Detecting BGP Hijacks on the Internet

AARUSHI GUPTA (AG2288)*, AKALYA ASOKAN (AA2442)*, and SADMAN CHOWDHURY (SKC86)*, Master of Engineering, CS, Cornell University, USA

Link to the project : *https://github.com/AkalyaAsokan/BGP-Hijacks*

In this project, we investigate various BGP hijacks, including MOAS, sub-MOAS, fake-path, and Defcon#16 hijacks. By capturing BGP route announcements and analyzing the routing information, we can identify and track cases of prefix hijacking. By analyzing real-world BGP data, we demonstrate the prevalence and impact of these hijacks on the routing system. This report focuses on our implementations to identify the attacks and our findings. Overall, this paper highlights the importance of understanding and addressing the vulnerabilities in the BGP system to ensure the stability and security of the Internet.

CCS Concepts: • **Networks** → **Routing protocols**.

Additional Key Words and Phrases: BGP, MoAS, sub-MoAS, Fake-Path, Defcon#16

1 INTRODUCTION

Internet routing is a complex and critical system that enables communication between millions of devices worldwide. Understanding the behavior of the routing system is important because it is inherently unlimited. This means that there is a concern that if every network were to announce only the most specific prefixes or use unstable routing configurations, it could result in an overwhelming number of routing updates and cause the routing system to rapidly switch between unstable states, which would create a significant problem for BGP [3].

1.1 Border Gateway Protocol (BGP)

BGP is the underlying protocol that enables the exchange of routing information between different Autonomous Systems (ASes) on the Internet in order to facilitate the forwarding of packets between them. BGP is primarily used to connect large-scale networks, such as Internet Service Providers (ISPs), and it plays a critical role in ensuring the stability and reliability of the Internet. It uses a routing algorithm known as the Bellman-Ford distance vector. It operates by exchanging routing information between neighboring routers using a TCP-based protocol.

1.2 BGP Hijacks

BGP lacks authentication mechanisms and is vulnerable to frequent misconfigurations and attacks, often resulting in hijacks and significant impacts on network performance and security [1]. BGP hijacking is a type of cyberattack that involves diverting Internet traffic away from its intended destination by manipulating the Border Gateway Protocol. Here, an attacker sends false BGP routing information to the Internet's backbone routers, causing them to redirect traffic to a different network or destination than originally intended. BGP hijacking is a serious threat to the Internet infrastructure, as it can potentially affect large numbers of users and cause significant economic damage. The most prevalent and successful forms of BGP attacks involve prefix and sub-prefix hijacking, whereby an unauthorized or improperly configured autonomous system (AS) announces an IP address block that it is not authorized to use [7].

*All authors contributed equally to this research.

Authors' address: Aarushi Gupta (ag2288); Akalya Asokan (aa2442); Sadman Chowdhury (skc86), Master of Engineering, CS, Cornell University, Ithaca, New York, USA, 14850.

2 TRAINING DATASET

In this section, we discuss the collection and preprocessing process of our training data for detecting BGP hijacks.

2.1 Data Collection

For our research on BGP data analysis, we utilized the BGPStream tool, a tool designed to analyze both live and historical BGP data. To enable us to write code in Python, we made use of the PyBGPStream Python package, which offers bindings to the underlying libBGPStream library written in C [5].

BGPStream route collectors are distributed systems that collect and store Border Gateway Protocol (BGP) updates and provide them for analysis [8]. We collected data from three collectors: "route-views.sydney", "route-views.sg", and "route-views2.routeviews.org", from June 30 to July 7, 2017. We chose these collectors because they cover a wide geographical area and have a high degree of completeness and stability. We collected updates and did not set a filter on the prefixes to maximize the amount of collected data. This resulted in a training dataset of 19.98GB.

2.2 Pre-processing

As part of our data analysis process, we conducted some preprocessing on the collected BGP data to enhance its suitability for analysis. One of the key steps in this process was to extract and store the AS paths associated with each prefix in individual files, which enabled more efficient and effective analysis of the data. An example of the format of the AS paths is shown below:
Prefix: IP/Mask
Total AS paths: X
AS path: A -> B -> C
AS path: D -> E -> F -> C

2.3 Visualization

In order to gain insights into the collected BGP data, we employed the NetworkX library to plot the graph of all connected Autonomous System Numbers (ASNs) (refer to Fig. 1). Additionally, we leveraged the RIPE Stat API to obtain supplementary information on autonomous systems. Our analysis revealed a notable observation regarding AS prepending, a technique used to manipulate routing decisions, where an AS repeatedly adds its own ASN to the AS path. We generated a visualization to identify the AS that employs this technique the most frequently (refer to Fig. 1).

We observed that Public Officials Benifits Association in Korea was prepending a lot. One possible speculation is that POBA and other Korean pension funds may be using AS prepending as a means of optimizing the performance of their network infrastructure and reducing potential network downtime or disruptions, which could have negative financial implications, in order to generate the necessary returns to meet their retirement income commitments [4].

In summary, we conducted data collection and preprocessing of BGP data from multiple collectors and utilized visualization techniques such as NetworkX and RIPE Stat to extract insights. Our collected training data will be instrumental in developing effective tools for detecting BGP hijacking.

3 TESTING DATASET

In our experimentation, we opted to use a fixed number of records as our testing data instead of time duration. We chose a dataset of 5000 records that were collected after the trading data. This decision was made due to the large amount of data collected with the parameters used, where 3000 records were captured in just 1 minute and 16 seconds in one instance. As a result, we capped the

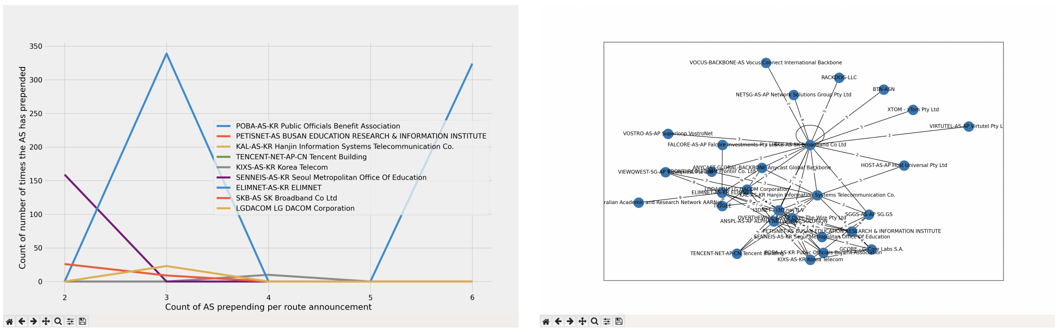

Fig. 1. The image on the left shows the count of AS prepending vs no of AS paths where an AS has prepended in the training data collected. Image on the right is a visualization of a small connected network of ASes and the number of hops between them

data at 5000 records to enable us to observe and comprehend the patterns and their associated attacks effectively.

4 TECHNICAL APPROACH

We implemented detectors for four different types of attacks: two origin hijacks (Maximum Origin Autonomous System (MOAS) and subMOAS attacks), a fake path hijack, and a Defcon#16 hijack. The algorithms, data structures, and assumptions we used for each of these detectors are mentioned below. We seperated the logic for each detector and optimized each detector's code by selecting the most efficient data structures for its specific needs. This design choice allows for easy integration and use of individual detectors without any interdependencies.

4.1 Origin Hijack #1: MoAS

A MOAS attack takes place when a prefix seems to originate from multiple Autonomous Systems (ASes). Suppose prefix d is associated with as-paths as1= $(x_1, x_2, x_3, ..., x_m)$ and as2 = $(y_1, y_2, y_3, ..., y_n)$, then, MoAS attack occurs if $x_m \neq y_n$ [10]. In order to detect MOAS attacks, we developed a method

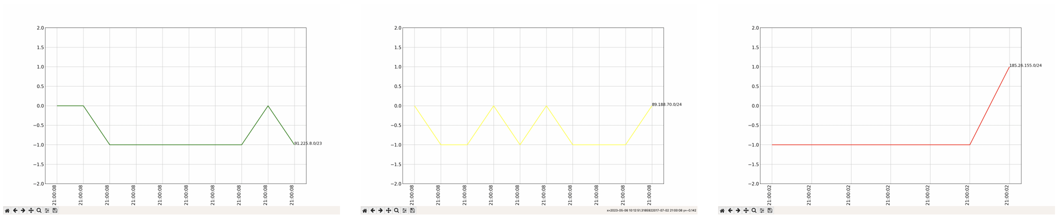

Fig. 2. These refer to screenshots of animations when running the detectors. They comprise of three types of colored lines. Green, with a value of 1 indicates that there is no hijack (the left image). Yellow, value -1 signifies that the prefix from the test data does not exist in the training data (the middle image). Lastly, red, value 1 indicates an instance of a hijack (the right image)

that involved creating a dictionary that mapped prefixes to their respective origin ASNs using training data. The ground truth ASN for each prefix was determined using a heuristic based on the most frequently occurring ASN in the collected paths. We then tested this approach on testing data, where we compared the origin ASN of each prefix to the ground truth ASN. If there was a mismatch, we classified it as a MOAS attack. Overall, this approach allowed us to quickly and efficiently identify MOAS attacks by comparing the origin ASN for each prefix to a predetermined ground truth value.

4.2 Origin Hijack #2: Sub-MoAS

A MOAS conflict happens only for a specific prefix "p". When there is another prefix "q" that is more specific than "p" and has a different origin than "p", it is referred to as a MOAS sub prefix or sub-MOAS [2].

In order to identify subMOAS attacks, we created a nested dictionary structure from the training data, which included the prefix and origin ASN for each path. The structure was designed to handle IP as well as the Mask separately and the format of this structure is as follows:

{IP1: { MASK1: ASN1, MASK2: ASN2, MASK1: ASN3, MASK1: ASN3}, IP2: { ... },...}

From this structure, we extracted the most common ASN for each IP and mask combination and created a new nested dictionary that mapped the IP to its corresponding ASN values for each mask, like this

{IP1: { MASK1: ASN1, MASK2: ASN2}, IP2: { ... },...}

To identify subMOAS attacks, we utilized a two-fold checking method when encountering a new IP in the testing data. We compared the mask value of the new IP to the maximum mask value recorded for that IP in the training data. If the mask value of the new IP was greater than the recorded maximum, it was considered a possible subMOAS attack. In addition, we also verified if the origin ASN for the new IP did not match the corresponding origin ASN value in the training data. Only if both conditions were met, we classified the event as a subMOAS attack. This approach allowed us to effectively detect and prevent subMOAS attacks.

4.3 Fake-Path Hijack

In our approach to detecting fake path hijacks, we utilized two key data structures. Firstly, we created a dictionary that mapped prefixes to their actual origin ASNs, similar to our approach for detecting MOAS attacks. Secondly, we constructed a connected graph for each prefix, in which each node represented an ASN in the path, and each edge linked adjacent ASNs.

To identify potential fake path hijacks, we checked for two conditions. First, if the AS path in the test data was not present in the corresponding graph for that prefix, we marked it as a possible hijack. Second, if the origin prefix in the test data matched the actual origin ASN for the prefix in the dictionary, it was considered a valid attack.

4.4 Defcon#16 Hijack

DEFCON 16 Hijack was a demonstration of a BGP hijacking attack conducted by security researchers at the DEFCON 16 conference in 2008. In the attack, the researchers were able to hijack the BGP route of a range of IP addresses belonging to a major financial services company, redirecting traffic intended for those addresses to a rogue network under their control [6][9].

To detect Defcon attacks, a comprehensive approach was required that utilized all three data structures, i.e., the dictionary for MOAS attacks, the nested dictionary for subMOAS attacks, and the connected graph for fake path hijacks. We then employed three conditions to detect Defcon attacks, including the matching of origin prefixes, the non-existence of the AS path in the connected graph built from training data, and the comparison of the mask length of the test data prefix to the ground truth. By combining these three methods, we were able to effectively detect Defcon attacks, thereby safeguarding against this advanced type of routing attack.

5 EXPERIMENTAL RESULTS

The experimental results indicate that the MOAS attack was more frequently detected than the subMOAS attack, as expected. Defcon #16 had the least number of attacks, which is consistent with the fact that it is the most sophisticated type of attack among the four. This finding is depicted in Figure 3, top left. However, the fake hijack attack produced unexpected results. Although identifying fake path hijacks is a greater challenge than other types of origin hijack attacks, we have still achieved significant success using our approach. This can be attributed to the algorithm and the small dataset used in the study. It should be noted that our current implementation utilizes only a few days' worth of training data, which increases the likelihood of identifying false positives. However, we have yet to explore ways of addressing this issue in our code. Also the algorithm only considers paths that have previously been observed to be valid, so any new path that differs could potentially be flagged as a hijack.

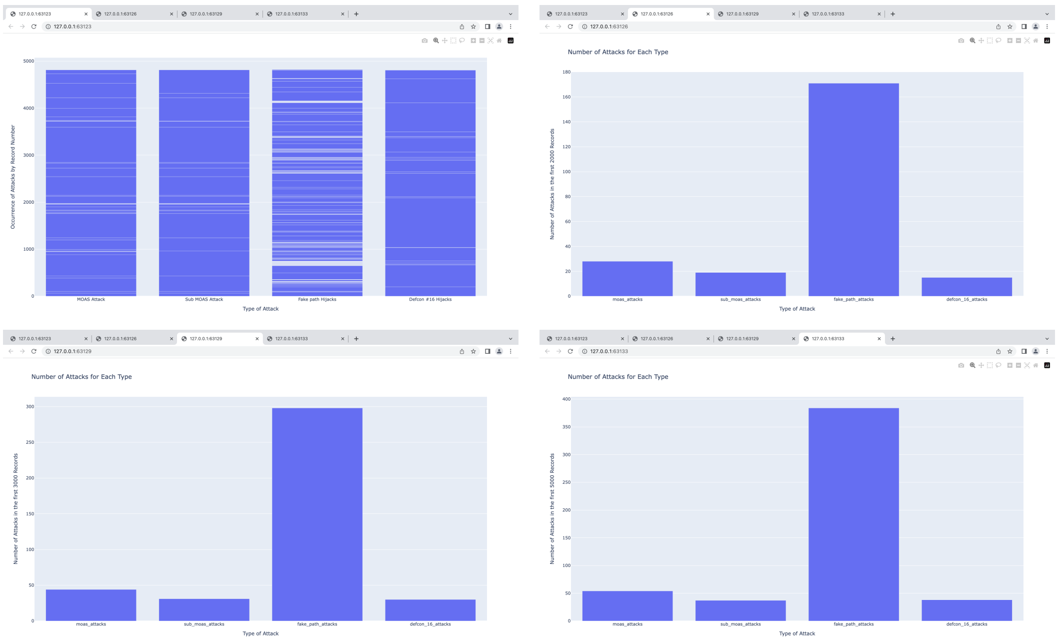

Fig. 3. The top left figure shows the record number at which an attack was found in each of the four types of attacks. The top right shows the total number of attacks detected for each type at the end of 2000 testing data records. Bottom left shows for 3000 records and the bottom right shows 5000 records.

It is important to note that the current implementation utilizes only a few days' worth of training data, which increases the likelihood of identifying false positives. Moreover, the number of attacks observed aligns with the expectation based on the number of records tested. For instance, there were approximately 30 MOAS attacks in the first 2000 records, while only around 20 subMOAS and 15 Defcon #16 attacks were detected, as expected. However, the study observed a significant number of fake path hijacks, with approximately 180 instances recorded, as depicted in Figure 3, top right. On increasing the number of testing data to 3000 records, the study observed a little less than 50, 30, and 25 attacks for MOAS, subMOAS, and Defcon #16, respectively. The number of fake path hijacks almost doubled to 300 this time, as depicted in Figure 3, bottom left. When the number of records was further increased to 5000, the study observed a little more than 50, around 40, and around 40 attacks for MOAS, subMOAS, and Defcon #16 attacks, respectively. Additionally, the number of fake path hijacks recorded increased to 400, as shown in Figure 3, bottom right.

It has been identified that a substantial proportion of the attacks were originated from the network of AMX Argentina S.A. However, it is noteworthy that the highest number of false path hijack incidents were traced back to TATA COMMUNICATIONS (AMERICA) INC. Nevertheless, these findings must be interpreted with caution as fake path hijacks are not always indicative of accurate information.

ACKNOWLEDGMENTS

To Professor Rachee Singh, for her unwavering support, guidance, and mentorship throughout this project. Your insights, feedback, and expertise have been invaluable in shaping the direction and scope of this project. We would also like to express our sincere gratitude to all the Teaching Assistants (TAs) who provided their unwavering support throughout our project.

REFERENCES

[1] Shinyoung Cho, Romain Fontugne, Kenjiro Cho, Alberto Dainotti, and Phillipa Gill. 2019. BGP hijacking classification. In *2019 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 25–32.

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

[3] Geoff Huston. 2018. BGP in 2017.

[4] Dong Hun Jang. 2020. *Providing a Model for other Korean Institutional Investors.* https://www.iinetworks.com/topic/dong-hun-jang-cio-koreas-public-officials-benefit-association-man-mission

[5] Roman Khavrona. 2022. *Analysing Internet route changes related to the Russia-Ukraine war using BGP historical data.* B.S. thesis. University of Twente.

[6] Fouad Kiamilev and Ryan Hoover. [n. d.]. Defcon 16, 2008. *Demonstration of Hardware Trojans* ([n. d.]).

[7] Reynaldo Morillo, Justin Furuness, Cameron Morris, James Breslin, Amir Herzberg, and Bing Wang. 2021. ROV++: Improved Deployable Defense against BGP Hijacking.. In *NDSS*.

[8] Chiara Orsini, Alistair King, Danilo Giordano, Vasileios Giotsas, and Alberto Dainotti. 2016. BGPStream: a software framework for live and historical BGP data analysis. In *Proceedings of the 2016 Internet Measurement Conference*. 429–444.

[9] Denis Richard and Ming Chow. [n. d.]. The Inherent Vulnerability of the Border Gateway Protocol (BGP). ([n. d.]).

[10] Xiaoliang Zhao, Dan Pei, Lan Wang, Dan Massey, Allison Mankin, S Felix Wu, and Lixia Zhang. 2001. An analysis of BGP multiple origin AS (MOAS) conflicts. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. 31–35.

A LIMITATIONS

In addition to successful data collection and preprocessing, we encountered some challenges and obstacles along the way.
Slow data collection: We attempted to collect data from 2023 but encountered significant delays due to slow data collection. One possible reason for this issue could be the increased demand for BGP data or technical issues with the collectors.
Multiple attempts: We made over 25 different attempts to collect training data, each with different parameters and collectors. Some of the training data were too large, making them difficult to process, while others were too small to provide meaningful insights. Additionally, some collectors were inactive during the time period we were attempting to collect data, which further complicated the data collection process.

In conclusion, despite encountering challenges during the data collection process, we were ultimately successful in obtaining high-quality training data for detecting BGP hijacking. The difficulties we faced serve as a reminder of the importance of developing robust and scalable data collection methods to ensure the timely and accurate detection of BGP hijacking.