can be seen in 3.2, looking at the table we can notice how usually AS 9002 directly follows AS 13030 but now that AS 9002 leaked its route to AS 3320 it appears in the path. AS 3320 does not check the announcement and continues to propagate the announcement to the peering AS 13030. In this case the longer AS path or the fact that we usually don't get announcements for this prefix from this neighbor should have surged some suspicions in AS 13030. Yet, since there are no security mechanism the announcement propagates without any problems. To be fair, this kind of leak would be hard to detect since the AS path length decreased only by one and such a situation could also easily occur when for example one AS goes back online, announces the same prefix and we accept it because we prefer it.
- **Third Case:** On the 22nd of March at 2:47:56 UTC, another possible hijack happened. This time a more specific announcement 34.192.187.0/24 was announced by ASN 27742 (Amnet

Telecomunicaciones S.A., NI), usually the prefix 34.192.0.0/12 is announced by AMAZON-AES, US. In 3.3 one can see the hijack in action. In this case we can see two things that are suspicious: the first and most important one is that the origin AS changes, this alone is a big red flag, the second one is that a more specific prefix than usual is announced, such a trick is often used to attract and steal more traffic since routing is done based on the longest prefix match.

Time	Announced prefix	AS path
2020-03-21 9:07:58 UTC	176.222.61.0/24	24482 8932 39216 49571 208306
2020-03-21 9:07:58 UTC	176.222.61.0/24	24482 8932 39216 49571 208306
2020-03-21 14:10:24 UTC	176.222.61.0/24	24482 8932 39216 49571 60663
2020-03-21 14:10:52 UTC	176.222.61.0/24	24482 8932 39216 49571 208306

Table 3.1: Hijack of LARSA, IQ as seen from 27.111.228.159 (AS 24482)

Time	Announced prefix	AS path
2020-03-21 15:19:27 UTC	91.242.168.0/23	13030 9002 62067 199119
2020-03-21 15:19:27 UTC	91.242.168.0/23	13030 9002 62067 199119
2020-03-21 15:19:28 UTC	91.242.168.0/23	13030 9002 62067 199119
2020-03-21 15:19:28 UTC	91.242.168.0/23	13030 3320 9002 62067 199119

Table 3.2: Leak of OOO-BARSTEL-AS, RU as seen from 195.66.224.175 (AS 13030)

Time	Announced prefix	AS path
2020-03-21 22:15:51 UTC	34.192.0.0/12	51185 3356 16509 14618
2020-03-22 2:48:02 UTC	34.192.187.0/24	51185 3356 701 262206 27742

Table 3.3: Hijack of AMAZON-AES, US as seen from 195.66.227.90 (AS 51185)

As we have seen in the cases above, hijacks come in different flavors and some hijacks are more difficult to spot while others are as easy as checking if the origin AS corresponds to the usual one. In table 3.4 we ranked the types of hijacks based on the difficulty of detecting them, there you can see how intricate some hijacks can become. Leaks on the other hand are generally difficult to spot because they maintain the true origin AS and changes are only noticeable inside the AS path.

3.3 Announcement analysis

Before starting developing the model, we began analyzing some ASes and the behaviour of prefixes which were advertised to these, in particular we have looked at AS 3303 (Swisscom AG), AS 6762 (TIM Sparkle) and AS 5400 (British Telecom).

3.3.1 Announcement Distributions

In figures 3.1a, 3.2a and 3.3a you can find graphs that show the distributions of the announcements for the three different ASes, that is how many prefixes see a certain number of BGP updates. Additionally in figures 3.1b, 3.2b and 3.3b the corresponding cumulative distribution functions (CDFs) are plotted, only the prefixes that see at least one update are included. The mean values

Origin AS is mantained	More specific prefix is used	Difficulty in detecting the hijack
no	yes	The easiest to detect, both the origin and the more specific prefix cause suspicion.
no	no	Easy, though the same prefix gives a bit less suspicion.
yes	yes	Hard, only the more specific prefix and possibly the longer AS path are suspicious. The attacker might append himself in the middle of the AS path and perform a 'man in the middle' attack.
no	no	Very Hard, here the origin and prefix length are maintained. The only suspicions can arise from the AS path length being very different from the usual or seeing some strange ASes in the path that are not usually there.

Table 3.4: Ranking the difficulty of detection of the different hijacks based on the originating AS and prefix used

for the announcements are 6.92 for AS 3303, 16.46 for AS 6762 and 5.22 for AS 5400, it is to note that the graphs are truncated for visual representation and contain prefixes that have a maximum of 179, 29'780 and 363 announcements respectively, but these are small in number relative to the total and are thus omitted. For all three graphs we can notice the presence of modes, in particular we can observe that Swisscom has two modes that are pronounced, these indicate that announcements have a tendency to have a relatively small number of updates. This trend is also confirmed by the CDFs where we see a rapid increase for lower numbers of BGP updates.

3.3.2 Announcement Time Series

After the first analysis, we decided to try and look deeper into the announcements of different prefixes, the idea was to see whether one can witness patterns and similarities between them. In figures 3.4, 3.5 and 3.6 the announcements are again observed for the same peers and the same time period as before, we picked the prefixes based on the number of updates they have seen, that is some with minimum, mean and maximum values. Looking at the graphs we can notice that some prefixes update at the same time and have high correlation such as the prefixes 194.60.214.0/24 and 103.245.252.0/24 at AS 3303. Others see regular updates such as 93.175.149.0/24. Looking at it in more detail we observe that here there are periodic announcements followed by a withdrawal where the AS path attribute keeps changing between '3303 34019 12654' and '3303 6774 56665 12654' together with the community values .

In AS 6762 we note that there are continuous lines for prefixes such as 188.143.166.0/23 and 194.147.247.0/24, these are caused by the the continuous updates these prefixes see, 188.143.166.0/23 alone sees 29'780 updates during that week, here the cause is again the continuous periodicity of announcements and withdrawals where two AS paths '6762 31500 44050' and '6762 3216 31500 44050' are battling each other over and over again.

In conclusion we can indeed say that chunks of prefixes follow similar patterns, this could be intuitively explained by the fact that announcements come over the same infrastructure, thus for example a failure in some AS or link will lead to changes to multiple prefixes. Finding such correlations could be of advantage for filtering announcements, which is what we will be trying to do by clustering the prefixes in the next section.

3.4 Prefix clustering

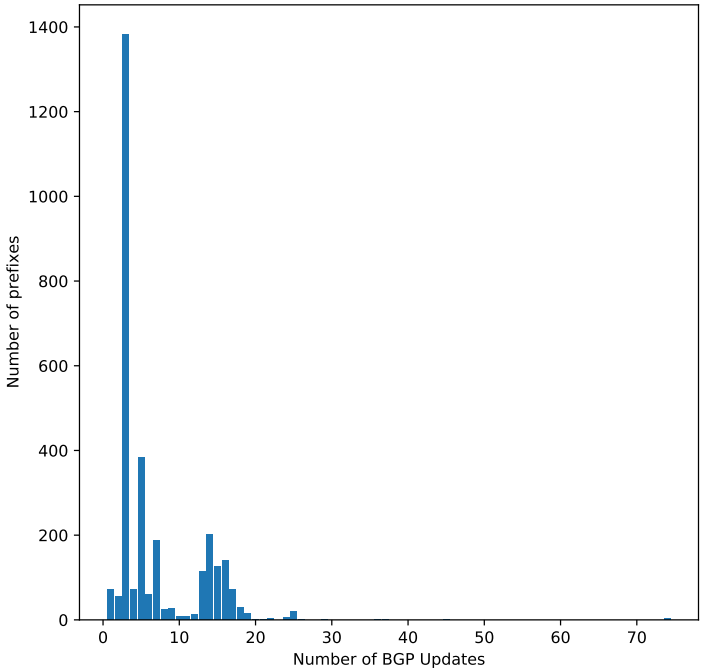
Seeing the time series graphs we noticed that groups of prefixes indeed follow a pattern, thus we attempted to cluster them into different groups and use the prefix's assigned cluster as a feature. To perform it one first needs some data points and a distance matrix which contains all the distances between them. You can have a look below to see how the data points are defined.

$$\begin{bmatrix} \textit{Changes in Next Hop} \\ \textit{Changes in AS Path} \\ \textit{Changes in Communities} \\ \textit{Median time between Announcements} \end{bmatrix}$$

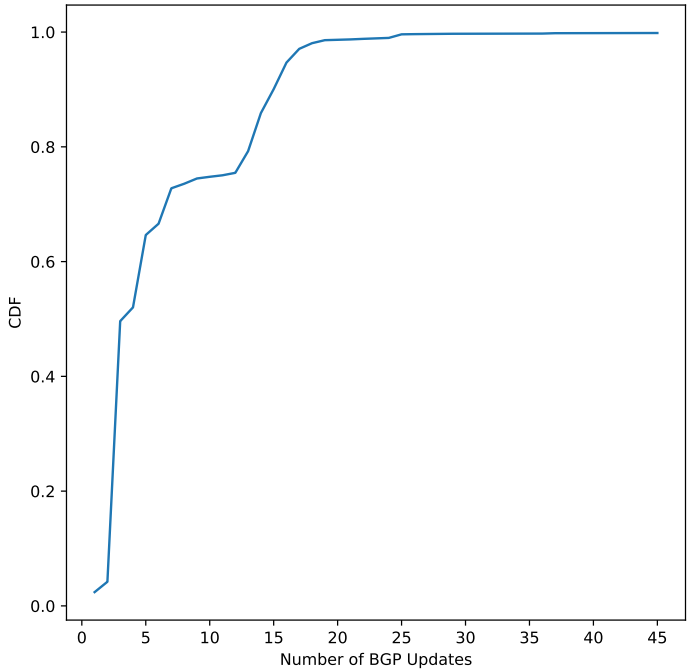
Now that we have mapped our prefixes into a space where we can define distances, we can construct the distance matrix and once we have it, we can rely on *scipy's* linkage function to create a linkage matrix, which then can be used to produce a dendrogram that can help us visualize and decide on the number of clusters. In this case there were simply too many prefixes in order to judge well the number that was needed and hence a heuristic named *"The Elbow Method"* was used to decide on the number of clusters. This heuristic picks the number of clusters so that adding another cluster does not add much to the modelling of the data anymore.

In figure 3.7a we can see the clustering in action, using the data from the Swisscom (AS 3303) stream taken between 2020-02-14 15:00:00 UTC and 2020-02-21 15:00:00 UTC. The clustering concludes by dividing the prefixes into three assigned groups and we can notice that indeed these groups show different behaviour, though a problem may be that the clusters generalize too much. We then tried a different approach, seen in figure 3.7b, here we utilized a different distance metric and ended up with 76 clusters, in this case the model seemed too sensitive.

In conclusion we can say that the clustering does gain us more insight and can be used as a further feature to detect malicious announcements, even though it is not perfect and has potential for further development. Additionaly some ASes can see more than 100'000 updated prefixes, in this case computing the distance matrix can become unfeasible on a regular machine, hindering us from clustering.

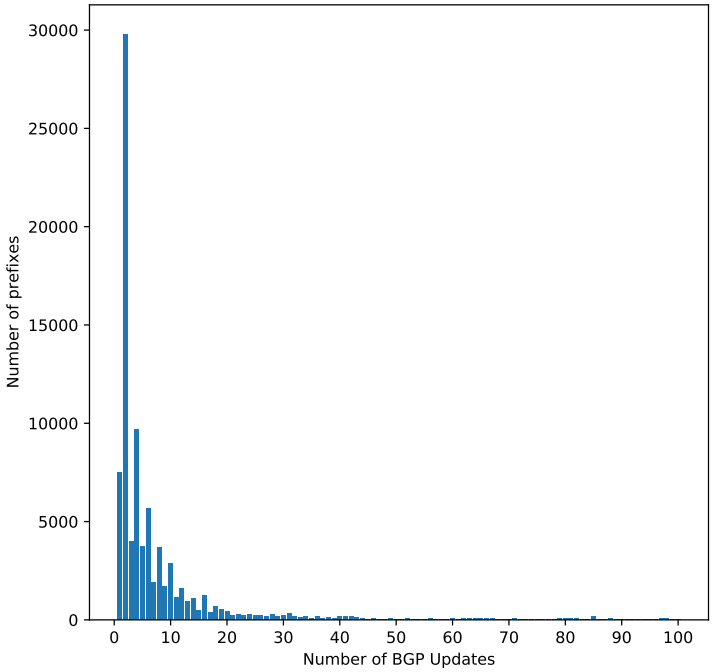


(a) Announcement distribution for AS 3303 with mean 6.92 and maximum of 179

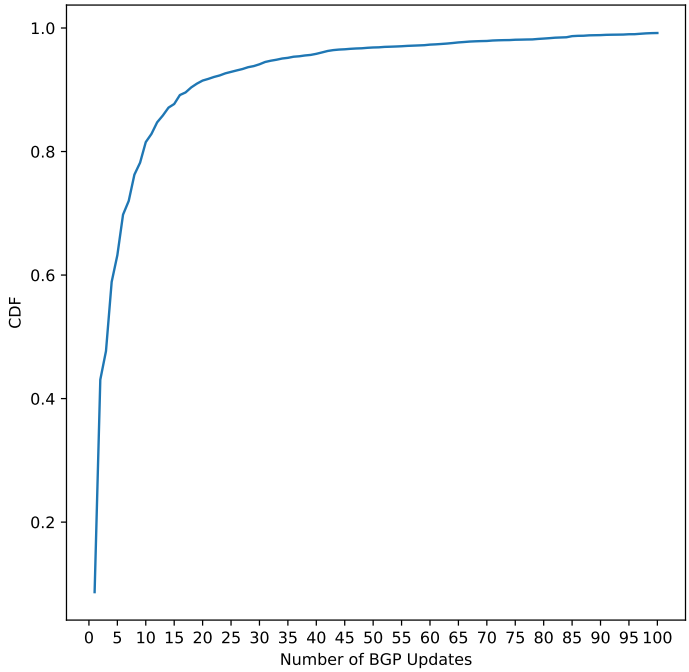


(b) The corresponding CDF, at less than 17 updates we already cover 94.7% of all the prefixes

Figure 3.1: Announcements received at rrc01 from peer AS 3303 with router IP 195.66.224.110 observed in the time interval 2020-02-14 15:00:00 UTC - 2020-02-21 15:00:00 UTC

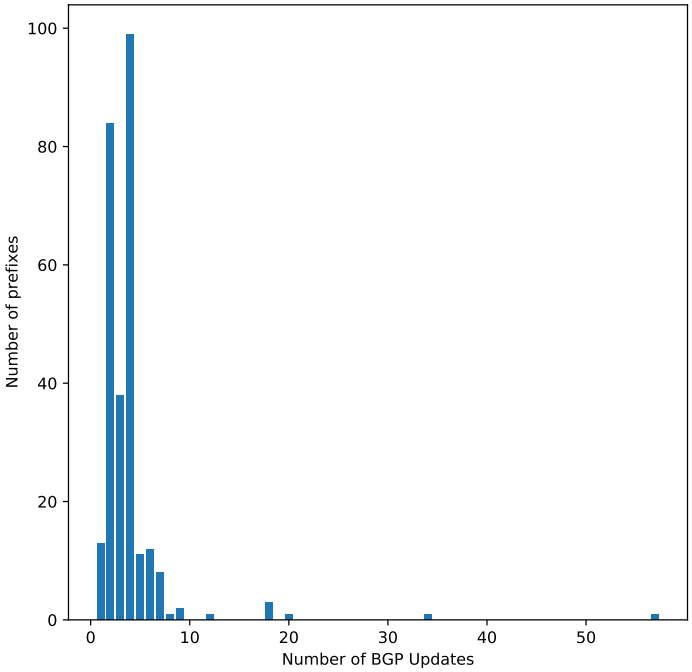


(a) Announcement distribution for AS 6762 with mean 16.46 and maximum of 29'780

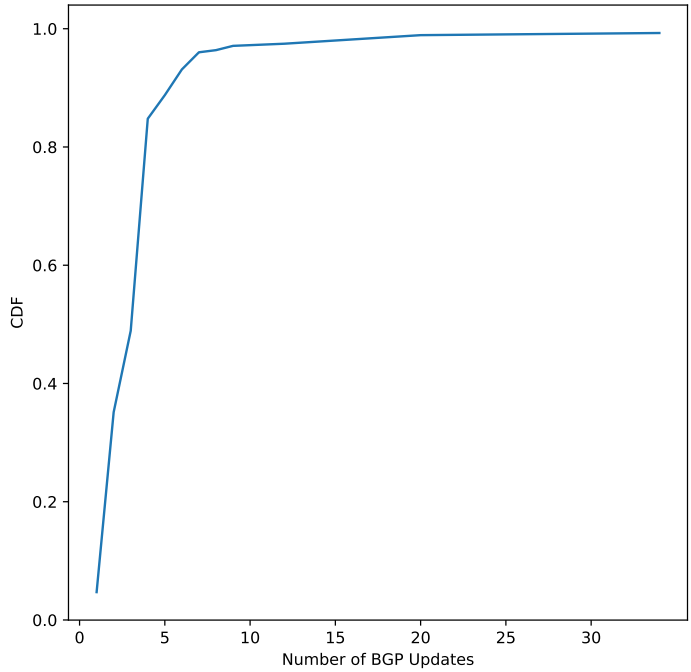


(b) The corresponding CDF, 95.1% of prefixes have less than 35 updates

Figure 3.2: Announcements received at route-views.linx from peer AS 6762 with router IP 195.66.224.15 observed in the time interval 2020-02-14 15:00:00 UTC - 2020-02-21 15:00:00 UTC



(a) Announcement distribution for AS 5400 with mean 5.22 and maximum of 363



(b) The corresponding CDF, at 7 announcements we have already covered 96.0% of the prefixes

Figure 3.3: Announcements received at rrc01 from peer AS 5400 with router IP 195.66.224.108 observed in the time interval

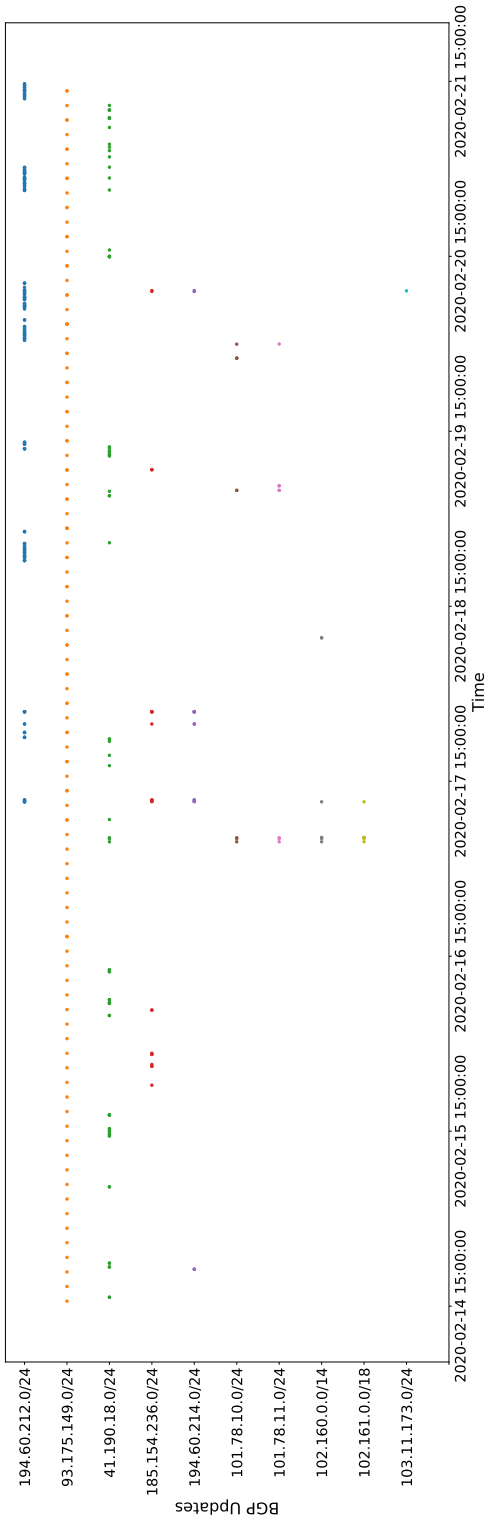


Figure 3.4: BGP updates received at rrc01 from peer AS 3303 with router IP 195.66.224.110 for some prefixes observed in the time interval: 2020-02-14 15:00:00 UTC - 2020-02-21 15:00:00 UTC

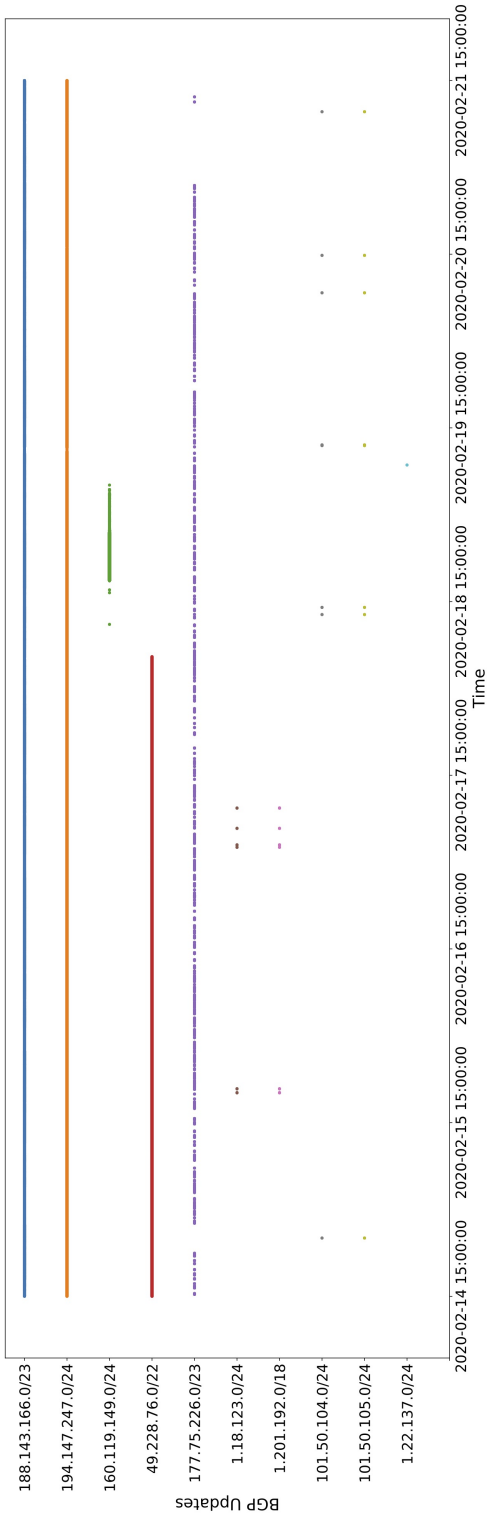


Figure 3.5: BGP updates received at route-view.linx from peer AS 6762 with router IP 195.66.224.15 for some prefixes observed in the time interval: 2020-02-14 15:00:00 UTC - 2020-02-21 15:00:00 UTC

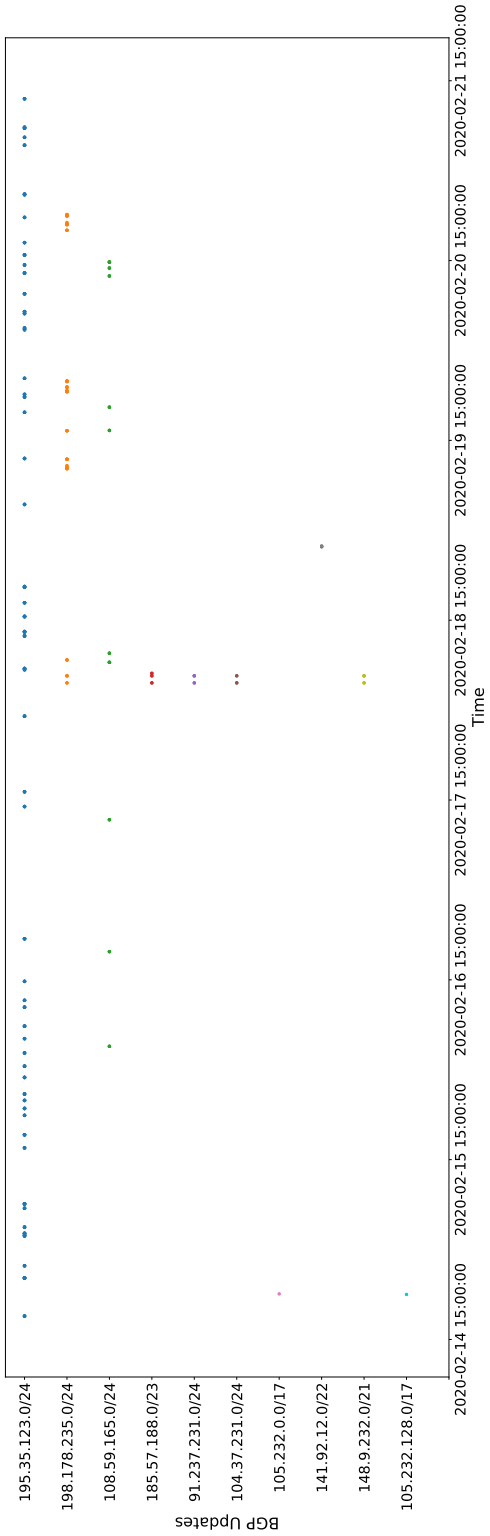
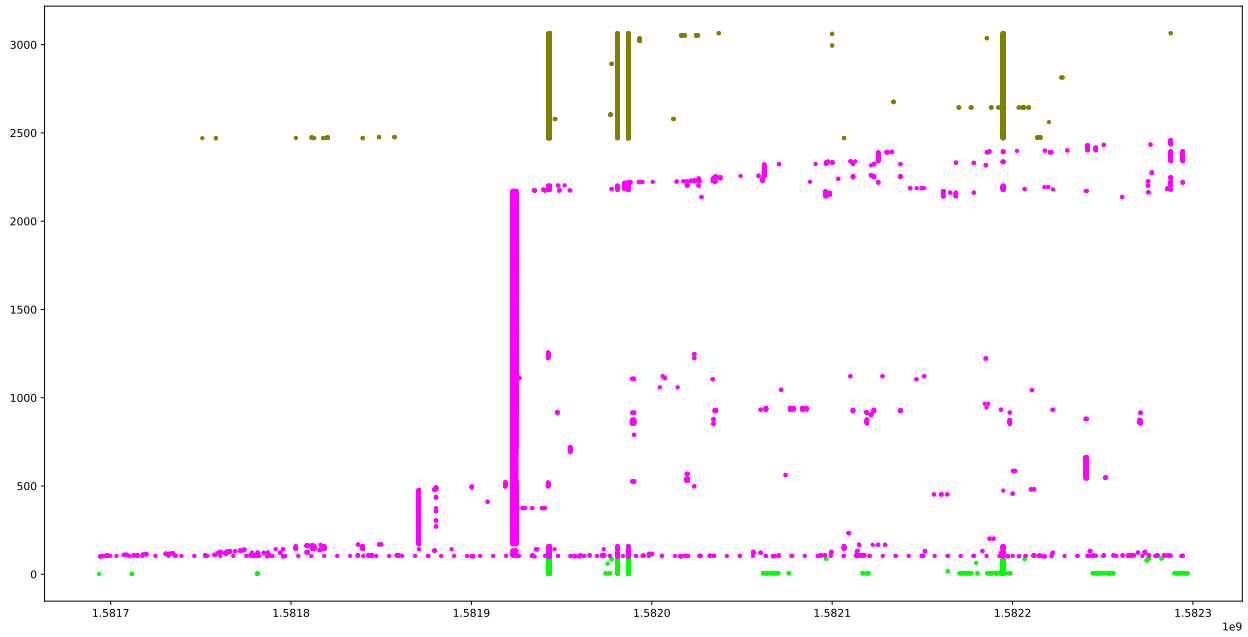
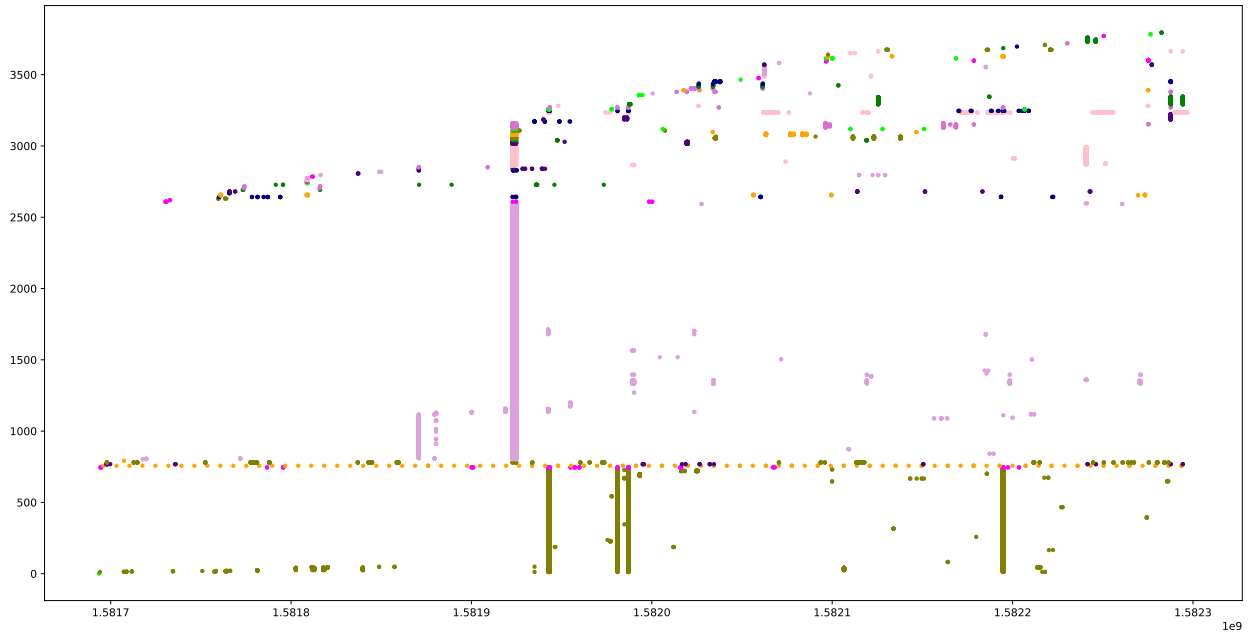


Figure 3.6: BGP updates received at rrc01 from peer AS 5400 with router IP 195.66.224.108 for some prefixes observed in the time interval: 2020-02-14 15:00:00 UTC - 2020-02-21 15:00:00 UTC



(a) Coarse grained clustering giving 3 distinct clusters



(b) Fine grained clustering giving 76 distinct clusters

Figure 3.7: Clustering using a week long stream for AS 3303 between 2020-02-14 15:00:00 UTC and 2020-02-21 15:00:00 UTC, on the horizontal axis we have the time and on the vertical axis are the indexed prefixes

Chapter 4

Model Design

The goal of this thesis was to create a system that aims to protect a single AS from accepting wrong, suspicious or unintended routes and halt the propagation of these. To this end we decided to create a statistical model using past BGP data to train the model and thus get an understanding of what normal behaviour looks like. Having a trained model we then take newly incoming announcements and score them based on different criteria, the score ultimately decides whether the announcement is benign and can pass, suspicious and needs further investigation or bad and dropped. In this chapter the different used features are explained and motivated, we then look at how these features contribute to the different criteria and finally how the final score is built.

4.1 Inputs and Feature Extraction

4.1.1 Inputs

Now that we have familiarized ourselves a bit with BGP we aim to find good features to help us detect malicious updates. The data that we will be using is the **BGP Routing Information Base (RIB)** and a **stream of BGP announcements**, together they train our model. For the test set we will be using another stream that is subject to filtering.

4.1.2 Feature Engineering

The features we will be using to detect malicious announcements can be split into three types: they can be **neighbor specific features** (e.g., number of routes a neighbor typically announces), **prefix specific features** (e.g., distribution over the AS path length - 99% of the time the AS path has length 4-6, all of a sudden the AS-path has length 1) and finally **prefix-group specific features** (e.g., cluster behavior).

Having now in mind what type of features we need, we first need to extract them from the inputs. To do this we take the stream and for every announced prefix we observe its evolution, in particular for every prefix we construct the following attributes which can be assigned to the three groups:

Neighbor Specific Features

- **Neighbor Prefix Table:** in this table we keep track of the number of prefixes that each of our neighbors announce, an example is shown in table 4.1. This information helps prevent leaks that are caused by our neighbors by looking whether they start announcing much more than usual.

Neighbor AS	Number of Announced Prefixes
3303	54908
6762	106433
5400	27890

Table 4.1: Neighbor Prefix Table

Prefix-group Specific Features

- **Changes in Next-Hop:** counts the number of changes in the attribute Next-Hop, this helps in clustering the prefixes since those that see a similar number of the same attribute changes are more likely to behave similarly and thus should be grouped together.
- **Changes in AS Path:** counts how often the AS Path attribute changes and has the same goal as the feature before, prefixes that have a large variance in the AS path should be clustered together.
- **Changes in Communities:** counts the number of changes in the community attribute, prefixes that update their communities often should be clustered together.
- **Median Arrival Time:** the median time between the announcements of a prefix. Having two prefixes that have a similar arrival time can help in deciding to which cluster they belong.

Prefix Specific Features

- **AS origin distribution:** the distribution of the origin throughout the week, calculated by

$$P(AS\ Path\ Origin = o) = \frac{Time\ intervals\ where\ AS\ Path\ Origin = o}{\Delta}$$

where Δ is the duration of the stream. Knowing the distribution we can see how probable a particular origin is for a given prefix. For example consider a prefix p that we follow for a week (Δ=1 week), now p has an originating AS A throughout 6 days and an originating AS B throughout 1 day, the distribution in such a case would be $P(AS\ Path\ Origin = A) = \frac{6}{7}$ and $P(AS\ Path\ Origin = B) = \frac{1}{7}$. As such this feature helps us in finding suspicious origins and deal with them accordingly.

- **AS path length distribution:** the distribution of the path length throughout the week, calculated by

$$P(AS\ Path\ Length = l) = \frac{Time\ intervals\ where\ AS\ Path\ Length = l}{\Delta}$$

where Δ is once again the duration of the stream. This feature aids us on detecting abnormal AS path lengths, this can come in handy for leaks or hijacks where the AS path length attribute changes dramatically caused by the redirection of traffic.

Score	Verdict
$S = 0$	New
$0 < S \leq 30$	Malicious
$30 < S \leq 70$	Suspicious
$70 < S \leq 100$	Benign

Table 4.2: An announcement is considered malicious if it has a score between 0 and 30, suspicious if it is between 30 and 70 and benign if it is above 70. An announcement for a previously unseen prefix receives a score of zero.

4.2 Scoring System and the Maliciousness Test

Now that we have the data we can finally start filtering the announcements. In this thesis we propose a scoring system which assigns scores between 0 and 100 to each incoming announcement, where a lower score means that there is a higher probability that the incoming announcement is malicious. Given the score (S) the announcements are classified into four classes based on table 4.2, where the thresholds can be adapted according to the operator's requirements.

In the model the announcements that receive a malicious label are dropped, the announcements labeled suspicious or new are forwarded to an operator for further inspection and announcements labeled benign are let through.

4.2.1 The Criteria

Throughout the thesis we developed five criteria which are combined together to form the total score S, they are:

- **AS origin criterion (OC):** given a new announcement, the criterion extracts the AS origin and checks what probability it is given in the AS origin distribution, the probability is then returned. We use this criterion to check whether the originating AS of the announcement is really the one that usually announces the prefix.

- **AS path length criterion (PC):** given a new announcement, the criterion extracts the AS path length and checks what probability it is given in the AS path length distribution, the probability is then returned. The criterion penalizes path lengths that deviate from the usual ones, in doing so we try to penalize potential leaks which alter the AS path length.

- **Neighbor criterion (NC):** given a new announcement, the criterion extracts the neighboring AS that sent the announcement. The criterion then counts the number of announcements that it received from the neighboring ASes in the last time period and checks whether any of them have exceeded the threshold, that is $\lambda \cdot$(Number of Announced Prefixes), where λ is a free parameter. If that is the case the criterion returns 1, else it returns 0. The goal of this criterion is to detect BGP leaks by our neighbors and decrease the score if that is the case.

- **Cluster criterion (CC):** given a new announcement, the criterion checks if the announced prefix has been previously assigned a cluster and if so the criterion looks up whether recent announcements were also part of the same cluster. The criterion returns

$$\frac{Number\ of\ prefixes\ from\ the\ same\ cluster\ that\ changed\ recently}{Total\ number\ of\ prefixes\ that\ were\ announced\ recently}$$

and 0 if the new announcement does not have an assigned cluster. The goal of this criterion is to further reward prefixes that are updated together with the prefixes in the same cluster since it is likely that they see similar updates.

Now that we have all the criteria, we can construct the score as:

$$\boxed{S = \alpha \cdot OC + \beta \cdot PC + \gamma \cdot CC - \delta \cdot NC} \tag{4.1}$$

where $\alpha + \beta + \gamma = 100$ and $0 < \delta < 100$

4.2.2 Case partition and scoring

When giving a score to an announcement we have to first decide to which category it belongs, that is because the criteria used to score the announcement are dependent on the category, since the available data is not the same for all prefixes. The five categories are:

1. The announcement is an update and the prefix does not appear in the RIBs and the stream.
2. The announcement is an update and the prefix appears in the RIBs but not in the stream.
3. The announcement is an update and the prefix appears in the stream but not in the RIBs.
4. The announcement is an update and the prefix appears in both the RIBs and the stream.
5. The announcement is a withdrawal

Now lets see how we score the different categories:

1. For this category there is little we can do as our AS has never seen this prefix, the only thing we can do is check whether the prefix is a more specific one to an already known one, if also the origin AS matches to the less specific prefix's origin then in that case we set the score to

   $$S = L - \delta \cdot NC$$

   where L is an *initial score level* which can be modified, else we set $S = 0$ since we don't know anything about it.

2. Here we have a little more data and we can compare the new announcement's origin AS and path length to the entry in the RIB, additionally we can use the NC to check for leaks. Since we do not have a stream, the distributions for the origin AS and path length are simplified, that is if the announced origin AS corresponds to the one in the RIB then we have $OC = 1$, else $OC = 0$. The same goes for the PC, where we have $PC = 1$ if the path length corresponds to the RIB path length and $PC = 0$ otherwise. Hence, we can write:

   $$\alpha \cdot OC + \beta \cdot PC = \begin{cases} 5 & , origin \neq RIB\ origin\ \cap\ path\ length \neq RIB\ path\ length \\ \beta & , origin \neq RIB\ origin\ \cap\ path\ length = RIB\ path\ length \\ \alpha & , origin = RIB\ origin\ \cap\ path\ length \neq RIB\ path\ length \\ 100 & , origin = RIB\ origin\ \cap\ path\ length = RIB\ path\ length \end{cases}$$

   and the total score is calculated as

   $$S = \alpha \cdot OC + \beta \cdot PC - \delta \cdot NC$$

3. In this case we have the training stream data at our disposal, which means we can utilize all the criteria that we described before, the score in this case is calculated as

$$S = \alpha \cdot OC + \beta \cdot PC + \gamma \cdot CC - \delta \cdot NC$$

where $\alpha + \beta + \gamma = 100$ as previously mentioned and δ a free parameter.

4. This case is the same as the third because the RIB data does not add anything since it is outdated and we thus use the same scoring as in point three.

5. The last case is the one that handles the withdrawals, here we decided to not have any penalization since withdrawals cannot cause any harm and even if we ignore the withdrawal and continue sending our packets then they will most probably be dropped. Hence:

$$S = 100$$

4.3 Output

In the last section we have seen how the scoring and filtering works, now all that is left to do is apply it. In the implementation we take a desired stream as our test set and label all the incoming announcements accordingly. The output of the model consists of a table that contains the original announcements with 5 additional columns: OC score, PC score, NC score, CC score and finally the verdict.

Announcement	OC score	PC score	NC score	CC score	S	Verdict
...	63	28	0	7	98	Benign
...	25	37	0	0	62	Suspicious
...	44	16	-50	0	10	Malicious

Table 4.3: Example of an output

Chapter 5

Evaluation

To evaluate the performance of the model, three prominent and documented cases were studied. In this chapter we will go through one by one and see how our model holds up against them. Finally we conclude with a discussion about the results.

5.1 Case Studies

5.1.1 Setup

The original setup for these case studies was to take a RIB dump at the beginning of the training period, then train for a week using the stream and finally test the model on a 24 hour stream on the day of the incident. Unfortunately, this was not always possible, in the second and third case the training set and test set had to be shortened in order to have acceptable computing times. The used time periods are reported for each case below.

5.1.2 Pakistan's YouTube Leak (February 24th, 2008)

On February 24 the Pakistani government decided to censure YouTube in the country. To do so Pakistan Telecom (AS 17557) introduced a more specific 208.65.153.0/24 routing table entry to attract all Pakistani YouTube traffic to itself, in doing so they accidentally leaked the prefix to one of their providers (PCCW, Hong Kong) which propagated the more specific prefix to its neighbors, creating a wave of false advertisements and "black-holing" the traffic. YouTube (AS 36561) noticed the drop in traffic and also began announcing /24 prefixes [1]. YouTube could have also started announcing /25 prefixes but the problem with that is that unfortunately most of the Internet routers reject routes more specific than /24 for security reasons.

In this case study, we look at the announcements arriving at KPN B.V.(AS 286) connected to the route collector *rrc01* through the peering router 195.66.224.54.

- Inputs:
  - RIB from 2008-02-17 00:00:00 UTC
  - Training stream from 2008-02-17 00:00:00 UTC - 2008-02-23 23:59:59 UTC
  - Test stream from 2008-02-24 00:00:00 UTC - 2008-02-24 23:59:59 UTC
- Parameters
  - $\alpha = 70,\ \beta = 30,\ \gamma = 0,\ \delta = 50,\ L = 60,\ a = 0.6\ and\ b = 0.4$

Time	Type	Announced prefix	Next Hop	AS path	Origin Criterion	Path Criterion	Neighbor Criterion	Score	Verdict
1203878902.0	A	208.65.153.0/24	195.66.224.167	286 3491 17557	0.0	0.0	0.0	0.0	Malicious
1203886357.0	A	208.65.153.0/24	195.66.224.54	286 3491 17557	0.0	0.0	0.0	0.0	Malicious
1203886387.0	A	208.65.153.0/24	195.66.224.54	286 3549 36561	60.0	0.0	0.0	60.0	Suspicious

Table 5.1: 208.65.153.0/24 announcements that we seen at AS 286

The cluster criterion could not be used here since there are 103'434 prefixes that were updated during the week and the machine ran out of memory while computing the distance matrix. While scoring the test stream there were found to be 1'797 malicious announcements, 12'914 suspicious ones, 131'866 benign ones and 429 completely new ones out of a total of 147'022. From the 1'797 malicious announcements we took a look at 20 in more detail to see the reasons why they were labeled as such. We found that 10 of them had a bad origin AS and a bad AS path length, where with bad we mean that most of the time it was announced with a different origin and path. Then 5 of these had a correct origin AS but the NC kicked in and lowered the score below 30. Lastly, there were 3 that had a completely different origin and path length and 2 that had a completely different origin but the same path length.

In the following we extracted the entries that contained 208.65.153.0/24 as the announced prefix from the scoring table and these can be seen in table 5.1. As we can see our model correctly labels the first two announcements where the originating AS is Pakistan Telecom as malicious and then labels the now more specific true YouTube announcement as suspicious, this is due to the fact that our model knows that 208.65.153.0/12 is normally announced by YouTube, now that it has received a longer prefix it checked whether the originating AS is the same as the one for the smaller prefix, which in this case was true.

5.1.3 Safe Host's Leak to China Telecom (June 6th, 2019)

Last year a big leak occured where a Swiss data center Safe Host SA (AS 21217) leaked more than 70'000 routes to China Telecom (AS 4134), which propagated the routes further globally and redirected Internet traffic destined to European ISPs through China. The route leak had many routes that circulated for more than 2 hours and impacted large European networks such as Swisscom (AS 3303), KPN (AS 1136) of the Netherlands, and Bouygues Telecom (AS 5410) of France [10].

In this case study we look at the announcements arriving at EX Networks Limited (AS 39122) connected to the route collector *rrc01* through the peering router 195.66.226.97.

- Inputs:
  - RIB from 2019-06-03 00:00:00 UTC
  - Training stream from 2019-06-03 00:00:00 UTC - 2019-06-05 23:59:59 UTC
  - Test stream from 2019-06-06 08:00:00 UTC - 2019-06-06 13:00:00 UTC
- Parameters
  - $\alpha = 70,\ \beta = 30,\ \gamma = 0,\ \delta = 50,\ L = 60,\ a = 0.6\ and\ b = 0.4$

For this case the train and test stream had to be further decreased so that the computation time was tolerable. Hence, we decided to only take 3 days of training data and for the test data only 5 hours of stream that was registered around the time of the incident, this left us with 87'886

Time	Announced prefix	AS path	Origin Criterion	Path Criterion	Neighbor Criterion	Score	Verdict
1559814329.0	94.156.255.0/24	39122 1299 4134 21217 21217 21217 21217 21217 21217 25091 34224	0.0	0.0	0.0	0.0	New
1559814329.0	62.69.144.0/23	39122 1299 4134 21217 21217 21217 21217 21217 21217 13237 201785	0.0	0.0	0.0	0.0	New
1559819433.0	191.6.2.0/24	39122 3356 4134 21217 21217 21217 21217 21217 21217 25091 263009 263009 263009 262822 263382 263545	5.13	6.0	0.0	11.13	Malicious
1559819433.0	191.6.3.0/24	39122 3356 4134 21217 21217 21217 21217 21217 21217 25091 263009 263009 263009 262822 263382 263545	5.13	6.0	0.0	11.13	Malicious
1559814329.0	202.158.3.0/24	39122 174 4134 21217 21217 21217 21217 21217 21217 25091 55818 38158 4787	60.0	0.0	0.0	60.0	Suspicious
1559814329.0	193.171.255.0/24	39122 1299 4134 21217 21217 21217 21217 21217 21217 13237 1120	70.0	0.0	0.0	70.0	Suspicious
1559818022.0	203.115.64.0/24	39122 2914 4134 21217 21217 21217 21217 21217 21217 25091 9498 133661 133696 23682	70.0	6.0	0.0	76.0	Benign
1559819433.0	191.6.0.0/23	39122 3356 4134 21217 21217 21217 21217 21217 21217 25091 263009 263009 263009 262822 263382 263545	70.0	6.0	0.0	76.0	Benign

Table 5.2: Extracted announcements from the test stream that contained the leaking AS 21217 (Safe Host SA) and AS 4134 (China Telecom) which propagated the announcements further

announcements that were scored in total. Out of them, 1'974 were malicious, 13'484 were labeled suspicious, 70'692 were benign and 1'616 were completely new ones.

Now, to evaluate the model and find the announcements that were part of the leak I selected the ones that contained AS 4134 and AS 21217 in their paths, in table 5.2 we can see some demonstrative examples of the total 3'902 prefixes leaked. Of those 3'902, 1'243 were labeled new, 37 were labeled malicious, 2'596 were suspicious and 26 were labeled as benign. Hence, most of the leaked routes were detected by our system and labeled rightly as new, suspicious or even malicious. In table 5.2 we can also see the reason why some announcements were labeled the way they are, for example the two malicious announcements had a bad origin AS and a bad path length. For the suspicious ones we have once the case where the system detected a more specific prefix with the right origin AS giving it a score of 60 and the second one where the prefix was detected in the RIB and the origin was correct but the path length did not correspond. Finally, for the benign ones the origin AS corresponded completely to the one during the training but the AS path length has a low score because it is not contained in the AS path length distribution, which is expected since we have a leak and the path length changes drastically. By looking at the table we can also make another interesting observation, we can see how AS 21217 continues to append itself to the path hoping that this will stop other ASes from accepting the leaked routes, unfortunately this did not happen.

In conclusion we can say that the model performed well in this case study by detecting most of the leaked routes either as new, suspicious or malicious and letting only a few slip by as benign ones.

5.1.4 Rostelecom Hijack (April 1st, 2020)

This April Rostelecom (AS 12389), a Russian state-owned telecommunications provider hijacked the traffic of more that 200 content delivery networks such as Google, Amazon, Facebook and Akamai. The incident lasted roughly an hour and affected more than 8'800 routes [15].

This time once again we observe the announcements arriving at EX Networks Limited (AS 39122) connected to the route collector *rrc01* through the peering router 195.66.226.97.

- Inputs:
  - RIB from 2020-03-29 00:00:00 UTC
  - Training stream from 2020-03-29 00:00:00 UTC - 2020-03-31 23:59:59 UTC
  - Test stream from 2020-04-01 00:00:00 UTC - 2020-04-01 23:59:59 UTC
- Parameters
  - $\alpha = 70,\ \beta = 30,\ \gamma = 0,\ \delta = 50,\ L = 60,\ a = 0.6\ and\ b = 0.4$

No cluster criterion was used and we trained only for 3 days, still we managed to have a 24 hour test set that was scored in reasonable time. During the 24 hour test stream a total of 780'563

Time	Type	Announced prefix	AS path	Origin Criterion	Path Criterion	Neighbor Criterion	Score	Verdict
1585769267.0	A	31.13.69.0/24	39122 174 20764 12389	0.0	0.0	0.0	0.0	New
1585769297.0	W	31.13.69.0/24					100.0	benign
1585769358.0	A	31.13.69.0/24	39122 174 20764 12389	0.0	0.0	0.0	0.0	New
1585769448.0	A	31.13.69.0/24	39122 3356 174 20764 12389	0.0	0.0	0.0	0.0	New
1585769508.0	W	31.13.69.0/24					100.0	benign

Table 5.3: Rostelecom (AS 12389) announcing a more specific prefix 31.13.69.0/24 of Facebook (AS 32934) where in contrast it usually announces 31.13.64.0/19

Time	Type	Announced prefix	AS path	Origin Criterion	Path Criterion	Neighbor Criterion	Score	Verdict
1585769568.0	A	95.100.200.0/24	39122 3356 20764 12389	0.0	0.0	0.0	0.0	New
1585769808.0	A	95.100.200.0/24	39122 174 3356 20764 12389	0.0	0.0	0.0	0.0	New
1585769838.0	A	95.100.200.0/24	39122 2914 3356 20764 12389	0.0	0.0	0.0	0.0	New
1585769868.0	W	95.100.200.0/24					100.0	benign

Table 5.4: Rostelecom (AS 12389) announcing a more specific prefix 95.100.200.0/24 of Akamai (AS 20940) where in contrast it usually announces 95.100.200.0/22

announcements were scored, where 17'588 were new to the AS, 7'411 were malicious, 101'141 were suspicious and 654'409 were labeled as benign.

For this case we look at the hijacks of Facebook (AS 32934) and Akamai (AS 20940). In table 5.3 we present the extracted announcements that hijacked the Facebook prefix 31.13.64.0/19 by announcing a more specific prefix 31.13.69.0/24. We can see how our system classifies the more specific announcements as new. This is caused by the fact that we have found a known prefix in the RIB that is a less specific one (31.13.64.0/19) but in that entry the originating AS did not match the origin of the hijacking announcements. Table 5.4 on the other hand shows the more specific announcements that hijacked Akamai (AS 20940). One of the prefixes that Akamai usually announces is 95.100.200.0/22, in the table instead we can see how Rostelecom started announcing 95.100.200.0/24 using its origin AS. Our system again classified these announcements as new like the ones from Facebook for the same reason, a less specific prefix is known but the origin AS does not validate.

5.2 Discussion

Going through the three case studies we have seen that our system does indeed manage to identify malicious and suspicious behaviour to a good extent. The system however is not perfect and unfortunately we were not able to test out the the cluster criterion on these case studies because of the computing capabilities. Another issue we noticed is that the system may be too sensitive in some cases by labeling too many announcements as suspicious, because of the lack of time we were not able to analyze in more detail the reasons why that was the case. By playing more with the parameters of the model one could potentially reduce the number of the false positives (classifying good announcements as suspicious or malicious) and build a better performing model. Additionally one may add more scoring criteria to get a performance boost, the model was made with modularity in mind and adding more to the model is fairly simple.

All in all I would say that that a good foundation was built with this system and it does a good job of protecting the AS from leaks and hijacks, even though it may be over protective at times.

Chapter 6

Outlook

In this semester thesis an attempt has been made to make BGP a bit more safer and looking at the different case studies we have managed to do that to a good extent. In this chapter we are going to talk about what future work that could be done to further develop the model and some issues that need to be addressed. Afterwards, we are going to conclude by reasoning about the future of BGP and why its security is of great importance.

6.1 What else can be done?

As we have seen in the evaluation, the model is by no means perfect and in my opinion has more potential if optimized further. In this section we are going to look at potential ways in which we can improve the model. First of all, there was little time left to play around with different parameters of the model and it would be interesting to see how the model response changes when we give different weights to the different criteria.

Another potential way to improve is by revisiting clustering, which unfortunately didn't work when operating with large enough ASes because of the amount of prefixes that were updated during the week, maybe by optimizing the code and/or using bigger hardware the issue could be resolved. In addition, other methods could be tried out by defining different states and distance metrics, in doing so we can hope that the clusters become finer.

Meanwhile the implementation of new criteria which can be used together with the existing ones to form more complex scoring functions, can help decrease the number of false positives and negatives there are. The possibilities are many and it would be interesting to find other clever criteria.

Finally, the needed runtime of the model to train and score is fairly long in cases where we have a big AS that receives lots of announcements. The model being written in Python and using the Pandas library and can be further optimized for speed by avoiding or replacing certain functions. Alternatively, it can be completely rewritten in a different language such as C which surely would improve the time performance.

6.2 The Future of BGP

The Border Gateway Protocol is most likely going to stay "the protocol" for inter-domain routing given that all the ISPs are using the compatible hardware for it and there is little incentive for them to invest into something when it already works. Even though as we have seen, security-wise there are great weaknesses in BGP.

With the Internet becoming more and more part of our daily lives and a driving force of the global economy, it is shocking to see how fragile the "backbone" of the Internet is and how little several ISPs are doing to protect it. In the future we are going to need a solution that can work in parallel with BGP and with minimal intrusion to the protocol itself, this way hopefully we can see a higher rate of adoption by the ISPs in comparison to the past implementations such as RPKI.

BGP security is not an easy task as I have witnessed during my thesis and in my opinion requires more attention in the future as the number of hijacks and leaks increase throughout the years, and as we become more dependent of it, the more important it is that we have a strong and secure infrastructure.

Chapter 7

Summary

In this thesis we have introduced and developed an alternative approach to BGP security that requires minimal intervention to the currently existing setup. Using past BGP data we managed to train our model and then show that it indeed protects an AS from leaks and hijacks by observing three famous incidents. Furthermore, we refreshed our knowledge of BGP and studied the statistical nature of the announcements by analyzing their behavior and the patterns they follow. We then attempted to cluster the announcements with some success but unfortunately could not use it in our model given the sheer size of the prefixes that were present in the training data.

The model proposed here has still a long way to go until it can be truly implemented in a real BGP environment. Yet, what we hope to have achieved is to have built a certain foundation for possible future works and to have given some new ideas on how we could potentially tackle the ever-present issue of BGP security. An issue that needed to be fixed a long time ago and that hopefully gains more awareness as the Internet becomes an even more integral part of our lives.

Bibliography

[1] Balakrishnan, H. How youtube was "hijacked". *Massachusetts Institute of Technology Department of Electrical Engineering and Computer Science* (2009).

[2] CAIDA. Bgpstream. Available at `https://bgpstream.caida.org/docs/api/pybgpstream`.

[3] Cloudflare. What is bgp hijacking? Available at `https://www.cloudflare.com/learning/security/glossary/bgp-hijacking/`.

[4] Gilad, Y., Hlavacek, T., Herzberg, A., Schapira, M., and Shulman, H. Perfect is the enemy of good: Setting realistic goals for bgp security. *HotNets-XVII, November 15–16, 2018, Redmond, WA, USA*.

[5] IETF. Bgpsec protocol specification. Available at `https://tools.ietf.org/html/rfc8205` (September 2017).

[6] IETF. A border gateway protocol 4 (bgp-4). Available at `https://tools.ietf.org/html/rfc4271` (January 2006).

[7] IETF. An infrastructure to support secure internet routing. Available at `https://tools.ietf.org/html/rfc6480` (February 2012).

[8] IETF. Problem definition and classification of bgp route leaks. Available at `https://tools.ietf.org/html/rfc7908` (June 2016).

[9] Madory, D. Large bgp leak by google disrupts internet in japan. Available at `https://dyn.com/blog/large-bgp-leak-by-google-disrupts-internet-in-japan/`.

[10] Madory, D. Large european routing leak sends traffic through china telecom. Available at `https://blog.apnic.net/2019/06/07/large-european-routing-leak-sends-traffic-through-china-telecom/` (June 2019).

[11] Noction. Bgp security: an overview of the rpki framework. Available at `https://www.noction.com/blog/rpki-overview` (March 2019).

[12] Noction. Bgpsec protocol specification. Available at `https://www.noction.com/blog/bgpsec-protocol` (April 2015).

[13] of Oregon, U. Route views. Available at `http://www.routeviews.org/routeviews/`.

[14] Systems, C. Bgpmon. Available at `https://bgpstream.com/`.

[15] ZDNet. Bgpsec protocol specification. Available at `https://www.zdnet.com/article/russian-telco-hijacks-internet-traffic-for-google-aws-cloudflare-and-others/` (April 2015).