THM | CAMPUS FRIEDBERG | IEM Informationstechnik-Elektrotechnik-Mechatronik

TECHNISCHE HOCHSCHULE MITTELHESSEN

MASTER THESIS

PROTECTION OF THE ENTERPRISE NETWORK AGAINST BGP HIJACKING

to obtain the academic degree

Master of Science (M.Sc.)

submitted to the

Fachbereich Informationstechnik - Elektrotechnik – Mechatronik der Technischen Hochschule Mittelhessen (THM)

October 2021

submitted by:
Tobias Pipp

maintained by:
Prof. Dr. Dieter Baums
(Technische Hochschule Mittelhessen)

and

Dr.-Ing. Matthäus Wander
(Bundesamt für Sicherheit in der Informationstechnik)

Abstract

The BGP protocol is the most widespread protocol for inter-domain communication and thus forms the backbone of worldwide Internet communication. The biggest advantage is at the same time the biggest disadvantage. The assumption that each message is correct, the trust model of BGP, allows easy connection between different providers. At the same time, it allows attackers to carry out large-scale attacks very easily on internet communication. In this paper, a general detection model for BGP hijacking will be designed. Furthermore, we will investigate how different BGP hijacking detection software works. There will also be a collection of the current possibilities for prevention, reaction and analysis of BGP hijacking. In the practical part, BGP long-term data and RPKI will be analysed on real examples.

As a result of this work, a BGP hijacking classification is created. Different possibilities for prevention, reaction and analysis were explained and evaluated. This information helps to answer the question of how emergency plans for an enterprise network can look like. Advantages and disadvantages of the BGP hijacking detection software were identified and presented. Thus, important points of the functioning of software like BGPalerter or Artemis can be explained. In the practical part, long-term data was successfully used to detect BGP attacks retrospectively. Furthermore, the usefulness of RPKI was demonstrated by means of a real attack.

Keywords: BGP (Border Gateway Protocol), BGP hijacking classification, BGP long term data, Artemis, BGPalerter

Kurzfassung

Das BGP Protokoll ist das meistverbreitete Protokoll für die Inter-Domain Kommunikation und bildet somit das Rückgrat der weltweiten Internetkommunikation. Der größte Vorteil ist gleichzeitig der größte Nachteil. Das Vertrauen, dass jede Nachricht korrekt ist, ermöglicht ein einfaches Peering zwischen verschiedenen Providern. Gleichzeit ermöglicht es Angreifern, auf sehr einfache Weise, weltweit großflächige Angriffe auf die Internetkommunikation durchzuführen. In dieser Masterthesis soll ein generelles Definitionsmodel für BGP Hijacking erstellt werden. Weiter soll untersucht werden wie verschiedene BGP Hijacking Erkennungssoftware arbeiten. Darüber hinaus wird es eine Sammlung der aktuellen

Möglichkeiten für Vorbeugung, Reaktion und Analyse von BGP Hijacking geben. Im praktischen Teil sollen BGP Langzeitdaten und RPKI an Realbeispielen analysiert werden.

Als Ergebnis dieser Arbeit konnte ein BGP Hijacking Klassifikation erstellt werden. Es wurden verschieden Möglichkeiten zur Prävention, Reaktion und Analyse erläutert und bewertet. Diese Informationen helfen, die Frage zu beantworten, wie Notfallpläne für ein Unternehmensnetzwerk aussehen können. Bei dem BGP Hijacking Erkennungssoftwaren konnten Vor- und Nachteile identifiziert und dargestellt werden. So können wichtige Punkte der Funktionsweise von solche einer Software wie BGPalerter oder Artemis erläutert werden. Im praktischen Teil konnten erfolgreich Langzeit-Daten genutzt werden und BGP Angriffe nachträglich zu erkennen. Weiter konnten die Nützlichkeit von RPKI anhand eines echten Angriffes nachgewiesen werden.

Acknowledgments

I am writing my Master's thesis as part of my Communication Engineering degree programme at the THM (Technischen Hochschule Mittelhessen) Friedberg. The thesis will take place at the Federal Office for Information Security (BSI) in Bonn/Germany.

My thanks go to Prof. Dr. Dieter Baums from the THM for his supervision and support. Special thanks go to Dr.-Ing. Matthäus Wander for the technical supervision on the part of the BSI.

I would also like to thank Vasileios Kotronis from the Artemis project, who always had time to support me in my research and answer my questions.

Declaration of authorship

I hereby certify that I have prepared this thesis without the unauthorised assistance of third parties and without the use of any other resources except those indicated. The ideas taken directly or indirectly from external sources are marked as such.

Bonn, the 30th October 2021

Table of contents

List of figures

List of tables

List of abbreviations

Artemis	Automatic and Real-Time dEtection and Mitigation System
AS	Autonomous system
BGP	Border Gateway Protocol
ERP	Exterior Routing Protocols
HEAP	Hijacking Event Analysis Program
IANA	Internet Assigned Numbers Authority
IGP	Interior Gateway Protocol
IRP	Interior Routing Protocol
ISPs	Internet Service Providers
MitM	Man-in-the-Middle
MOAS	Multiple origin AS
OSI	Open Systems Interconnection Reference Model
RA	RIPE Atlas
RC	Route collectors
RCC	Remote route collector
RIB	Routing information base
RIPE NCC	Réseaux IP Européens Network Coordination Centre
RIR	Regional Internet Registry
RIS	Routing Information Service (from RIPE NCC)
ROA	Route Origin Authorisations
ROV	Route Origin Validation

RTR	Resource Public Key Infrastructure (RPKI) to Router Protocol
SaaS	Software as a service

1 Introduction

The Internet is one of the most important resources in the world today. Almost everyone is directly or indirectly dependent on its functioning. Thus, a trouble-free operation is important for people, companies and governments. The Border Gateway Protocol (BGP) is the backbone of global communications and is indispensable today. BGP is used to exchange routing information with each other. A symbolic example of this. We live in Frankfurt and have the telephone book from Frankfurt. So we can call anyone from Frankfurt. But if we want to reach someone from Berlin, we need the phone book/ right telephone line there, and also the other way round. In the same sense, BGP provides for the exchange of IP address routing information. So that everyone knows where and how someone can be reached. It is therefore the most important protocol in the field of Exterior Routing Protocols (ERP). ERP is a collective term for the routing protocols used between different large WAN (wide area network) like the networks connections between different ISPs (Internet Service Providers).

Although BGP is unknown to most people, they all rely on it. The biggest advantage is at the same time the biggest disadvantage. Since BGP does not require a complex validation process in its basic function, many different ISPs can exchange data with each other at a very low threshold. However, this protocol is also very easy to attack by attackers. Not in every case is an attack intentional but can be traced back to a misconfiguration. If attackers act unhindered, they can very strongly influence the worldwide Internet traffic.

Current examples are the hijacking incident of AS212416 on 29.07.2021 [1], which disrupted services from Telkom, among others. But also the BGP misconfiguration of Facebook on 4.10.2021 [2] shows how important BGP and its correct functioning is. Facebook had a complete communication breakdown for about 6 hours due to an error in the BGP announcement. The malicious attack of BGP traffic is called BGP hijacking.

It is therefore important to ask what possibilities there are to counter this threat. The goal of this thesis is to develop a standard model for BGP hijacking classification. Furthermore, possibilities to protect the network (prevention, reaction and analysis) will be investigated. Special attention will be paid to open-source detection software. What database they use and how the BGP hijacking decisions work within the software. As a practical goal, the possibilities of the BGP hijacking software are to be demonstrated. For a given event, an RPKI will also be analysed on an incident.

For this purpose, firstly an introduction to the basics (chapter 2) will be given, then a concept for BGP hijacking class defection will be developed (chapter 3). So various sources used to collect recent findings, and their findings brought together. The following chapter (4) will deal with prevention, reaction and damage assessment (analysis) in case of BGP hijacking. Various possible improvements are presented for the BGP system. In chapter 5 different software for the detection of BGP hijacking will be evaluated. A decision matrix is constructed, and the various advantages and disadvantages of the tools are discussed. Afterwards (chapter 6), historic BGP data will be extracted from public databases using BGPStream and analysed for anomalies using Artemis. Followed (chapter 6.5) by a practical analysis of RPKI. A summary and conclusion build the final part of the thesis in chapter 7.

2 Theoretical Background

In this chapter, BGP is to be placed in the world of routing protocols. In the further course of this work, it will only be dealt with the BGP protocol. The other protocols addressed in the chapter are intended to help classify BGP in the world of routing protocols.

2.1 Basic Information about Routing

Routing protocols are the key for the internet as we know it today. Only with routing protocols is it possible to keep a constantly changing network up to date. The IP networks use IP addresses to determine the destination for every data packet. Every packet has a source and destination address. A router uses the destination address and compares this information with its routing table. Then the router can decide where to forward the packet so that it reaches its destination.

Routers can communicate using custom protocols to determine the routing of messages. The routers exchange routing information with each other. In this way, each router builds its own routing table. According to this table, the router then decides how to deal with each IP packet. Depending on the protocol, different metrics can be transmitted, such as bandwidth, load, delay, reliability, or hops. Routing entries divided into static and dynamic. Both types of information can be transmitted using the routing protocol. Furthermore, dynamic routes can be influenced by metric changes during operation. This gives a router the opportunity to optimise the connection path as best as possible based on the metrics and the configuration specifications.

Routing protocols are basically divided into two areas of application. The Interior Routing Protocol (IRP)/ intra-domain routing is used within an AS (autonomous system). Communication between different ASs is ensured by the Exterior Routing Protocols (ERP)/ inter-domain routing. In the RFC 1983, AS is defined as follows: “A collection of routers under a single administrative authority using a common Interior Gateway Protocol for routing packets.”

There are two basic routing protocol algorithms. Distance vector algorithm and link state. The distance vector algorithm is based on the Bellman Ford algorithm. Link State uses the Dijkstra algorithm for its calculation. Distance vector algorithm are used in large networks. Here, only the neighbour is communicated with. Therefore the convergence is slow. With link state algorithm, one device knows all the others. Convergence is fast. Convergence in this case describes how fast all routers in a network are up to date. Thus, a fast convergence is useful because the routers decide faster on the routing based on the newest information. Distance vector algorithms require less computation than the Dijkstra algorithm. This is because with distance vector algorithms routers only communicate with their neighbours. In the other algorithm, routing information is sent from the router to all other routers. A good explanation about routing protocols algorithm can be found in the following literature [3, pp. 67-68].

The most popular IRPs are listed below:

- Distance vector algorithm
    - Gateway to Gateway Protocol (GGP)
    - Routing Information Protocol (RIP)
    - Interior Gateway Routing Protocol (IGRP)
    - Enhanced Interior Gateway Routing Protocol (EIGRP)
- Link state algorithm
    - Open Shortest Path First (OSPF)
    - Intermediate System - Intermediate System (IS-IS)

The most popular ERPs are listed below (all use distance vector algorithm):

- Exterior Gateway Protocol (EGP)
- Border Gateway Protocol (BGP)
- Interdomain Routing Protocol (IDRP)

The most commonly used ERP for global communication is BGP. The main reason why this protocol is primarily used by the ERPs is that for worldwide communication, everyone has to use the same protocol. BGP has prevailed in this respect. However, this does not mean that it is the best protocol. There is the problem that a large number of equal partners communicate with each other, making it very complex to switch to a new protocol at the same time. Therefore, adaptations to the protocol are only possible to the extent that the basic function is not disturbed. BGP was developed as a replacement for EGP, as EGP does not support multipath networks [4]. IDRP is based on BGP and works together with the End System to Intermediate System protocol (ES-IS) and the Intermediate System to Intermediate System protocol (IS-IS) [5]. BGP is strictly speaking a path vector protocol. Path vector protocols belong to the group of distance vector protocols. As the name suggests, path vector protocols forward path information. So the router does not only receive a distance vector (costs and distance factors) from its neighbour router. It also receives the path as a sequence of AS numbers.

2.2 BGP

The Border Gateway Protocol (BGP) is the most common protocol for routing information exchange between ASs. BGP is a path vector protocol. Path-vector protocols are used between different networks and consider the networks as ASs. BGP can make routing decisions based on metrics sent via the messages.

With the help of the BGP protocol, an AS can determine the complete connection path between the various ASs. The communication of the BGP systems runs via TCP on port 179. The information used there comes from the network administrators of the individual ASs or is determined automatically from the router. With this information, each router builds its own database for data exchange with the known BGP routers. [6]

The BGP routers only establish a connection with the neighbours, but always transmit the complete path information between them and the destination AS for a prefix. This means that no loops can occur, as a BGP router discards a packet if its own AS number is already contained in the AS path of a message.

The BGP protocol is defined by various RFCs (Table 1) by the IETF. The most important ones are listed below.

Table 1 RFC description for the BGP

RFC No.	Date	Title/Content
1105	June 1989	Border Gateway Protocol (BGP)
1163	June 1990	Border Gateway Protocol (BGP); replacement for RFC 1163
1267	October 1991	BGP in version three (BGP-3)
1771	March 1995	BGP in version four (BGP-4)
4271	January 2006	BGP in version four (BGP-4); replacement for RFC 1771
4364	February 2006	BGP/MPLS IP Virtual Private Networks
4760	January 2007	Multiprotocol Extensions for BGP-4; replacement for RFC 2858
8092	February 2017	BGP Large Communities attribute at BGP 4

2.2.1 Explanation of BGP communication process

The basic functioning of BGP can be described as follows. Each participant has its own unique AS number. It can receive and send BGP announcements. Each message is assumed to be true. If a prefix is possessed, it can be announced by means of its AS. The connected BGP routers then forward the information to the next BGP router and adjust their own routing tables based on the new information. If an IP data packet arrive to an AS, the AS compares the IP address with its routing table and if it found a matching prefix, it is forwarded to the correct next AS. This blind trust results in a security gap that is exploited in BGP hijacking. During an attack, the BGP message is modified to change the routes in the interest of the attacker.

Under certain circumstances, this false information can be adopted by BGP routers worldwide. If no protective mechanisms have been put in place by the ISP beforehand and the necessary worldwide connections are in place. Despite this vulnerability, BGP is used because it can be used without major hurdles. A peering connection is enough for data exchange. Peering is defined as the exchange of data between computer networks on an equal footing.

2.2.2 BGP message types

Following is the BGP message header:

```
0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    +                                                               +
    |                                                               |
    +                                                               +
    |                          Marker                               |
    +                                                               +
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         Length               |      Type     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The “Marker” field is set to one. The “Type” field can have four different values:

1 - Open

2 - Update

3 - Notification

4 - Keepalive

5 - Route-refresh (RFC 2918) (no further explanation here)

BGP uses four different messages (definition and more detailed explanation in RFC 4271):

- Open

After an established TCP connection, the BGP session start with an Open Message.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+
|    Version    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     My Autonomous System      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Hold Time           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         BGP Identifier                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Opt Parm Len  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|             Optional Parameters (variable)                    |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Update

This message is used to exchange routing information between two BGP routers.

```
+-----------------------------------------------------+
|   Withdrawn Routes Length (2 octets)                |
+-----------------------------------------------------+
|   Withdrawn Routes (variable)                       |
+-----------------------------------------------------+
|   Total Path Attribute Length (2 octets)            |
+-----------------------------------------------------+
|   Path Attributes (variable)                        |
+-----------------------------------------------------+
|   Network Layer Reachability Information (variable) |
+-----------------------------------------------------+
```

- Notification

In the event of an error, a notification message is sent before the connection is immediately terminated.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Error code    | Error subcode |   Data (variable)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Keepalive

The Keepalive message is not mandatory. Only if the hold timer is used in the opposite position a message must be sent before the timer expires. The TCP packet only contains the BGP header with the type value (=4) for Keep alive as a message. No other information is contained in the message.

2.2.3 BGP routing decision

The most important message for route exchange is the update message. This contains information about AS-path, next hop, IGP-metrics, communities (explanation in chapter 2.3.6), Origin, etc [7]. This contains all the information according to which the BGP router determines the priority of individual routes. However, the following rules are standard [8, p. 3]:

- A more precise prefix is preferred, regardless of the path length.
- If the prefix length is the same, the shorter path wins.
- Metrics (e.g. bandwidth) or communities can also influence the path selection. However, this is usually dependent on the provider. The following source [9] provides a detailed breakdown of how Cisco weights the routing decision with the different values.

The different values can partly be influenced by the administrator of a BGP router. These can be useful for BGP hijacking, among other things. One technique is called de-aggregation. Here, a more precise prefix is announced than the one used by the attacker. This has the consequence that the corresponding traffic for this prefix send to the own router; even if the path is longer.

MOAS stands for "Multiple origin AS" and describes the circumstance that an IP prefix or an AS number occurs more than once in the Internet. This case does not generate any errors in the BGP router. By default, the router would prefer the shorter route for the same prefix. A description of how MOAS is used can be found in Chapter 4.1.

2.2.4 Data and control plane definition for this thesis

In this thesis, a BGP hijacking classification is created based on the data and control plane. The following definition is used for this part of the thesis. The control level is the BGP protocol. The data level is then about the routed data packets (e.g., user data), e.g., via UDP or TCP.

2.3 Security for BGP

There have been repeated attempts to eliminate the known weaknesses of BGP. The following attack vectors are available here:

- BGP (more in chapter 3)
- Protocol under the BGP-Layer; like TCP-, MAC- and Physical layer
- Hardware (e.g. BGP router)

This chapter deals with possible solutions to this problem. Solutions are offered by protocols/approaches such as S-BGP (Secure BGP), soBGP (Secure Origin BGP), psBGP (Pretty Secure BGP), RPKI (Resource Public Key Infrastructure) or BGPsec (Border Gateway Protocol Security). In the following, RPKI, BGPsec, S-BGP and so BGP will be discussed. The more reliable techniques are RPKI and BGPsec.

S-BGP and soBGP are protocols that build on BGP and are intended to bring more security. Nevertheless, they are rather of a theoretic nature, as they have no relevance in the worldwide BGP data traffic. The biggest hurdle here is the necessary investment in hardware and know-how. The high costs are mainly caused by the fact that the current hardware does not support the computing effort and would have to be replaced. This also applies to BGPsec. With RPKI, only additional hardware has to be implemented, but it avoids having to buy completely new routers (see explanation in chapter 2.3.3).

2.3.1 S-BGP

Here is a brief explanation of the difference between S-BGP and soBGP. S-BGP follows a hierarchical certificate structure. This structure is also called onion style and is visible in Figure 1. During the verification process, the certificates are checked one after the other, like the layers of an onion. This requires a very high computing effort.

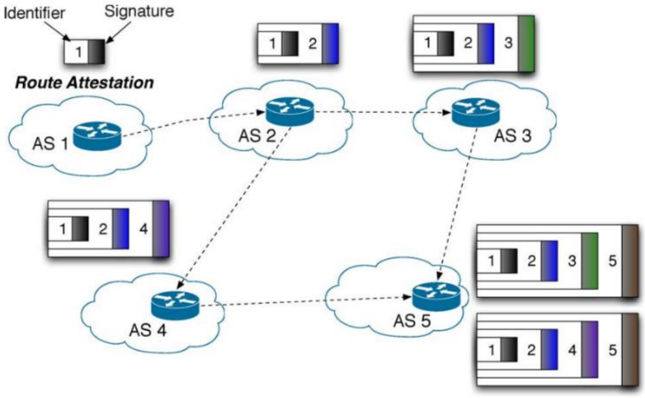


Figure 1: Establishing a BGP connection using S-BGP [10]

2.3.2 soBGP

soBGP distributes certificates to certify the AS and its peering. Only paths that are permitted by the certificates are taken. For example in Figure 2: Path {AS4,AS5,AS2,AS1} instead of {AS4,AS3,AS2,AS1}.

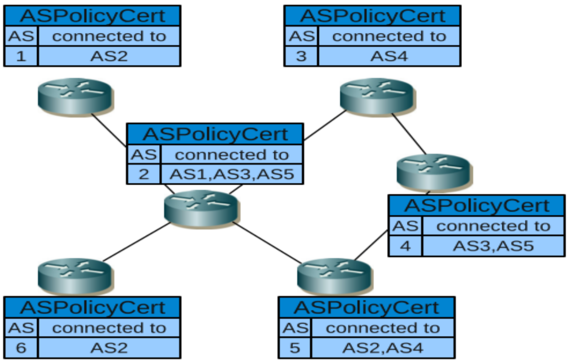


Figure 2: Building a BGP network with soBGP [11]

2.3.3 RPKI

RPKI stands for Resource Public Key Infrastructure and is a security function that runs as an optional function to BGP. However, it is not part of the BGP protocol. This technology provides X.509 certificates according to the RFC3779 standard. Descriptions about RPKI can be found in RFC 6482, 6810 and 6811. Special attention should be paid here to RFC 6810. Here, the RTR (Resource Public Key Infrastructure (RPKI) to Router Protocol) protocol is introduced. This protocol is used for communication between validator (local cache) and router (see Figure 3). The message format for IPv4 prefixes is shown here:

```
0          8          16          24         31
.-------------------------------------------.
| Protocol |   PDU    |                      |
| Version  |   Type   |    reserved = zero   |
|    0     |    4     |                      |
+-------------------------------------------+
|                                           |
|                 Length=20                 |
|                                           |
+-------------------------------------------+
|          |  Prefix  |   Max    |          |
|  Flags   |  Length  |  Length  |   zero   |
|          |   0..32  |   0..32  |          |
+-------------------------------------------+
|                                           |
|                IPv4 Prefix                |
|                                           |
+-------------------------------------------+
|                                           |
|         Autonomous System Number          |
|                                           |
`-------------------------------------------'
```

The certificate certifies the correctness of the association of AS, prefix and maximum prefix length. These certificates follow a hierarchical structure. At the top is the IANA (Internet Assigned Numbers Authority) followed by the RIRs (Regional Internet Registry), such as the RIPE NCC (Réseaux IP Européens Network Coordination Centre). RIPE NCC is responsible for the allocation of address ranges and AS numbers in Europe, the Middle East and Central Asia. The certificates can then be applied for via the RIRs [12]. These certificates are then called ROA (Route Origin Authorisations). These ROAs are cryptographically signed assurances of the validity of an announced IP block. The ROA are then used in the ROV (Route Origin Validation). Here a BGP router checks whether the BGP message does not violate an existing ROA. To do this, the BGP router compares the information in the BGP message with its data from a local RPKI cache. The cache downloads the corresponding RPKI information from the various RIRs. This process shows in Figure 3. The figure also shows that RPKI is an additional function and is not related to the BGP protocol. Therefore, it can be added later and does not require new BGP router hardware. RPKI, however, only checks the AS-Origin. This means that it is still possible to have the data traffic routed via itself as a hop (Type-N hijacking – explanation in chapter 3) [13, p. 6].

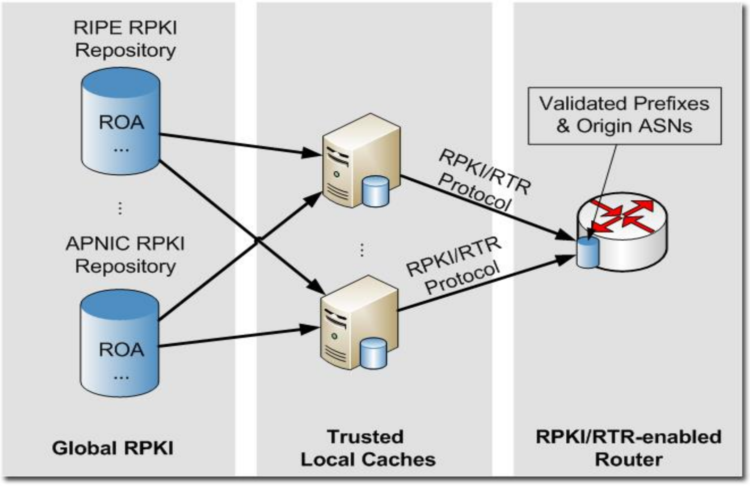


Figure 3: RPKI validation process [14]

2.3.4 BGPsec

BGPsec is a security extension of BGP. It was standardised in 2017 with RFC8205. With BGPsec, secure protocol transport is mandatory. TLS (RFC 5246), TCP/MD5 (RFC 2385) or Tunnel SSH can be used for that, see for example [13, p. 6]. The use of secure layer 3 connections is already in use today. However, this is something that only takes place with the agreement of both peering providers and is not a prerequisite for the use of BGP. With BGPsec, each BGP router must certify on the AS path as soon as it extends it around itself and sends it on. This ensures that no one can modify the path in a BGP message [15]. With certificate signing, the path can no longer be manipulated. The disadvantage is that several prefixes can no longer be transmitted by one update message. BGPsec can be used incrementally. However, it requires a continuous BGPsec path for full functionality.

An advantage that comes with BGPsec and RPKI is that RPKI uses caches. This makes it possible to outsource the computationally intensive operation and thus avoid the otherwise necessary extensions to the BGP router [13, p. 7].

2.3.5 Actual local BGP security

In addition to the already mentioned securing of the TCP connection using TLS, there are other techniques with which ISPs currently secure their BGP connections. These precautions are normally only agreed locally or individually with the peering partner. In any case, a global approach is missing here. Further, these security features are built around BGP and do not make it secure from within (No improvement in the trust model). In some cases, a high level of security can be achieved with the techniques (like ACLs/filters) listed here, but this requires a high level of maintenance (if many changes) of the corresponding technique, which is usually not practicable.

One possibility to protect the BGP protocol against intrusion is IPsec or MD5. Here, the BGP packets are protected by encryption or, in the case of MD5, by a hash value. However, setting up the connection individually with each peering partner is time-consuming but necessary. Another possibility are filters. Here, only predefined prefixes or AS-path are accepted for announcements. There are also solutions that interrupt a BGP connection if a BGP router sends an unusually large number of announcements. Some ISPs are preparing for attacks with de-aggregation. The "more exact" prefix rule is exploited here. It is important to note that most BGP routers only accept announcements up to a /24 network. A possible large peering number is another way to keep the hops to the other ASs small and thus to be treated preferentially in the routing.

GTSM (Generalised TTL Security Mechanism) is based on the idea that peering partners are usually only a few hops away. The BGP router then only accepts BGP messages whose TCP TTL value is not too low [10, p. 108].

2.3.6 BGP Communities

Initial investigations by RIPE have shown that there is a potential danger from BGP communities [16].

BGP communities are a special function of BGP, which were introduced with the RFC 1997 and are currently defined with the RFC 8092. With these communities, which are sent with the BGP routing information, one can trigger various actions. However, there is no uniform standard for this [17]. For example at DE-CIX communities are used to react on DDoS attacks (more detailed explanation in the next chapter) [18]. Thus, communities can be used to distribute information and protect the Internet. On the other hand, this also allows attackers to target routes more precisely. In the example of DE-CIX, communities can be used to redirect data traffic from individual ASs to blackhole servers. If attackers now had this possibility, they could carry out this action for foreign prefixes. Then the connection between e.g. company and customer would be interrupted.

The advantage of using communities is that one can change routes but does not have to change the entire routing tables. Thus, much smaller changes can be made. Alternatively, with BGP, the only option is to delete the entire prefix using "withdrawn". This means that all accesses, authorised and unauthorised, are no longer possible. [19].

As with BGP, BGP communities are not specially protected [20, p. 5]. Here too, the sender is trusted blindly as long as BGP communities are allowed by the corresponding AS.

The BGP communities presented in the chapters are a special case because this problem is relatively new and very individual. Since the breakdown of this topic is too large to be dealt with in the thesis, only an introduction and sensitisation for this topic took place here. The investigations by RIPE show that it is sure that BGP communities have the possibility used for hijacking in the future. Above all, it offers the advantage that the attacks can be triggered more precisely. Here, further evaluation is required in the future and the possible options that are possible by means of BGP communities must be investigated.

Special function of BGP to protect networks with BGP communities

This part is about how BGP can help to prevent other attacks on networks. It is possible to protect against attacks (e.g. DDoS) with BGP. Since the launch of BGP Communities, it has been possible to sort BGP messages by using filter rules and thus there is a further possibility to intervene in the BGP process. Here we will talk about the new possibilities offered by BGP communities. Nowadays, DE-CIX, for example, offers the possibility to prevent DDoS attacks by using BGP communities [21]. Here, the BGP communities are sent with the BGP message, and the infrastructure is informed which AS traffic is to be redirected to a black hole server. This means that users who are also connected to the hijacker AS are no longer able to reach the target server. But all other ASs from which no attack originates are still connected. As a result, the attacked system is no longer overloaded and can respond normally again.

2.4 Current adoption of BGP security

In this chapter, the development and spread of security measures around the BGP protocol will be shown based on different surveys.

A survey [22] of network operators from 2018 shows the current state of security around BGP. The survey shows that most companies are aware of the dangers around BGP. Approximately more than the half (57.1%) of the respondents assume that a BGP attack would affect their network for several hours or longer. Only around 28% expect a short hijack attack between a few seconds and minutes.

One possibility against BGP hijacking is the use of RPKI. But 71% do not use this technology. The main reason for that is the cost, complexity and that the system is not widely use. Currently (08.09.2021), 27.72% of the prefixes (IPv4 only) are listed as valid. Approx. 59.52% have not yet been signed using RPKI and 0.64% are recognised as invalid. This data comes from NIST RPKI Monitor. Here, extensive statistics are compiled regarding RPKI [23]. 60% of respondents in the survey use other defence mechanisms like AS-path/ prefix filtering, de-aggregation or extensive peering (minimising the hop count).

In the area of detection, most network providers have a concept. Here, 61.3% rely on 3rd-party detection service called BGPmon. About 21 % use other software to detect attacks.

To mitigate a BGP hijack, the survey show it is common to use de-aggregation (publish more specific prefixes) and contact the other network operators.

The result of the survey is that BGP hijacking is tried to be solved with rather old methods like de-aggregation or communication between the network providers. Further, filtering and extensive peering are also used to try to weaken attacks. However, each of these reaction options has its limitations (see chapter 2.3). A major problem is that network operators are reluctant to hand over technology and knowledge. Thus, everyone tries to solve the problem for themselves, and centralised solutions are difficult to realize. RPKI is still off to a difficult start, as it needs a global application to work well. This is hindered by cost and complexity, which is compounded by rather limited benefits. [22] No information could be found on the daily distribution of BGPsec.

A literature survey, “A Survey of BGP Security Issues and Solutions” [10], from 2010 gives us an insight into the state of the art at that time and allows us to draw conclusions about the development over the last 10 years. For example, the RPKI system pioneered by ARIN, RIPE and LACNIC is now (2021) well established and growing. At that time (2010), the basic idea was there but no established system [10, p. 110]. This paper presents technologies that should lead to more security in BGP. Some that start at layer 3 (OSI). In BGP, this is TCP. Here, for example, the use of IPsec or GTSM are presented. It was noted that there are good approaches, but that they are not reassessed as this increase's complexity and costs. This leaves only local security solutions such as filters or policies.

The paper "Securing BGP - A Literature Survey" [22] looks at and evaluates various security options for BGP. As in the other paper, techniques such as MD5 or IPsec are explained. Furthermore, this paper points out that MD5 keys should normally be renewed every 90 days and that this makes their use less attractive. Since BGP connections are actually kept as long as possible. It also discusses techniques such as S-BGP or soBGP. This paper notes that no unified solution has yet been found that strikes an appropriate balance between reasonable security and reasonable deployment overhead. It is also not seen as a solution to migrate the whole system at once. The only option, if there is a solution, is to allow a piecemeal renewal.

2.5 Global BGP Message Information

BGP is used to exchange messages with routing information between BGP routers worldwide. Some of these BGP routers have systems installed that fork this communication and make it publicly available. This chapter is about these databases. It should be explained which data are provided and which format they use. These databases offer the possibility to retrieve BGP updates from points around the world. For an example it is possible to detect an BGP hijacking by analyse this data.

2.5.1 Routing Information Service

The Routing Information Service (RIS) is a service from the RIPE NCC. Network operators can join this program as volunteers. They use Remote Route Collectors (RRCs) to collect BGP updates and withdraw. Then the RRCs stores this data to the RIS [24].

RIS database include currently (May 2021) 25 RRCs (3 RRCs no longer supply new data). Basic information about these RRCs can be found in Table 10 in the appendix. The RRCs save the data in MRT format, which is described in RFC6396 [25]. This database is updated every five minutes and a complete RIB (Routing Information Base) is created every eight hours. [26]. RIS Live is available since 2017. It is a WebSocket JSON API to receive live BGP Updates. [27]

2.5.2 Route Views

Route Views is a project similar to RIPE RIS. It was founded in 1995 and started in 1997 with the collection of global routing tables [28]. In April 2021 Route Views has a active connection to 31 RRCs. Basic information about these RRCs can be found in Table 11 in the appendix. They are spread around the world, but most are in North America[29]. It uses also the MRT format like RIS and updates the data every 15 minutes and create a full RIB every two hours [30]. Route Views also provides the option to dial into the BGP collector via Telnet. Here it is possible to have a user with show rights.

2.5.3 Data access

Besides the data sources like RIPE RIS or Route Views, there are other services that use the data sources and want to achieve a better use with their features and technology. One of these systems are called BGPStream. A main point at BGPStream is the use of a meta-data broker to reduce data size [31]. BGPStream then downloads the corresponding files from the various sources and makes them available as a stream. This process is described by Figure 4. The different update intervals of Ripe RIS and Route Views are also arranged here. In the figure, one can see the chronological order of the messages (updates and RIBs) intervals of both data providers. Now these are merged into a single stream which is then output by BGPStream.

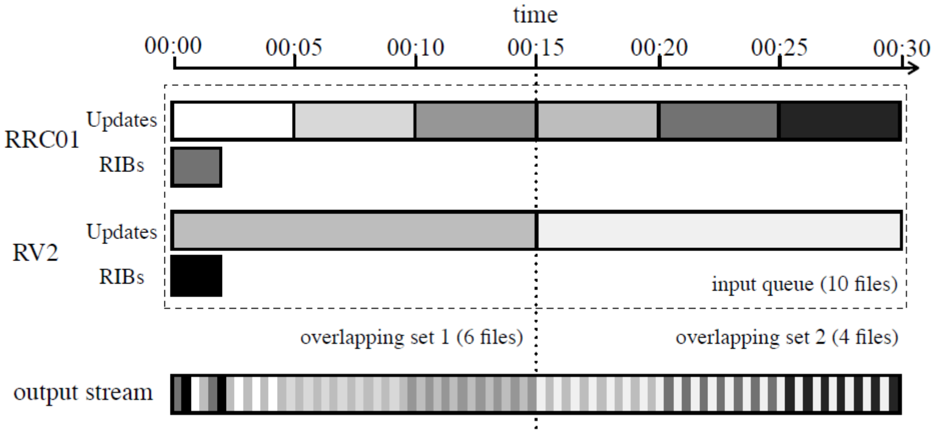


Figure 4: Intra- and inter-collector sorting [31]

exaBGP

exaBGP is a Python module [32] that can be used to communicate with ones own BGP router [33]. This is to enable a user-friendly handling of BGP messages by converting them into plain text or JSON.

3 BGP Hijacking Classification

Currently, there is no standardized classification scheme for BGP hijacking. The goal of this chapter is to collect the various attack possibilities and then present them in a structured manner. For this purpose, various papers are analysed and their schemes are merged [13] [34] [35].

For examples we use the following notation. The origin AS is noted as AS-O and the hijacker AS call AS-H. AS-O announces the prefix 10.10.100.0/21 as his own.

3.1 Classification with the AS-Path

The first characterization is about the announced AS-Path.

- **Origin AS (or Type-0) Hijacking:**
  In this case, the attacker is publishing BGP messages by pretending to be the owner of a prefix that he does not have. In our example it looks as following {AS-H – 10.10.100.0/21}.
  If the attacker leaves the prefix the same, we have a MOAS case. MOAS stands for Multiple Origin AS conflict. In this case, more than one AS advertises the prefix and the different ASs decide where to send the data according to the standard rules (chapter 2.2). In most cases the packets are sent to the AS with the shorter distance. Thus, depending on the network topology, part of the data can be redirected to the hijacker. Figure 5 show a Type-0 hijacking. All data traffic of the PC is sent to the attacker AS H. The connection to AS O still exists, but no one sends data to it.

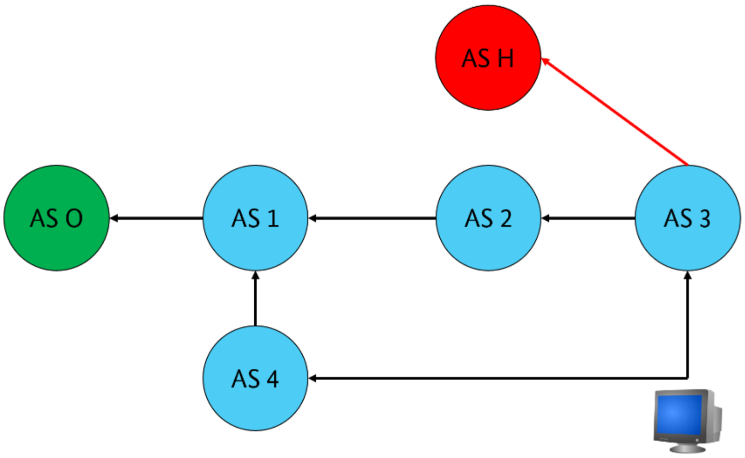


Figure 5: Type-0 hijacking

- **Type-N Hijacking (N≥1):**
  If an attacker places himself without authorisation in the AS path, then this is referred to as Type-N hijacking. Here N is the distance to the Origin AS. A Type-1 hijacking would then be e.g. {AS-H; AS-O – 10.10.100.0/21} and a Type-2 would then be {AS-H; AS-1; AS-O – 10.10.100.0/21}. Figure 6 shows a Type-1 hijacking. Here, the attacker AS is one hop away from the origin AS. In Figure 7, the attacker is two hops away from the origin AS. This is a Type-2 hijacking. This principle continues, as can be seen in Figure 8 with a Type-3 hijacking.

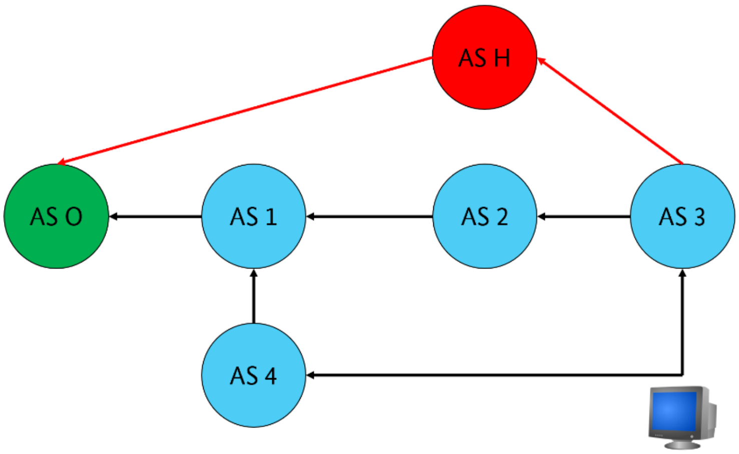


Figure 6: Type-1 hijacking

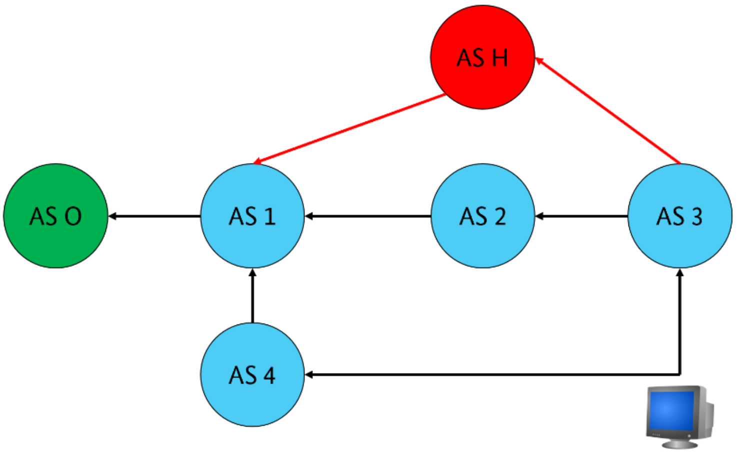


Figure 7: Type-N (N=2) hijacking

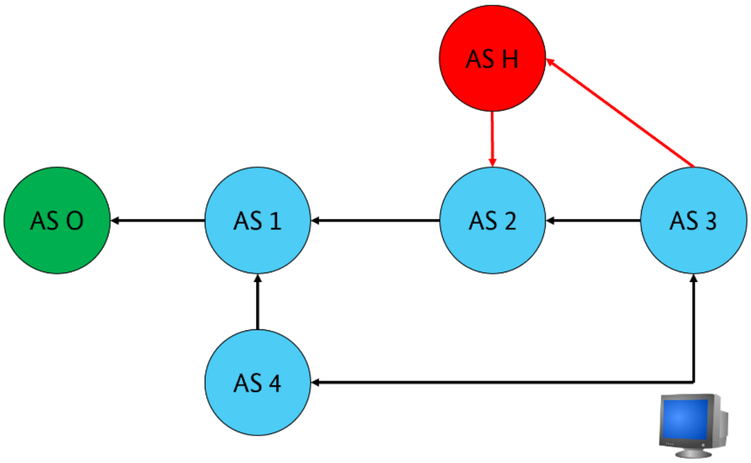


Figure 8: Type-N (N=3) hijacking

- **Type-U:**
  The hijack does not change the AS path (U = unaltered). In this case, the attacker is in the AS path, but this was done by normal process. It is only an attack if, for example, he changes his prefix so that the route via him is favoured by the other BGP routers. This is called sub prefix hijacking. More about this in the next section. This attack can be used by peering AS.
  In Figure 9, the AS H is already in the path, which is normal and desired. Now, the AS H then publish a more specific prefix (AS-H; AS 1; AS-O – 10.10.100.0/22) and thus receive all the data traffic. This data can then be manipulated in the next step. The path seems to be OK and cannot be declared as Type-0 or Type-N hijacking.

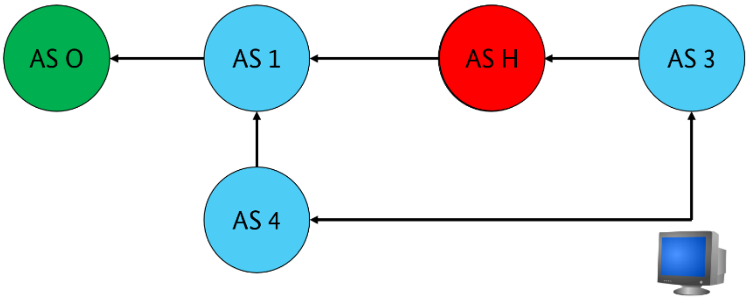


Figure 9: Type-U hijacking

- **Type-P:**
  In this case, the AS path information is artificially altered by the hijacker [36]. This has the advantage that the AS path becomes shorter and is thus better accepted by other BGP routers. Furthermore, the attack is less noticeable, e.g. through traceroute testing [37, p. 26]. Let's assume Figure 10 shows the real connection present. The AS H, however, is now pretending to be the upstream provider for AS O. Then, in the case of a Type-P attack, the data packets would be manipulated as a example the TTL from the

ICMP. So that it is not noticed that the data packets were also routed by AS 1 and AS 2.

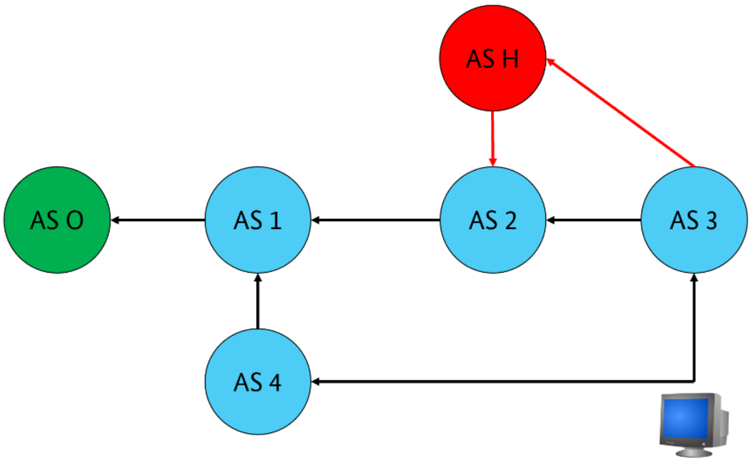


Figure 10: Type-P hijacking

3.2 Classification with the prefix

The next characterization step is by the affected prefix:

- **Exact prefix:**
  The attacker uses exactly the exact same prefix as the AS origin in his attack. With this method, only a part of the Internet traffic is affected. The decisive factor here is the metric, i.e. in the simplest case the number of hops. In this case, it is referred to as a MOAS.
- **Sub prefix:**
  With sub-prefix hijacking, the attacker announces a smaller range, than that of the legitimate AS. For example {AS-H – 10.10.100.0/24} or {AS-H; AS-X; AS-O – 10.10.100.0/24}. Due to the property of BGP, more specific prefixes are preferred. Thus, the data is redirected to the attacker even if the number of hops is higher than to the origin AS.
- **Squatting:**
  With squatting, an attacker uses an IP range that is not (yet) announced by its owner.
- **Super prefix:**
  In this case, the attacker uses a larger prefix. For example, AS-O announces 10.10.100.0/21 and AS-H announces 10.10.100.0/19. With this method, AS-H can pretend to own this range or to be a transit AS. However, this method is very unattractive because it only works if the prefix to the AS-O is withdrawn.
  An example where this attack would make sense is when an attacker announces a super prefix. Since RPKI does not block super prefixes (investigation from chapter 6.6), the announcements find a wide distribution. Then the BGP router of the victim can be attacked so that its announcement is interrupted. So now only the attacker's announcement is there.

3.3 Classification at the data/control plane

Now follows the classification by use of the data/control plane manipulation. The definition for data/control plane for this part of the thesis can be found in the chapter 2.2.

- **Blackholing:**
  With blackholing, the aim is to completely disrupt the connection with the AS-O and thus make its services inaccessible. In the best case, for the attacker, all data packets are now sent to the AS-H.
  This attack methodology is an obvious one. Once the new BGP route has been adopted by many ASs, the services that are reachable via the AS-O are no longer available. Customers cannot access the services and providers see a collapse in access and bandwidth to their services. This means that such an intervention is quickly recognised compared to the other methods. In this classification, blackholing is the only one that attacks only the control level.
- **Man-in-the-Middle attack:**
  A Man-in-the-Middle (MitM) attack is used to sniff or manipulate the data that is sent to AS-O. To achieve this, the AS-H manipulates the BGP messages so that it is included in the AS path. The closer the AS-H is to the AS-O, the greater the proportion of the total data traffic that passes through the AS-H. However, this increases the risk that the hijacking will be detected. An evaluation of this statement is given in chapter 5.5 in connection with the results of the BGP hijacking detection software.
- **Imposture:**
  When an attacker hijacks the traffic and responds in the name of the origin AS, it is called imposture. This is a very extensive and complicated attack methodology. Not only the routes in the BGP system have to be manipulated. There must also be a system behind the AS, which then reacts to incoming requests and responds accordingly. A simple example would be a shopping website. The customer enters the URL and is now redirected to the replicated page. Here he enters his login data, and the attacker can now use these to log in to the correct page.
  Another possibility would be to use prefixes that companies/organisations own and are not announced. This is called squatting. The aim is to discredit the victim. An example of such an attack is the spam attack on the company Northrop Grumman in 2003, where a lot of spam mails were sent. As a result of this attack, the IP addresses were included in almost all block lists [13, p. 3].

The MitM and Imposture attack can be distinguished in two basic ways. The first is to try to be the prefix owner or to remove hops (Type-P hijacking) and then forward the data to the original owner. The second way is to distribute routes where one is in the chain, but all hops are valid. To improve the effect, a sub prefix hijacking attack can also be used (Type-U hijacking).

In the first case, the success is greater, since routes with a smaller number of hops are usually preferred. The danger, however, is that BGP routers that are between Hijacker-AS and Origin-

AS will also take over this route. This interrupts the data flow and the the MitM attack becomes blackholing attack.

In the second case, this danger does not exist, but the path length is longer and is therefore not so strongly preferred by BGP routers. Since the necessary BGP routers between Hijacker-AS and Origin-AS are included in the path, the BGP loop detection prevents these routers from taking over this route.

Finally, there is a classification of what was the overriding goal of the attack. This information is intended to classify the intention of an attack. However, they are not part of the explanation of how a BGP hijacking works.

- **Availability:**
  An attack on the availability of the system normally only affects the information in the routing tables. The connectivity of the AS-O is affected. An example of this is blackholing, which is equivalent to a DoS attack.
  The end user is directly informed that the service is not available and may lose confidence in the security of the service.
- **Authenticity:**
  If an IP range is taken over by the attacker, this poses a considerable problem for the authenticity. The attacker can try to discredit the IPs of organisations/companies and thus introduce this IP range into filtering algorithms. If these IP addresses have then lost their authenticity and are included in blocking lists, this can cause considerable effort until the IP addresses are accepted everywhere again. When the AS-O takes over again, the data traffic is disrupted until the filters are updated again. Another problem arises from the web server where a spoofing attack is possible by manipulating the BGP protocol. Here, the DNS IP looks correct for the user, but the query is forwarded to the AS-H. So, it is possible to answer with a wrong IP address to this query and connect the user with an attacker server.
- **Confidentiality:**
  Any attack method always allows the confidentiality of information to be compromised. Blackholing is a slightly different matter, as the data is normally only deleted and not changed or analysed. Here, it is more the user's confidence in the service that is attacked. The attacker can enable a bidirectional connection and intercept authentication data when he uses e.g., a Type-0 imposture attack. For example, between a shopping website and a customer.

3.4 Discussion

The concept created here is formed from various sources, which in turn have already created definition concepts. The concept created here forms a three-stage structure. First the AS path is evaluated, then the prefix and the data/control level. This allows a simple and clear assessment and a targeted path in the creation of countermeasures. Such countermeasures are outlined in the following chapter. With the result of the classification, one can try to find the intention of the attacker. The classification of the attack possibilities creates an awareness of BGP hijacking and thus enables a quicker reaction. The benefit of this model, which is created here, make sense to build as a uniform standard in order to facilitate worldwide understanding in the event of attacks. This would avoid misunderstandings and simplify a common approach. At the end of this chapter, a decision matrix (next page) has been created. This represents the three-stage decision-making structure. It also gives an insight into the impact and basic reaction to the various possible attacks.

Path classification

- Type 0 or N
- Type U
- Type P

Prefix classification

- Exact prefix
- Sub prefix
- Squatting
- Super prefix

Data/control plane classification

- Blackholing
- Imposture
- Man in the Middle

Impact

- Blackholing
  - Type 0
    - MOAS conflict
  - Shortest path wins
  - Service availability is under attack
  - Sub prefix:
    - attack: all traffic for the sub prefix goes to the attacker
  - Type N:
    - AS can be disrupted by the attacker redirecting large amounts of data to another AS
    - Not as large as Type-0, but more precise and more difficult to discover.
    - Data is discarded at the hop and not forwarded to the destination
- Imposture
  - Data can be stolen (only remote point)
  - Shortest path wins
  - Type 0
    - MOAS conflict
- MitM:
  - Data can be stolen (remote and server)
  - The attacker tries to get more data by changing the prefix (Sub prefix).
  - Type N:
    - Redirecting traffic
- Squatting
  - IP addresses can be brought into disrepute or used for identity checks
- Super prefix
  - No Impact
- Type P
  - Data is manipulated so that the higher hop number is not noticed

Reaction

- Blackholing
  - De-aggregation
  - Contact attackers upstream provider
- Imposture
  - De-aggregation
  - Contact attackers upstream provider
  - All users should stop traffic to the attacked prefixes
- MitM
  - De-aggregation
  - Contact attackers upstream provider
  - Encrypt all data
- Squatting
  - Always announce all prefixes
- Super Prefix
  - Watch out for attempts to disrupt your own announcement
  - Ensure that your announcement is available worldwide

4 Prevention, reaction and analysis

This chapter discusses different ways to prevent, detect and analyse BGP attacks. If an attack occurs, it is important to know how to react to the attack. On the one hand, how to mitigate or interrupt the attack, and on the other hand, how to analyse it, what is becoming more and more important nowadays. A focus is on the impact and spread in its own system and in the world BGP network.

In the current chapter, different methodologies and software are presented and their use for different possibilities is explained. There is some overlap between prevention and reaction, as some tools support both. With this information, a basis should be created for what possibilities there are, for example, for an emergency plan in the event of an attack.

4.1 Possibilities of prevention against BGP hijacking

4.1.1 MOAS

MOAS can be divided into two basic areas. Useful and malicious MOAS.

For example, when an attacker attacks the IP prefix with a Type-0 hijacking. In this case, the attacker pretends to be the origin AS. Thus, there are now two ASs that claim the data. Here it is a malicious MOAS, because the attacker will not handle the data in the intended sense.

On the other hand, there is also the useful MOAS. Here, the owner of the prefix publishes his prefix from different ASs or announces the same AS number from different locations. Here it makes sense to have the largest possible, worldwide uniform coverage in order to keep the hop count to the various ASs as low as possible. According to a standard rule, the shortest AS path is normally preferred in BGP. This procedure has the advantage that one can now be reached worldwide with only a few hops and thus a BGP attack will have a significantly lower impact.

If this technique is used as protection against BGP attacks, a further distinction can be made between two approaches. First, is that one has a worldwide infrastructure and there are corresponding servers (behind ASs with same number) that respond. This is done, for example, at Cloudflare [38]. The second way would be to collect this data, of different (foreign) AS announcing the prefix, and then forward it via VPN to the correct AS or infrastructure.

4.1.2 RPKI

RPKI can be used to defend against Type-0 attacks. However, this requires that the prefix owner has requested a ROA in which he specifies the prefix, AS and maximum prefix length. Furthermore, the BGP routers must also support RPKI in order to recognise a false announcement.

4.1.3 New BGP protocols

The use of new protocols such as S-BGP, SoBGP or BGPsec would protect BGP data traffic very well when it used worldwide. This would mean a change from "I trust every message" to "every message has to prove that it is true". With the appropriate security of the certificates, BGP hijacking would no longer be possible. This can be seen very well in the S-BGP or BGPsec protocol. Here, each AS must sign the message when it is forwarded, and the receiver can check each individual AS path extension. If, in addition, the legitimacy of the announcement is confirmed with RPKI, then the origin and path are completely verifiable.

4.2 Reaction options against BGP hijacking

The reaction possibilities are complex with BGP because it is a decentralised system. It is therefore not possible to stop the wrong announcements from a central point. However, there are ways to mitigate an attack.

The first option is to use the basic feature of BGP that more specific prefixes are preferred, even if the AS path is longer. This is known as de-aggregation. In this case, it is also useful to have a corresponding certificate for RPKI. If there is no corresponding RPKI certificate, the new announcement is discarded by the BGP routers with RPKI. This would significantly reduce the benefit.

The second option is to identify the attacker and inform this AS about the misconfiguration or hijacking. It is also possible to inform the upstream/peering provider and to ensure that the incorrect BGP messages are manually filtered/deleted, or the peering is completely stopped.

An example of this is the hijacking attack on YouTube by Pakistan. Pakistan wanted to block YouTube for their country. Due to a mistake, the announcement was distributed worldwide and YouTube was offline for about 2 hours. Both possibilities were used here. YouTube gave out a more specific prefix and the Pakistani upstream provider deleted the false announcement using Withdrawn messages [39].

However, it must be clear that the possibilities can only work to a limited extent and thus a part of the data flows further to the attacker. Only the widespread use of technique like RPKI and BGPsec would remedy this.

4.3 Impact analysis of a BGP hijacking

An important part of the response is an assessment of the impact of the attack. New research approaches offer new ways and possibilities to determine this by means of scientific methods. The paper "Estimating the Impact of BGP Prefix Hijacking" [40] offers a very good first approach. It shows (explanation and practical tests) three different ways of estimating the impact of a BGP hijacking attack. In the following, these three possibilities will be presented, and the most important findings will be listed. Furthermore, there will be a theoretical consideration of how the implementation in the enterprise sector can look. Finally, BGP hijacking detection software is discussed. In contrast to the previously mentioned technology approaches, this is already in widespread use.

Currently there are three different sources of information:

- Pings
- Route collectors (RC) like RIPE RIS or Route Views
- RIPE Atlas probes (RA)

The main difference between the different measurement methods is the use of public monitors and the use of the IT infrastructure itself.

4.3.1 Pings

Pings belong to the Internet Control Message Protocol (ICMP) and are the only technique in this selection that works without external infrastructure. The attacked AS (victim AS) pings a previously defined IP address to the AS list. In this list, IP addresses (possible to ping) are linked to AS numbers. If the IP addresses respond, then these ASs are not affected by the attack. A special feature here is that this analysis only works with blackholing and imposture attacks. Studies have shown that it makes sense to ping between two and three IP addresses per AS. In order to have the largest possible coverage, as many ASs as possible must be pinged. With an estimated 90,000 ASs currently available [41], corresponding resources must be available. This reduces errors caused by interruptions in availability [40, p. 6].

4.3.2 Route collectors (RC)

Using route collectors, it is possible to obtain routing information from BGP routers distributed worldwide. By analysing these BGP packets, an attack can be detected, and an assessment of the affected region can be made based on the geolocation of the corresponding router/AS number. However, the strong influence of geolocation must be taken into account [40, p. 4]. Before using this technique, it must be analysed in any case whether sufficient RC are in the desired monitoring area, so that a sufficient overview can be obtained. To do this, first the location of the RC/AS must be determined and then the locations of the corresponding peers whose BGP messages are sent to the RC/AS. A project to evaluate the importance of different ASs is being carried out by the TU Munich [42, p. 8].

4.3.3 RIPE Atlas probes (RA)

RIPE Atlas is a project of the RIPE NCC. This project comprises around 12 thousand probes (as of 30.06.2021), which can be used for various measurements (e.g. traceroute or ping). With the help of the probes of RIPE Atlas, traceroute commands can be sent to the own AS. In this way, it can be determined which AS, in whose network the probe is, is sending the data correctly or incorrectly.

An advantage over the Ping method is also the usability with MitM. A feasibility study has already been undertaken for the method [40].

Let us first look at the facts about the RIPE Atlas system. There are (as of 30.06.2021) 11180 probes and 723 anchors, which brings the total to 11903 measuring devices. An anchor is a probe with extended functions/capacities. According to current analysis by RIPE, there is an IPv4 ASNs overlap of 3671 probes, which makes a share of 5.128%. For IPv6, there are 1638 probes with a share of 6.123%. The country coverage is 88.265% with 173 countries [43].

If we now look at Figure 11 we see that the distribution of the probes is quite uneven. From this we can conclude that in the event of an attack, global assessment is more difficult in countries with poor probes coverage. On the other hand, we can expect a good result for local assessments in countries like Germany or the USA. This is due to the higher coverage. Thus, it would be interesting, for example, for companies that are based in Germany and have their main field of activity, to carry out a damage analysis using this system.

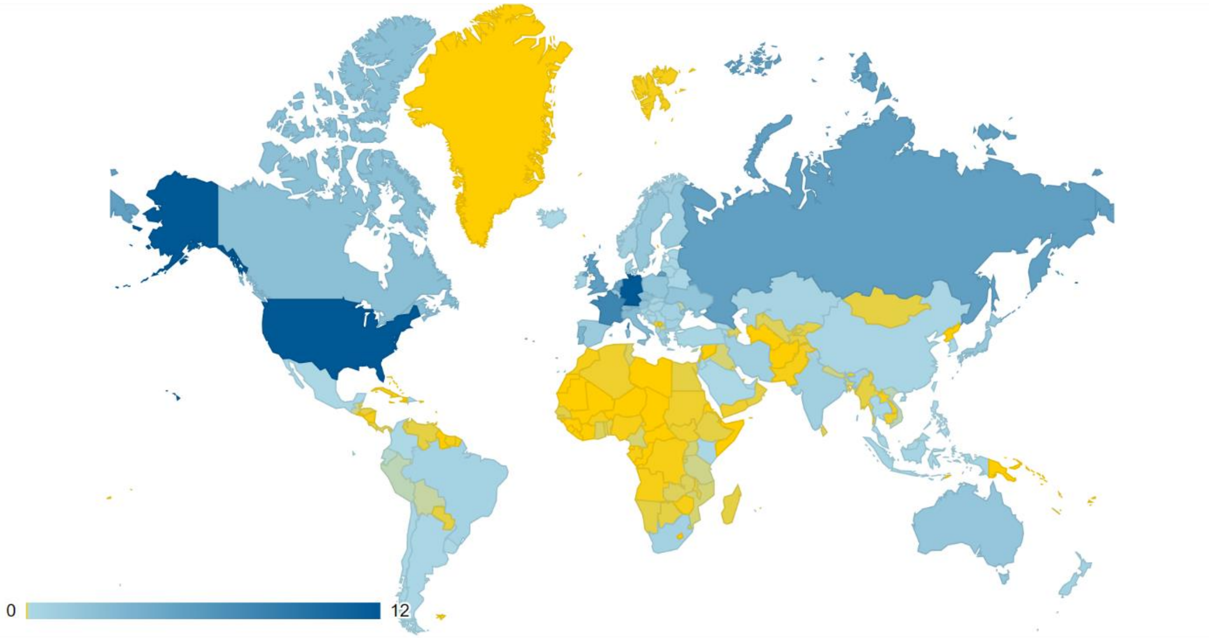


Figure 11: RIPE Atlas global distribution (in percent) [43]

Furthermore, when using Ripe Atlas, it must be taken into account that this service is not free of charge. Credits are required, which are obtained by becoming a member of RIPE or by supporting the project [44].

4.3.4 Detection Software

The classification of BGP hijacking detection software into the three categories (prevention, reaction, analysis) is not easy and are in themselves a class of their own. Some software covers several areas and have the possibility to adapt the BGP announcement after the detection of an attack. Nevertheless, this part is included in analysis, as this is the core function of the various software products.

Detection tools such as BGPalerter or Artemis provide a better monitoring than just the view from one’s own network. With this software, one' s own announcements and their impact in the worldwide BGP data traffic can be seen. In addition to the open-source tools already mentioned, there are also paid software from various providers. An example for paid software is BGPmon from Cisco. These can be advantageous because they may have other data sources besides the public ones. A decisive disadvantage is that one cannot understand exactly how the tool works. So one does not know exactly which attacks the software doesn't recognise. It has been shown (chapter 5.5) that the use of detection software is already possible with low hurdles and brings advantages.

5 Comparison of BGP hijacking detection tools

This chapter will now compare different tools with each other and show their advantages and disadvantages. In the following, the focus will be on the software Artemis and BGPalerter. These are the most up-to-date programmes. The differences between the two software tools, such as the recognition possibilities or the data basis, will be described.

5.1 Other developments of BGP hijack detection tools

There were already implementations of BGP hijacking detection software, before Artemis and BGPalerter. There are conceptual approaches to solutions, some of which have been implemented.

An example here would be PHAS (Prefix Hijack Alert System) [45]. PHAS uses Route Views and RIPE RIS as data source. Unlike BGPalerter or Artemis, it is a SaaS (Software as a Service). It provides a web interface where one can enter one's email address and receive a notification when PHAS detects a hijack. PHAS takes a different approach to configuration than Artemis or BGPalerter. It does not work with a config file, but with an origin_set value. The value origin_set contains all known ASs for a prefix that PHAS has recognised up to the current time. If a new AS is added later, either by a deliberate MOAS or a hijacker, the new ASs are added to Origin_set. This change then sends a notification to the persons who have subscribed to this prefix. Currently it looks like PHAS is not being maintained or developed. The website is offline and the service is therefore not available.

Another software is HEAP. HEAP stands for Hijacking Event Analysis Program and is developed by the Technical University of Munich. This tool is not a BGP hijacking detection software. It is rather an additional software to carry out an extended check whether an attack is taking place (Figure 12). For this purpose, HEAP examines the register inference (Internet Routing Registries (IRR)), the topology and SSL/TLS. Using SSL/TLS as an example, HEAP examines whether a valid certificate is present. This allows blackholing and imposture attacks to be detected [46]. However, HEAP is not currently available as an open-source project and is not a stand-alone software for detecting BGP attacks. Therefore, it will not be discussed further here. However, since there are current presentations [42], it makes sense to observe the further development.

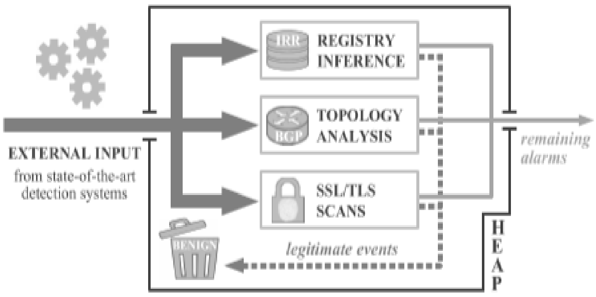
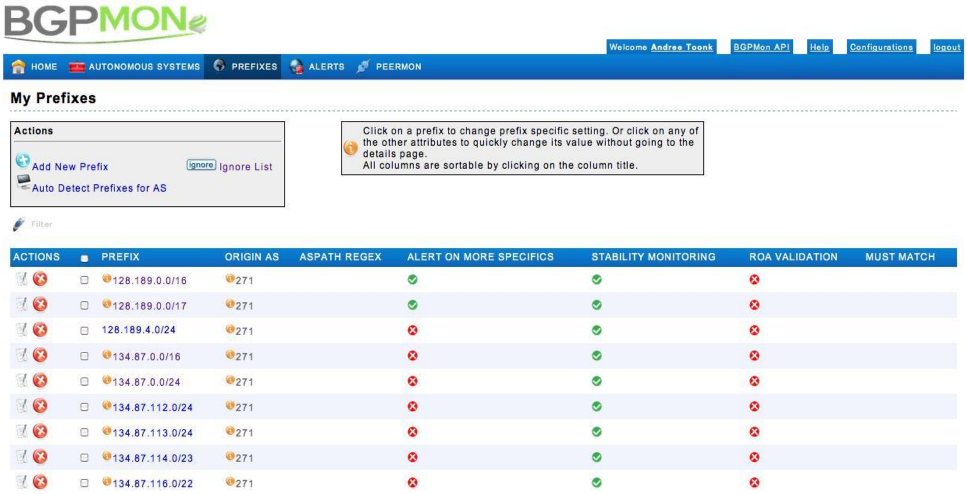


Figure 12: Schematic structure/functionality of HEAP [47]

Besides theoretically developed concepts, open-source projects and projects that are not open to the public, there are also service providers such as BGPmon from Cisco. This service is also only offered as SaaS. However, it provides an API to integrate it into the company's internal IT. In addition to BGP hijacking, BGPmon also detects ROA (RPKI) errors in its own prefixes. One service is the telephone/SMS notification of an alert. This is advantageous because it relies on alternative alerting. E-mail can have the problem that the own mail servers are also affected by the attack and cannot receive an alarm message. It is not known exactly how many probes or RRC are connected to BGPmon. According to BGPmon's own statement, however, there are "hundreds" [48]. The high coverage also allows country-specific availability to be displayed in the event of an attack. Figure 13 shows the BGPmon web interface.



My Prefixes

Actions

Add New Prefix

ignore Ignore List

Auto Detect Prefixes for AS

Click on a prefix to change prefix specific setting. Or click on any of the other attributes to quickly change its value without going to the details page.
All columns are sortable by clicking on the column title.

ACTIONS	■	PREFIX	ORIGIN AS	ASPATH REGEX	ALERT ON MORE SPECIFICS	STABILITY MONITORING	ROA VALIDATION	MUST MATCH
	□	128.189.0.0/16	271		✓	✓	✗	
	□	128.189.0.0/17	271		✓	✓	✗	
	□	128.189.4.0/24	271		✗	✓	✗	
	□	134.87.0.0/16	271		✗	✓	✗	
	□	134.87.0.0/24	271		✗	✓	✗	
	□	134.87.112.0/24	271		✗	✓	✗	
	□	134.87.113.0/24	271		✗	✓	✗	
	□	134.87.114.0/23	271		✗	✓	✗	
	□	134.87.116.0/22	271		✗	✓	✗	

Figure 13: Route monitoring BGPmon [48]

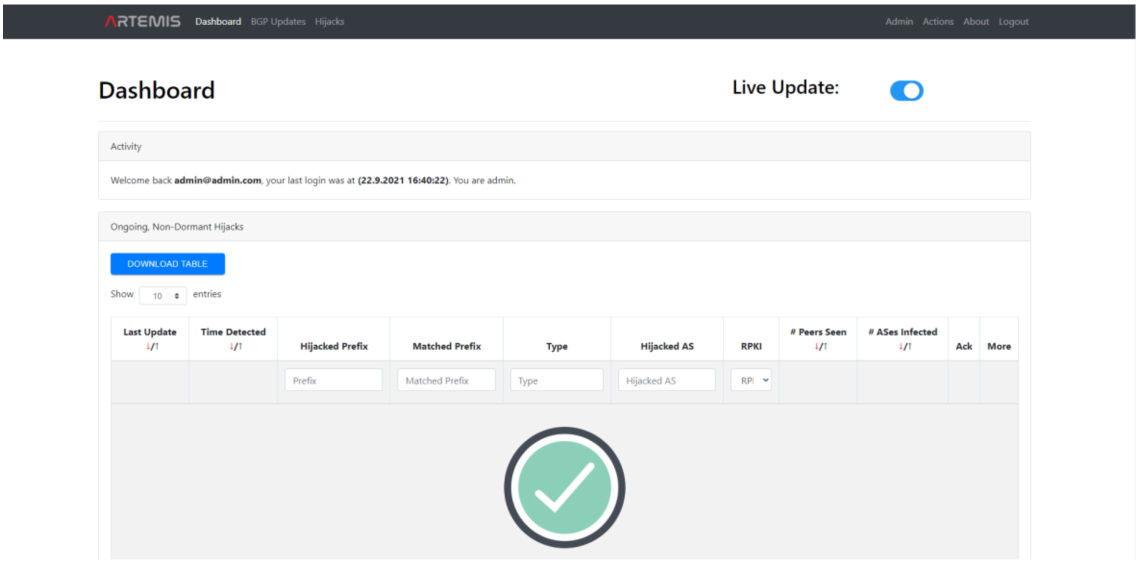
TaBi is a free software to detect BGP hijacking/conflicts. It uses the data from MRT files such as those available via RIPE RIS [49]. In the course of evaluation, it became clear that this tool does not provide live monitoring or a user-friendly, like Artemis or BGPalerter. Artemis and BGPalerter provide an integrated automatic query of the data. With TaBi, the MRT files must be downloaded manually for evaluation. Furthermore, the software does not correspond to the technical/functional scope of current available software. The decision has therefore been made not to investigate TaBi further and not to include it in the comparison.

5.2 Artemis

Artemis (Automatic and Real-Time dEtection and MItigation System) is a tool developed by researchers at the Foundation for Research & Technology - Hellas (FORTH) and the Center for Applied Internet Data Analysis (CAIDA). This project is freely available via GitHub (https://github.com/FORTH-ICS-INSPIRE/artemis) and thus also offers the possibility for external contributors to work on it [8]. The first version v1.0.0 was released in December 2018. Artemis is therefore one of the newer software compared, for example, to PHAS. The whole system comes out-of-the-box and does not require a deep understanding of the software or operating system. As of 30.06.2021, the current version is v2.1.0 (Bellerophon) [50].

The software uses the public Remote Route Collectors (RRC) of RIPE RIS and Route Views for the data basis (see chapter 2.5). Both services also offer live data from RIPE RIS and Route Views, which are also processed. Furthermore, exaBGP (see chapter 2.5) can be used to include one's own BGP routers as data sources. A special feature is the possibility to read in historic data to analyse older BGP data sets.

Artemis provides a web interface (Figure 14) with a dashboard. Current and past attacks are displayed here. Furthermore, the configuration file can be customized via the GUI. The most important functions are available over the Artemis the GUI. Nevertheless, in some cases it is necessary to work with the shell.



Dashboard

Live Update:

Activity

Welcome back **admin@admin.com**, your last login was at **(22.9.2021 16:40:22)**. You are admin.

Ongoing, Non-Dormant Hijacks

Show 10 entries

Last Update ↓/↑	Time Detected ↓/↑	Hijacked Prefix	Matched Prefix	Type	Hijacked AS	RPKI	# Peers Seen ↓/↑	# ASes Infected ↓/↑	Ack	More
		Prefix	Matched Prefix	Type	Hijacked AS	RPI				

Figure 14: Artemis web front end

Currently, Artemis is specified with the following attack detections shown in Table 2. The coding of the hijacking works in Artemis by placing a letter in the first position that refers to the attack type. For example, exact- or sub prefix hijacking. Next is the distance from the attacker in the AS path. So, zero for a false origin AS (Type-0 hijacking) and one for one that pretends to be a false upstream provider (Type-N hijacking). In the third position is a placeholder to indicate attacks on the data plane. This information cannot be used at present because it is not yet implemented. The last placeholder is used to visualise problems with the policy.

The functioning of Artemis can be described as follows. Various actions are triggered with the configuration file. First, the prefixes that Artemis is to monitor are defined. To do this, the corresponding messages that match them are downloaded from the various RRC sources. In the next step, the AS path is checked using the origin AS and first hop specified in the config file. If the BGP message deviates from the norm specified in the configuration file, this triggers an alarm. RPKI validation can also be activated. In this case, Artemis accesses either its own validation server or via an interface to GitHub NLnetLabs/routinator.

Table 2 Artemis detection capabilities(excerpt from [51])

Designation in Artemis	Explanation	Hijacking Type after definition (chapter 3)
S|0|-|-	sub-prefix announced by illegal origin	Type 0; sub prefix
S|1|-|-	sub-prefix announced by seemingly legal origin, but with an illegal first hop	Type 1; sub prefix
S|-|-|-	not S|0|- or S|1|-, potential Type-N or Type-U hijack	--------
E|0|-|-	exact prefix announced by illegal origin	Type 0; exact prefix
E|1|-|-	exact prefix announced by seemingly legal origin, but with an illegal first hop	Type 1; exact prefix
Q|0|-|-	squatting hijack (is always '0' on the path dimension since any origin is illegal)	Squatting
||*|L	no-export policy violation	--------
E|-|-|-	not a hijack	--------

To mitigate the attacks, pre-defined scripts can be automatically triggered by the tool as soon as a hijacking attack is detected. Technically, Artemis runs on an Ubuntu Linux system in a multi-docker container. These are connected to each other via MBUS [51].

Artemis covers basic detection functions. These are exact and sub-prefix hijacking as origin or first hop. Further recognition options such as Type-N, Super Prefix, Type-U or Type-P are not yet included. As more functions are supported, the benefit of this software increases.

In the future, the developers from Artemis plan to support also detecting hijacking of Type N (N>1). The current statement of a member (Vasileios Kotronis, private correspondence, 02.09.2021) of the Artemis project is: „ We have not implemented this yet since the asserted link info requires a more global approach and is a bit orthogonal to the localized one we have adopted in Artemis. It is a long term plan though. "

5.3 BGPalerter

BGPalerter is one of the newer tools, along with Artemis. The first version (v1.19.1) was published on GitHub (https://github.com/nttgin/BGPalerter) in September 2019. In the basic version, it provides a ready-made software that can be used without a deep understanding of the software or OS. The tool can be used on Linux, Mac, Windows and Docker [52]. BGPalerter also provides only one shell configuration. But it is possible to include e.g. alerta (https://alerta.io/) and get a graphical overview from the alarm messages.

Because BGPalerter only uses RIS Live as a data source, this software is very resource efficient. This is because the messages come directly via web socket and not via MRT files [53]. It is possible to retrieve the BGP messages of the last 2 hours collected by RIS Live using RISDump [54]. However, this is a disadvantage in terms of worldwide coverage and detection of attacks. BGPalerter makes configuration much easier with its autoconfiguration. Here, for example, the AS number is used to determine the corresponding prefixes. This works via the API of Ripe Stat. Currently, BGPalerter is specified with the following attack detections which are in Table 3.

The way it works is very similar to Artemis. Here, too, there is a configuration file. With the prefixes stored there, BGPalerter downloads the appropriate BGP messages via RIS live. The messages are then analysed using the filters stored in the configuration file. If messages are found that violate the filter rules, an alarm is triggered.

Table 3 BGPalerter detection capabilities [54]

Designation in BGPalerter	Explanation	Hijacking Type after definition (chapter 3)
monitorHijack	This monitor triggers an alarm if there is a Type-0 hijacking with the exact or sub prefix.	Type 0; exact and sub prefix
monitorVisibility	If the number of RRCs that do not see updates via the prefix falls below a definable value.	--------
monitorPath	Here one can create one's own filters via the AS path. If these no longer apply, an alarm is generated. With this filter, Type-N hijacking detection can be realised.	Type N
monitorNewPrefix	This monitor alarms if the correct AS announces a more specific prefix that is not stored in the configuration.	--------
monitorAS	Monitoring of the own AS and alerting when a new prefix is announced that is not present in the configuration.	--------
monitorRPKI	Monitors AS and prefix and reports if the RPKI is invalid or not covered	--------
monitorROAS	Monitoring the ROAS to see if they edited, added or removed; expiring ROAs; TA malfunctions.	--------
monitorPathNeighbors	The AS neighbours can be specified via the configuration. The alarm is triggered when false neighbours are detected in the AS path. With this filter, Type-1 hijacking detection can be realised.	Type 1; exact and sub prefix

5.4 BGP hijacking software comparison results

In Table 4 the two software, Artemis and BGPalerter, are compared with each other. The main part of the comparison is the functionality for the user and the BGP data connection. Not included in the comparison are the technical functionalities, e.g., how the software is designed.

Table 4 Comparison matrix for Artemis und BGPalerter

	Artemis	**BGPalerter**
Data source	RIPE RIS, Route Views, exaBGP, Historic BGP Messages, RIS Live, Route Views Stream [8]	RIS Live [53] RISDump (last two hours from live)[55]
Operation System	Ubuntu Linux 16.04+ [56]	Linux, Mac, Windows, Docker [52]
Detection/ Mitigate	Yes/ Yes (own Scripts)	Yes/ No
Live BGP Data	Yes	Yes
Historic BGP Data	Yes	No
Reporting	Dashboard, Script, Mail, Slack	File, Mail, Slack, Kafka, Syslog, Alerta, Webex, HTTP, Telegram, PullApi
RPKI support	For a hijacking, check whether it is an ROA for the prefix and what status it has.	Checks that there are no false ROAs being published via the RPKI system.
BGP communities	Scans the received BGP messages according to a pre-defined filter is possible [57].	Not included
BGP hijacking detection capabilities after chapter 3	Type 0 and 1 with exact and sub prefix	Type 0 and 1 with exact and sub prefix; manual Type N hijacking

The comparison of the different software showed that overall Artemis is ahead, if you exclude the manual type-N hijacking at BGPalerter. But BGPalerter also has functions that make it worthwhile to use. Artemis shows clear advantages in the points of the databases. The databases used here are the long-term databases RIPE RIS and Route Views, as well as the live streams RIS Live and Route Views Stream. Artemis can also mitigate attacks by executing self-programmed Python scripts when an attack is detected. These can, for example, trigger a reconfiguration of the network technology.

When using RPKI, BGPalerter is better. Here, it is constantly sensed whether there is a deviating (through misconfiguration or attacks) ROA. This clearly shows that BGPalerter is a very good self-monitoring software compared to Artemis. Artemis, on the other hand, provides good functions to better detect an attack. In BGPalerter, the following monitors (from Table 3) show that there is also a strong focus on self-monitoring.

- monitorVisibility
- monitorNewPrefix
- monitorAS
- mointorRPKI
- monitorROAS

A big step forward would be if Artemis could also detect Type-N hijacking. BGPalerter has the basic ability to detect Type N hijacking (see "monitorPath" in Table 3). However, the filter must be created manually, which can be difficult with a very dynamic change of peering partner. The detection capabilities Type 0 and Type 1 hijacking are covered by both software.

Both software tools are useful. With BGPalerter, the focus is more on internal/self-monitoring and with Artemis more on external/attack monitoring.

5.5 Comparative Conclusion

After examining the two software tools, Artemis and BGPalerter, the question arises, how well the tools handle BGP hijacking. Both tools have the basic ability to detect the simpler hijacking attacks. These are Type-0 and Type-1 hijacking on the exact or sub prefix. However, the functionality to detect Type-N (N>1) hijacking automatically is still missing. This is especially problematic in view of MitM attacks. Here, the attacker no longer receives a large share of data, but is also not recognised. Artemis has the better detection capabilities, as it uses more data sources and is therefore more likely to detect them. Modern BGP hijacking detection software have significantly reduced the necessary knowledge and the complexity of these application. Since no understanding of the data structure (e.g. MRT files) or how this software works is necessary. Both software offers a low-threshold implementation and maintenance effort.

However, the tool and the data sources are open public projects. This means that everyone can find out how the software tools work and where and how they get and process data. This gives attackers the advantage of circumventing exactly these alarm mechanisms. On the other hand, it also makes it possible to detect and fix bugs in the software. Nevertheless, it is possible to carry out highly accurate attacks without being detected. This thought experiment becomes very real if the attacker has sufficient resources, for example, to redirect the data directly at the upstream provider of a victim. The goal here is to infect a single AS and not as many ASs as possible. This makes an attack much more difficult and not as effective. But is reduces the probability that an RRC will receive and store this message. Thus, one focus for the future will be to clarify how these targeted attacks can be warded off and how the data can best be secured against espionage.

6 Historic BGP data

This chapter will focus on how historic BGP data can be retrieved and then processed. With this knowledge, we can reproduce historic BGP attacks, for example. For this purpose, we will take a look at the software BGPstream and how to download data with it. Then how to process the data using Artemis.

Practical tests will be used to find out how is the functionality of this method and the results will be presented. The intensity lies in the fact that we can look at old data sets with new software tools, such as Artemis. Thus, we can apply new analysis techniques to the data, collected in the past, and detect attacks/patterns that were not possible to detect in the past.

6.1 Data download

To use BGP data in Artemis, one has to download the data using a Python script. This script uses BGPstream, or more precisely the Python library of it called PyBGPStream. The necessary Python script is stored in the Artemis software.

Since the script still used PyBGPStream v1, it was rewritten in the context of this master thesis. As a result, PyBGPStream v2 is now used. This change was proposed via a GitHub pull request for inclusion in the Artemis software. This request was accepted. The new version of this script can now be downloaded from the GitHub page of Artemis (https://github.com/FORTH-ICS-INSPIRE/artemis/blob/master/other/bgpstream_retrieve_prefix_records.py) or is automatically downloaded during a new installation.

Figure 15 shows the framework of BGPStream. The interface to PyBGPStream visible here is then used to access the data providers via libBGPStream. The database from RIPE RIS und Route Views from chapter 2.5 will be use here. libBGPStream accesses the meta-data and downloads the relevant data with the information and analyses it using the pre-defined filters.

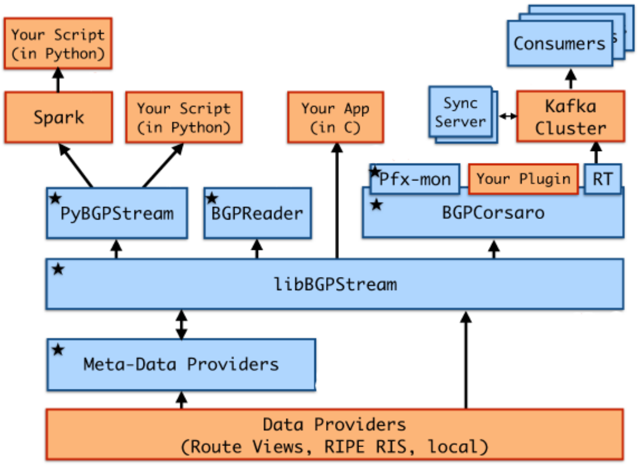


Figure 15: BGPStream framework overview (blue = framework; orange = project or placeholders) [31]

When using the script, the following settings have proven to be useful:

- Ubuntu 20 Container
- CPU: 2 to 4
- RAM: 20 GB
- Disk: 20 GB

In the context of this master thesis, the containers were operated using Proxmox 6.4.

The script that comes with Artemis has been adapted to download the historic BGP data. Since long-term data should be evaluated here, the script was adapted so that the data is always downloaded for one day (24h BGP data). Afterwards, the script automatically jumps to the next day. Additionally, all error messages are saved, and a log file is created. This contains a timestamp of when the download was started/completed for the individual day and what date it was.

The Python script is configured to save the data in the correct format for Artemis (format show in chapter 6.2). Example messages in the correct format are shown in the Table 13 (appendix).

6.2 Artemis historic BGP interface

Artemis provides an interface that allows to load historic BGP data. Here a CSV file with the following format must be read in:

```
<prefix>|<origin_asn>|<peer_asn>|<blank_separated_as_path>|<project>|<collector>|<update_type_A_o
r_W>|<bgpstream_community_json_dump>|<timestamp>
```

In cooperation with the BGPStream interface, data from the RIPE RIS and Route Views databases can be loaded into the software. These are then processed by Artemis. Thus also past

BGP hijacking attacks can be analysed. A variation of the configuration can also be tested with the same data set to analyse how the software reacts.

To use the evaluation of historic BGP data in Artemis, the function must be activated according to the instructions (https://bgpartemis.readthedocs.io/en/latest/history/). The directory that is stored in the docker-compose.yaml file can look like this:

```
- /home/bgp_messages/:/tmp/bgp_message/
```

The first part (green) is the path on the computer and the second part (blue) is in the Artemis Docker container. The second path is then set in the configuration on the Artemis user interface website.

6.3 Findings from the data download

The log files from the python script allow an analysis to be made of how long the query took. This can then be used to make an approximate estimate of how long a download will take. However, this statement is affected by a strong variation, as not all information can be determined. This applies especially to the PyBGPStream interface. Here it cannot be determined whether a slow download is due to a high workload or other factors.
For this evaluation, the following settings were made in the BGPstream data retrieval:

- Prefix:193.17.240.0/21
- Unix start time:1080604800
- Unix end time: 1402876800
- Filter: any
- RRC: all
- Time interval per request in Unix: 86400

If we now look at Figure 16 we can see a graph that shows the duration of the data retrieval per day. As the slope increases, the duration of the data retrieval increases. The colour markings can also be found on Figure 17, Figure 18, Figure 19 and are used for orientation/comparison points. Figure 17 shows the network load, which is not relevant factor for a long download/processing time, because the container runs within the THM (Technische Hochschule Mittelhessen) network and therefore a higher network load would be possible. Figure 18 and Figure 19 show the memory and CPU utilisation. This was adjusted during the data retrieval. Interestingly, however, it can be seen from the data that the highest retrieval time took place when the system resources were not fully utilised. The gap in the records is due to a temporary break, because at that time the resources of the server were used for other purposes.

The following conclusion can be drawn from this data. The retrieval of long-term data using BGPstream requires considerable time and resources. It remains to be seen what the reason for the high retrieval time is. It may be due to the Python library PyBGPStream or to the infrastructure of the data source. The most likely assumption is that the BGP router network will grow over time and that there will be more data to process. However, this question could not be clarified conclusively within in this master's thesis. The current assumption is that it was due to the I/O load caused by swapping.

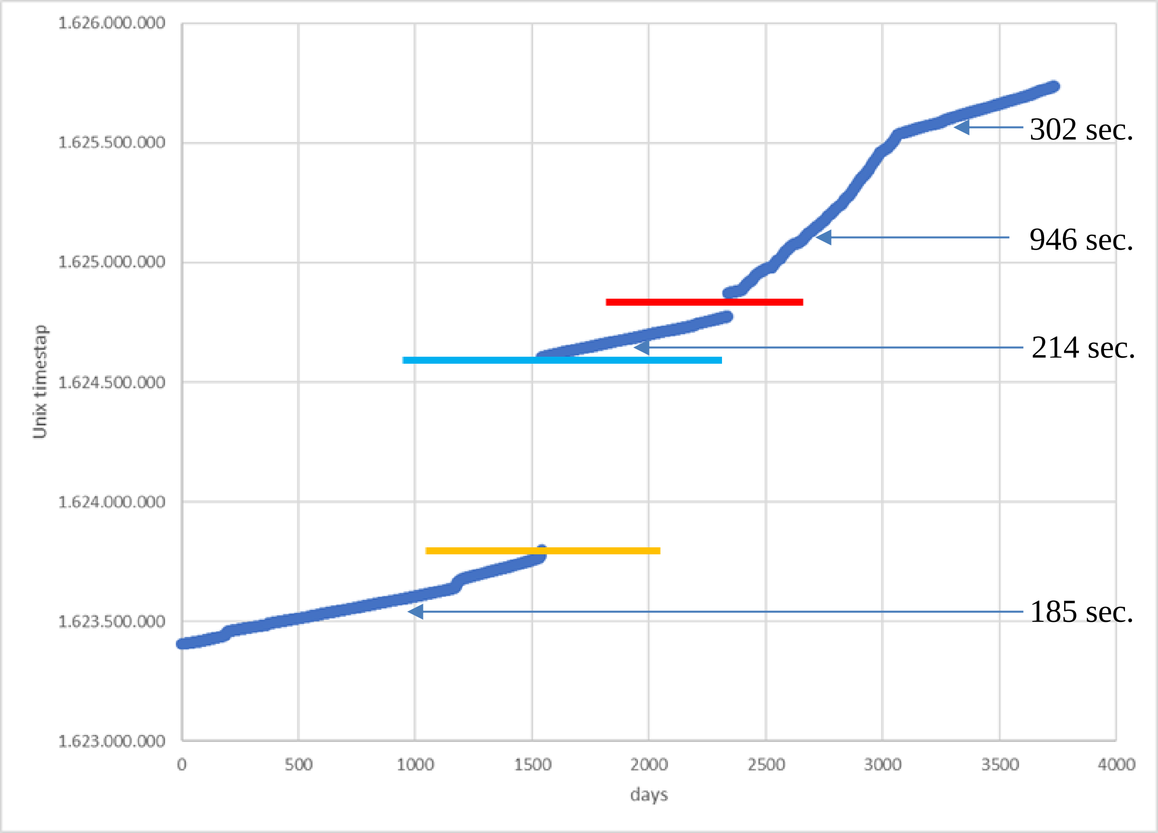


Figure 16: BGPStream retrieval time from BGPstream Linux container

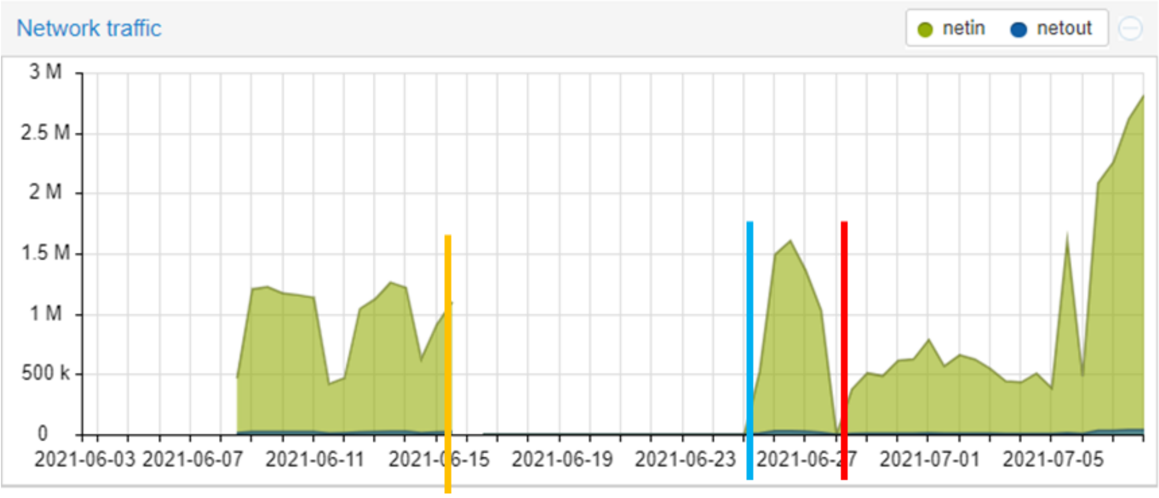


Figure 17: Network traffic from BGPstream Linux container

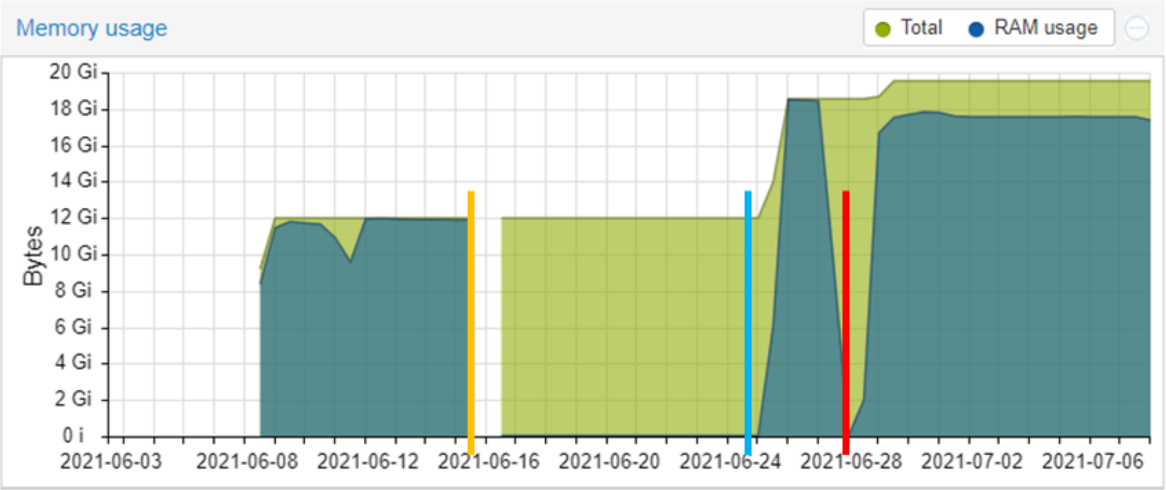


Figure 18: Memory usage from BGPstream Linux container

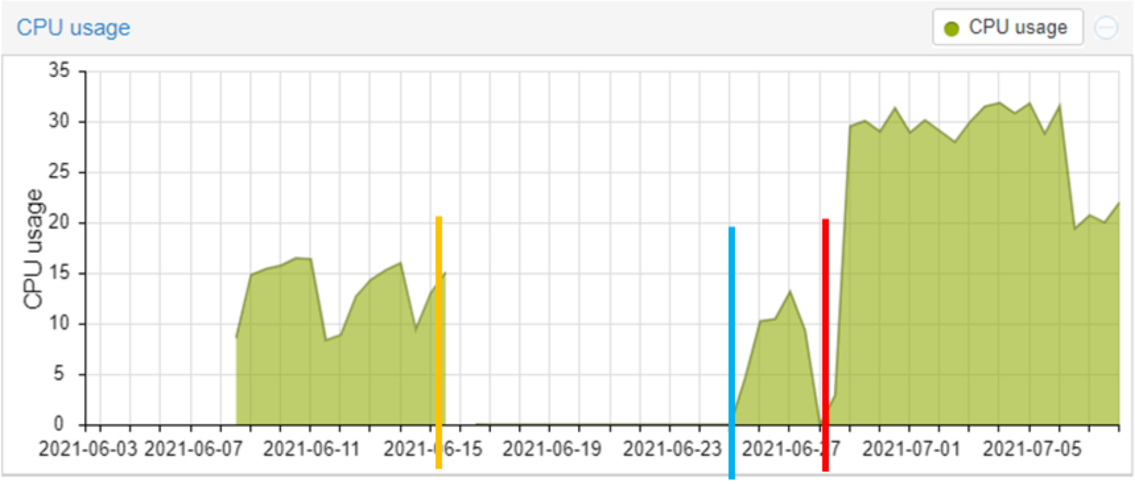
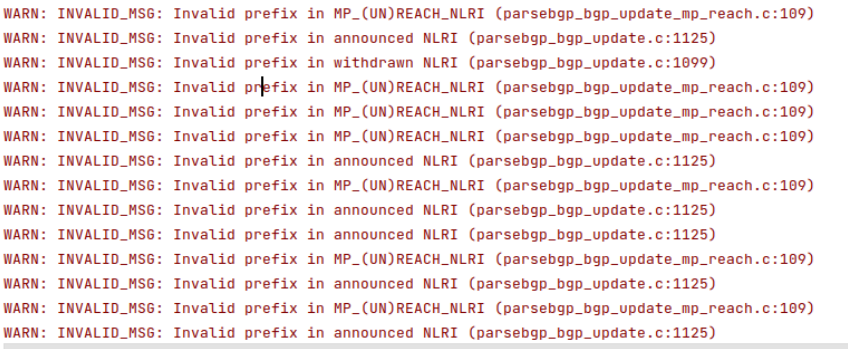


Figure 19: CPU usage from BGPstream Linux container

A bug in the PyBGPStream tool was also detected during data retrieval. In certain query configurations, messages with the prefix 0.0.0.0/0 are found after the data retrieval. PyBGPStream also generates a large number of error messages in another query configuration. An overview of the findings can be found in the Table 14 (appendix).

It has been shown that the retrieval of data from 1584489600 (Unix) onwards no longer works completely with filter "any". Otherwise, messages with the prefix 0.0.0.0/0 appear in the result file. Furthermore, the software generates error messages (Figure 20) from time 1590019200 (Unix) onwards when retrieving the data. This is then independent of the prefix filter. This examination (Table 14) was carried out on 09.06.2021. A repeat of the test from 9.6.2021 took place on 11.6.2021 and produced the same result. It can therefore be currently ruled out that the problem shifts in time, and instead starts at a fixed point in time. The problem is reported as error #218 at CAIDA/libbgpstream (GitHub).



```
WARN: INVALID_MSG: Invalid prefix in MP_(UN)REACH_NLRI (parsebgp_bgp_update_mp_reach.c:109)
WARN: INVALID_MSG: Invalid prefix in announced NLRI (parsebgp_bgp_update.c:1125)
WARN: INVALID_MSG: Invalid prefix in withdrawn NLRI (parsebgp_bgp_update.c:1099)
WARN: INVALID_MSG: Invalid prefix in MP_(UN)REACH_NLRI (parsebgp_bgp_update_mp_reach.c:109)
WARN: INVALID_MSG: Invalid prefix in MP_(UN)REACH_NLRI (parsebgp_bgp_update_mp_reach.c:109)
WARN: INVALID_MSG: Invalid prefix in MP_(UN)REACH_NLRI (parsebgp_bgp_update_mp_reach.c:109)
WARN: INVALID_MSG: Invalid prefix in announced NLRI (parsebgp_bgp_update.c:1125)
WARN: INVALID_MSG: Invalid prefix in MP_(UN)REACH_NLRI (parsebgp_bgp_update_mp_reach.c:109)
WARN: INVALID_MSG: Invalid prefix in announced NLRI (parsebgp_bgp_update.c:1125)
WARN: INVALID_MSG: Invalid prefix in announced NLRI (parsebgp_bgp_update.c:1125)
WARN: INVALID_MSG: Invalid prefix in MP_(UN)REACH_NLRI (parsebgp_bgp_update_mp_reach.c:109)
WARN: INVALID_MSG: Invalid prefix in announced NLRI (parsebgp_bgp_update.c:1125)
WARN: INVALID_MSG: Invalid prefix in MP_(UN)REACH_NLRI (parsebgp_bgp_update_mp_reach.c:109)
WARN: INVALID_MSG: Invalid prefix in announced NLRI (parsebgp_bgp_update.c:1125)
```

Figure 20: Section of the error message for PyBGPStream

6.4 Historic BGP data analysis

The result of the evaluation of the previously collected historic BGP data was positive. Using the data and the Artemis software, it was possible to find BGP messages that can be classified as BGP hijacking.

The BGP data analysed are data relating to the BSI (Federal Office for Information Security/ Bundesamt für Sicherheit in der Informationstechnik). The BSI has its own AS with the number 49234. All data run via the two upstream providers AS 680 (German Research Network/ Deutsche Forschungsnetz) and AS3320 (Deutsche Telekom). If another AS is seen in the first hop, this suggests a Type-1 hijacking.

The BSI announces three address ranges. The prefix 77.87.224.0/21 is announced once completely and additionally with 77.87.224.0/22 and 77.87.228.0/22. Furthermore, the prefixes

193.24.128.0/18 and 193.17.240.0/21 are announced. A special case is the prefix 193.17.240.0/23, which is announced via the German Research Network.

A further investigation was then carried out more comprehensively. Here, long-term data was downloaded using BGPStream. The following time ranges were considered in the long-term analysis:

- 77.87.224.0/21
    - Start time: Friday, 31. August 2007 00:00:00
    - End time: Friday, 26. February 2021 17:59:59
- 193.24.128.0/18
    - Start time: Wednesday, 20. February 2013 00:00:00
    - End time: Monday, 2. September 2019 23:59:59
- 193.17.240.0/21
    - Start time: Tuesday, 30. March 2004 00:00:00
    - End time: Thursday, 12. September 2019 23:59:59

At the following Unix times there were errors in the data retrieval, so no statement can be made for these days:

- 1282435200 (Sunday, 22. August 2010)
- 1442188800 (Monday, 14. September 2015)
- 1484611200 (Tuesday, 17. January 2017) – 1485388800 (Thursday, 26. January 2017)
- 1491696000 (Sunday, 9. April 2017)
- 1537488000 (Friday, 21. September 2018)
- 1537747200 (Monday, 24. September 2018)

A further evaluation over the year 2019 was not made, as the workload exceeded the benefit. Data retrieval was aborted almost every data day. By reducing the time span per programme run from one day down to one hour, a better and more performant download could be achieved. This allowed the download of the data to be continued. For time reasons, this finding could only be applied to the prefix 77.87.224.0/21.

The analysis of historic BGP data (Table 5 and Table 6) has shown that the AS 174 has a sub prefix (more exact prefix) as first hop announced (on 2017-11-1 13:42:31). Only the AS 680 and AS 3320 are official upstream providers of AS49234. This is clear evidence of an attack or misconfiguration. The data also shows that the attack lasted about four hours. In combination with the fact that AS 174 is a Tier 1 provider, very large amounts of data were probably passed through it. What is interesting is that only three RRCs recorded the hijacking. Because the BGP router owner is a Tier 1, one can assume that normally more than three RRC should be affected. When other routers take over the false information. From this, one can conclude that either the error was limited (possibly due to protection mechanisms) or that it was a targeted attack.

In this case, the attack could be picked up by one RRC from Ripe RIS and two from Route Views. However, it shows that a high RRC number is necessary to see as many attacks as possible. The assumption is proven by the discovered hijackings from the year 2020. There were several exact prefix hijackings (Table 8). Twice by AS 48237 (Mobily - Etihad Etisalat Company) and three times by AS 39386 (Saudi Telecom Company). This was only detected by one RRC of Route Views (Table 7). The attacks lasted several hours and had only a limited impact, as only a few RRCs took up the attack.

This demonstrates the importance of using as many data sources as possible. In this case, only Artemis would have seen this attack, compared to BGPalerter. Since BGPalerter only uses RIPE Live and the information from Route Views is missing there. This shows that in the case of BGP hijacking tools like Artemis or BGPalerter, Artemis has a better data base and thus potentially a larger coverage of the worldwide BGP data traffic.

The following peculiarities could be identified:

A= Announcement

W = Withdrawn

implicit-withdrawal = same AS as at the announcement

Table 5 BGP Message with abnormalities in 2017 (long-term analysis)

Timestamp	Prefix	Matched Prefix	Origin AS	AS Path	Peer AS	Service	BGP Type
2017-11-1 14:42:31	77.87.224.0/23	77.87.224.0/22	49234	37497, 174, 49234	37497	historical -> routeviews -> route-views.jinx	A
2017-11-1 14:42:44	77.87.224.0/23	77.87.224.0/22	49234	37497, 174, 49234	37497	historical -> routeviews -> route-views.linx	A
2017-11-1 15:10:59	77.87.224.0/23	77.87.224.0/22	49234	37497, 174, 49234	37497	historical -> ris -> rrc19	A
2017-11-1 18:50:40	77.87.224.0/23	77.87.224.0/22				historical -> routeviews -> route-views.linx	W
2017-11-1 14:42:31	77.87.226.0/23	77.87.224.0/22	49234	37497, 174, 49234	37497	historical -> routeviews -> route-views.jinx	A
2017-11-1 14:42:44	77.87.226.0/23	77.87.224.0/22	49234	37497, 174, 49234	37497	historical -> routeviews -> route-views.linx	A

2017-11-1 15:10:59	77.87.226.0/23	77.87.224.0/22	49234	37497, 174, 49234	37497	historical -> ris -> rrc19	A
2017-11-1 18:50:40	77.87.226.0/23	77.87.224.0/22			37497	historical -> routeviews -> route-views.linx	W

Table 6 Artemis recognised attacks in 2017 (long-term analysis)

Last Update/ Time Ended	Time Started	Hijacked Prefix	Matched Prefix	Type	Hijacker AS	# Peers Seen	# ASes Infected
2017-11-1 17:50:40	2017-11-1 13:42:31	77.87.224.0/23	77.87.224.0/22	S|1|-|-	174	1	1
2017-11-1 17:50:40	2017-11-1 13:42:31	77.87.226.0/23	77.87.224.0/22	S|1|-|-	174	1	1

Table 7 BGP Message with abnormalities in 2020

Timestamp	Prefix	Matched Prefix	Origin AS	AS Path	Peer AS	Service	BGP Type
2020-10-8 08:58:48	77.87.224.0/22	77.87.224.0/22	49234	35313, 51375, 48237, 49234	35313	historical -> routeviews -> route-views.linx	A
2020-10-8 12:25:07	77.87.224.0/22	77.87.224.0/22	------		35313	implicit-withdrawal	W

2020-10-11 15:19:05	77.87.228.0/22	77.87.228.0/22	49234	35313, 51375, 48237, 49234	35313	historical -> routeviews -> route-views.linx	A
2020-10-11 18:48:10	77.87.228.0/22	77.87.228.0/22	------		35313	implicit-withdrawal	W
2020-10-13 08:48:45	77.87.224.0/22	77.87.224.0/22	49234	35313, 51375, 39386, 49234	35313	historical -> routeviews -> route-views.linx	A
2020-10-13 11:10:49	77.87.224.0/22	77.87.224.0/22	------		35313	implicit-withdrawal	W
2020-11-11 08:23:05	77.87.224.0/22	77.87.224.0/22	49234	3513, 51375, 39386, 49234	35313	historical -> routeviews -> route-views.linx	A
2020-11-11 10:28:47	77.87.224.0/22	77.87.224.0/22	------		35313	implicit-withdrawal	W
2020-11-11 20:22:49	77.87.224.0/22	77.87.224.0/22	49234	35313, 51375, 39386, 49234	35313	historical -> routeviews -> route-views.linx	A

Table 8 Artemis recognised attacks

Last Update/ Time ended	Time Started	Hijacked Prefix	Matched Prefix	Type	Hijacker AS	# Peers Seen	# ASes Infected
2020-10-8 12:25:07	2020-10-8 07:58:48	77.87.224.0/22	77.87.224.0/22	E|1|-|-	48237	1	2

2020-10-11 18:48:10	2020-10-11 14:19:05	77.87.228.0/22	77.87.228.0/22	E|1|-|-	48237	1	2
2020-10-13 11:10:49	2020-10-13 08:48:45	77.87.224.0/22	77.87.224.0/22	E|1|-|-	39386	1	2
2020-11-11 10:28:47	2020-11-11 08:23:05	77.87.224.0/22	77.87.224.0/22	E|1|-|-	39386	1	2
2020-11-11 20:22:49	2020-11-11 20:22:49	77.87.224.0/22	77.87.224.0/22	E|1|-|-	39386	1	2

6.5 Practical test of detection capability

In order to check whether Artemis has implemented the promised functions in a functional way, a test setup was created. A test file (Table 13 in appendix) was created to see if Artemis can detect the most important attacks. The result show in Table 9. The test data was fed into the system via the Historic BGP interface (chapter: 6.2). Since the data was loaded via the "Historical BGP data" interface, no statement can be made here about the live data feed interfaces.

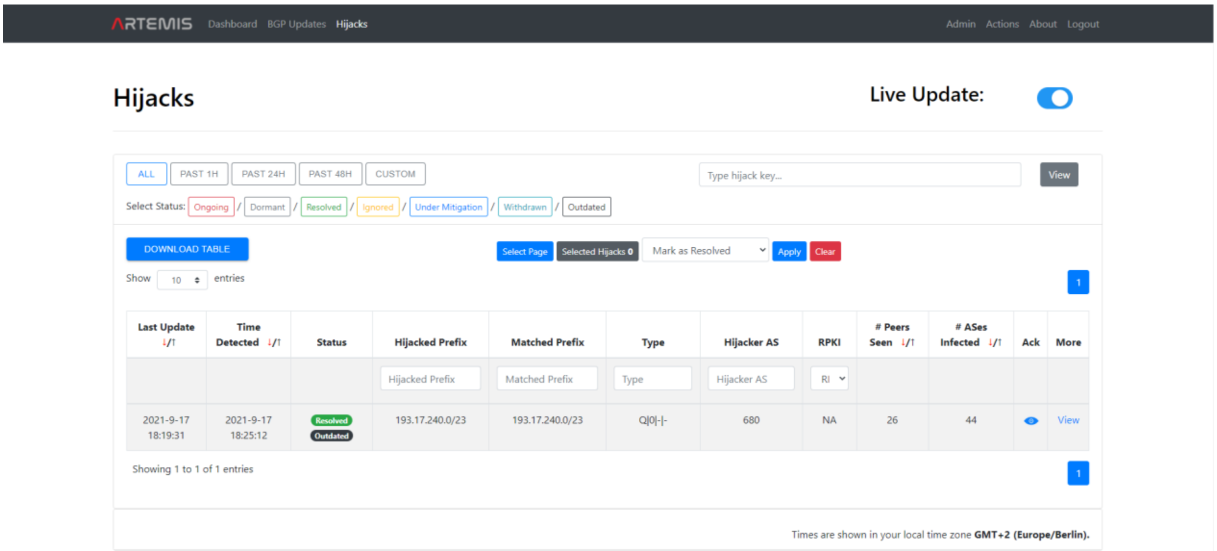
For the squatting attack, the config file has been modified so that Artemis declares a normally announced prefix as squatting. Thus, it could be established that following attacks can be detected:

Table 9 Artemis' proven detection capabilities

Artemis BGP hijacking classification (visible in the Artemis GUI)	Description	Hijacking Type after definition (chapter 3)
S|0|-|-	Could detect a sub prefix hijacking from a fake origin	Type 0; sub prefix
S|1|-|-	Could detect a sub prefix hijacking from a fake first hop	Type 1; sub prefix
E|0|-|-	Could detect an exact prefix hijacking from a fake origin	Type 0; exact prefix
E|1|-|-	Could detect an exact prefix hijacking from a fake first hop	Type 1; exact prefix
Q|0|-|-	Could detect a squatting attack. To achieve this, the configuration was adjusted to declare a normally announced prefix as stolen.	Squatting
RPKI:	The system has detected that the ROA is valid during a simulated squatting attack. Thus, it can be assumed that RPKI detection also works in other cases.	--------

Table 9 shows that Artemis can recognise Type 0 (sub and exact prefix) and 1 (sub and exact prefix). Furthermore, the detection of squatting could also be verified. The RPKI validation via GitHub NLnetLabs/routinator also worked.

In the Figure 21 a squatting attack (Q|0|-|-) has been simulated. The interface shows the most important data. The attack can be further analysed by selecting the corresponding incident. Here (Figure 22) all basic data like attacked prefix, start and end time are displayed. Furthermore, all BGP messages are saved where the filter has detected a contradiction.



Hijacks

Live Update:

Select Status: Ongoing / Dormant / Resolved / Ignored / Under Mitigation / Withdrawn / Outdated

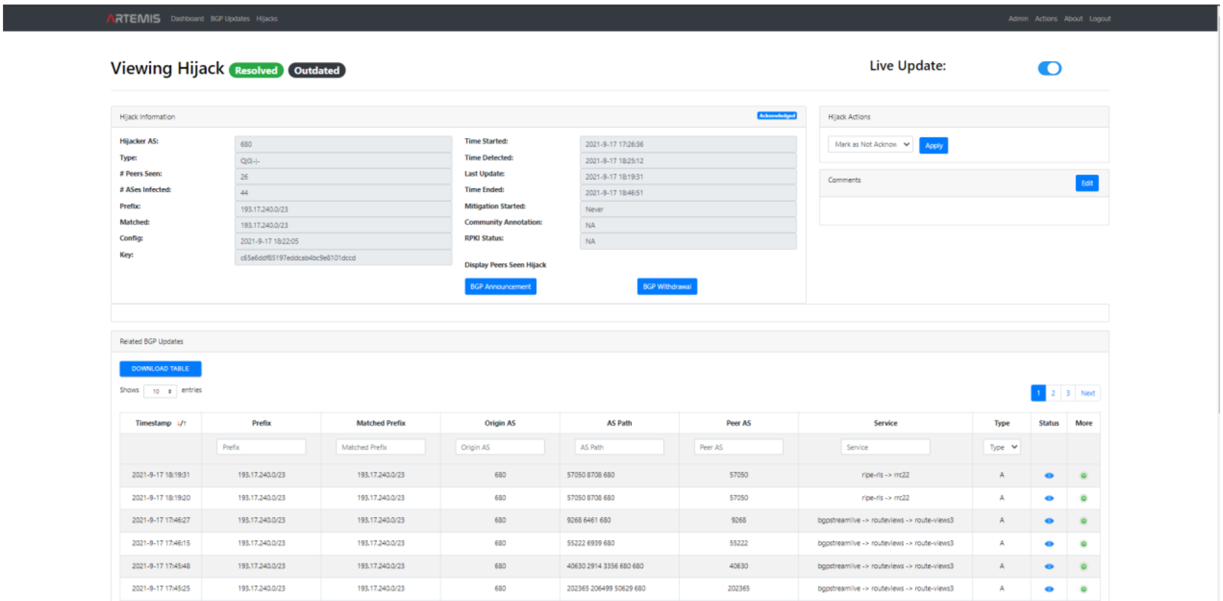
Show 10 entries

Last Update ↓/↑	Time Detected ↓/↑	Status	Hijacked Prefix	Matched Prefix	Type	Hijacker AS	RPKI	# Peers Seen ↓/↑	# ASes Infected ↓/↑	Ack	More
			Hijacked Prefix	Matched Prefix	Type	Hijacker AS	RI				
2021-9-17 18:19:31	2021-9-17 18:25:12	Resolved Outdated	193.17.240.0/23	193.17.240.0/23	Q|0|-|-	680	NA	26	44		View

Showing 1 to 1 of 1 entries

Times are shown in your local time zone **GMT+2 (Europe/Berlin).**

Figure 21: Artemis GUI with BGP hijacking example



Viewing Hijack Resolved Outdated

Live Update:

Hijack Information

Hijacker AS:	680	Time Started:	2021-9-17 17:26:36
Type:	Q|0|-|-	Time Detected:	2021-9-17 18:25:12
# Peers Seen:	26	Last Update:	2021-9-17 18:19:31
# ASes Infected:	44	Time Ended:	2021-9-17 18:46:51
Prefix:	193.17.240.0/23	Mitigation Started:	Never
Matched:	193.17.240.0/23	Community Annotation:	NA
Config:	2021-9-17 18:22:05	RPKI Status:	NA
Key:	c65a6ddf85197eddcab4bc9e8101dccd		

Display Peers Seen Hijack

Related BGP Updates

Shows 10 entries

Timestamp ↓/↑	Prefix	Matched Prefix	Origin AS	AS Path	Peer AS	Service	Type	Status	More
	Prefix	Matched Prefix	Origin AS	AS Path	Peer AS	Service	Type		
2021-9-17 18:19:31	193.17.240.0/23	193.17.240.0/23	680	57050 8708 680	57050	ripe-ris -> rrc22	A		
2021-9-17 18:19:20	193.17.240.0/23	193.17.240.0/23	680	57050 8708 680	57050	ripe-ris -> rrc22	A		
2021-9-17 17:46:27	193.17.240.0/23	193.17.240.0/23	680	9268 6461 680	9268	bgpstreamlive -> routeviews -> route-views3	A		
2021-9-17 17:46:15	193.17.240.0/23	193.17.240.0/23	680	55222 6939 680	55222	bgpstreamlive -> routeviews -> route-views3	A		
2021-9-17 17:45:48	193.17.240.0/23	193.17.240.0/23	680	40630 2914 3356 680 680	40630	bgpstreamlive -> routeviews -> route-views3	A		
2021-9-17 17:45:25	193.17.240.0/23	193.17.240.0/23	680	202365 206499 50629 680	202365	bgpstreamlive -> routeviews -> route-views3	A		

Figure 22: Artemis hijack detail view

Artemis also offers the possibility to forward the alarm by mail. However, this is a simple mail with an alarm message in JSON format. If necessary, a custom tool must be used to process this message in the mailbox to improve readability. An alert mail content generated by the squatting test looks like this:

```
messaging - 2021-09-17 16:25:12,151 - INFO @ _receive_callback:
{"key":"c65a6ddf85197eddcab4bc9e8101dccd","asns_inf":[24961,61218],"time_start
ed":1631892396.0000970364,"timestamp_of_config":1631895725.9926617146,"time_la
st":1631892396.0000970364,"type":"Q|0|-|-
","hijack_as":680,"prefix":"193.17.240.0\/23","end_tag":null,"outdated_parent"
:null,"peers_seen":[61218],"community_annotation":"NA","time_detected":1631895
912.1490185261,"configured_prefix":"193.17.240.0\/23","hijack_url":"https:\/\/
artemis.com\/main\/hijack?key=c65a6ddf85197eddcab4bc9e8101dccd"}
```

Unlike what is currently described in the manual (https://bgpartemis.readthedocs.io/en/latest/loggingconf/), only the file local_configs/backend/logging.yaml must be edited during configuration. If errors occur, one can try to evaluate them with the command "docker-compose logs -f".

As part of the investigation, BGPalerter was also tested in a container. Here, the tests are limited to the handling and the investigation of how the automatic configuration creation works. In this context, it was found that BGPalerter generates an alarm when a new prefix is announced. This was detected by AS49234 announcing the new prefix 193.30.80.0/24. This function can therefore be confirmed.

6.6 Analysis of RPKI in case of BGP hijacking

On 29.07.2021 there was a major disruption in the Telekom network [58]. It turned out that a foreign AS (AS212416; 41 hijacked prefixes) announced addresses from various providers. This incident is now to be dealt with here in more detail to demonstrate the advantages of RPKI.

The data set in Table 12 (appendix) was determined using RIPE Stat.

The first finding from the data is that with a super prefix attack RPKI has no effect. In this case, there were four prefixes (data set no. 1,2,25,35) that were announced as super prefixes by the hijacker AS. Of these, all four have more specific prefixes announced by their owners. Again, two of these also use RPKI. Despite this, 155 peers have adopted the information, which corresponds to the maximum value in the entire data set. Thus, it can be concluded that RPKI does not apply if there is no use of RPKI for an exact or more specific prefix. In the case of such an attack, the information has now been taken over, but has no effect as long as the more specific prefixes continue to be announced. Because there is the basic rule in BGP that prefer announcements with a more specific prefix.

Now we come to the exact and sub prefix hijacking. Here, the use of RPKI becomes visible as follows. The maximum measured peering number, which has taken over the false announcement, for a prefix with RPKI is 106 (data set no. 17) of the maximum data set value 155. The relation between prefixes without RPKI below this peering value compared to above it is 4:8. This shows that RPKI could achieve an improvement.

This incident has shown that BGP hijacking is a real threat and RPKI is a way to make the system more secure. It is important to remember that this technique only helps against Type-0 hijacking. Furthermore, it has been shown that it is essential to achieve the widest possible use of RPKI. Here, the network administrators are still holding back [22]. This means that the widespread use of false announcements is still possible. Furthermore, no further investigations were carried out to find out which security measures were still in place. Thus, it cannot be ensured that other mechanisms outside RPKI have falsified the results.

The hijacking incident of 29.07.2021, which was carried out by AS212416 raises further questions. There are other security mechanisms besides RPKI, such as filters (chapter 2.3) or counters that limit the maximum number of attacks. These possibilities are not related to the BGP protocol and may be additional functions in a BGP router. The question now arises why the upstream provider did not take action against this incident. Since the AS in question has only two neighbours (AS174 and AS57344), it is probably a Tier 3. If both upstream providers had used RPKI, this attack would have been prevented. This shows that a small ISP could attack large companies. The reasons for this could be worked out in a future work. One hypothesis that can be explored is that ISPs in small or insignificant Internet countries have an easier time hijacking. This assumption is based on the idea that in the countries concerned, small ISPs are more directly connected to large ISPs. This means that the simple network structure in such countries means that there are no more intermediate stations and thus fewer possibilities for control. The large ISPs will probably be connected to Tier-1 providers despite their low importance. Here it is questionable to what extent filters and counters are used to avoid interfering with data traffic. In this way, small ISPs will probably have very direct and unrestricted access to worldwide traffic.

7 Summary and Conclusion

In the master's thesis, various aspects around the topic of BGP hijacking were to be investigated and evaluated. It turned out that there was no standardised classification of BGP hijacking. For this reason, a classification was developed here that is based on the classification of previous scientific work and combine them in the best possible way. In the best case, this can lead to a worldwide uniform definition for BGP hijacking.

In the next part of the work, the possibilities for prevention, reaction and analysis were shown. It has been shown in the thesis that there are established methods, especially for prevention and reaction. Some of these were practical and others more theoretical. Attack analysis like Pings or RIPE Atlas probes, however, is still in its early stages. The first possibilities have already been investigated in scientific papers. However, it will probably take some time before these methods (Pings and RIPE Atlas probes) are available with ease of use.

Another focus of the work is the comparison between Artemis and BGPalerter. Both are software tools for detecting BGP hijacking. The software tools are both open source and on premise. The analysis showed that Artemis offers more advantages when it comes to protection against attacks because of a larger data supply. BGPalerter, on the other hand, offers better functions for monitoring one's own AS and prefixes, like monitoring the announcement and availability of the AS. Both software can detect a Type-0 and 1 hijacking attack with exact or sub prefix. BGPalerter already has the ability to detect Type-N hijacking via manual configuration; however, this will only really be usable if Artemis or BGPalerter can do it automatically. Since a frequent adjustment of the routes via BGP can be normal and there a manual Type-N configuration is not useful.

Working with BGPStream has shown the extensive possibilities of filtering large amounts of data for BGP analysis. Using BGPstream and Artemis, it is possible to generate and analyse historic data. Here, long-term data for the prefixes announced by the BSI were examined. In the process, a false announcement could be found. It has been shown that the generation of long-term data using PyBGPStream takes a considerable amount of time. This high time requirement could not be determined conclusively and requires a more detailed investigation of BGPStream. It has been shown, however, that this can be useful, since an attack has been detected and could now be analysed using Artemis. The open questions from this work on BGPStream can be further elaborated. Furthermore, a test interface could be built for

BGPalerter, as it is currently only possible to load own data into Artemis via the interface for historic data.

In addition to the scientific findings, improvements and insights for the open-source tools were also achieved in the course of this master's thesis. For example, the programme for the historic data download at Artemis was revised so that it can use BGPStream v2. Furthermore, anomalies in BGPstream were investigated and passed on to the developers.

RPKI was evaluated in more detail on a real hijacking. The result was that RPKI would probably have prevented this attack completely if it had been used by the hijacker's upstream provider.

References

[1] M. Holland, “BGP-Hijacking: Massive Internet-Störungen im Festnetz der Telekom,” *heise online*, 29 Jul., 2021. https://www.heise.de/news/Internet-Massive-Probleme-im-Festnetz-der-Telekom-6150438.html (accessed: Oct. 20 2021).

[2] C. Martinho, “Understanding How Facebook Disappeared from the Internet,” *The Cloudflare Blog*, 04 Oct., 2021. https://blog.cloudflare.com/october-2021-facebook-outage/ (accessed: Oct. 20 2021).

[3] W. Schulte, *Handbuch der Routing-Protokolle: Eine Einführung in RIP, IGRP, EIGRP, HSRP, VRRP, OSPF, IS-IS und BGP*. Berlin, Offenbach: VDE Verlag GmbH, 2016.

[4] T. Keary, “Types of Routing Protocols – The Ultimate Guide,” *Comparitech*, 28 Nov., 2018 (accessed: Oct. 20 2021).

[5] *IDRP-Protokoll :: interdomain routing protocol (IDRP) :: ITWissen.info* (accessed: Oct. 20 2021).

[6] *BGP (border gateway protocol) :: BGP-Protokoll :: ITWissen.info*. [Online]. Available: https://www.itwissen.info/BGP-border-gateway-protocol-BGP-Protokoll.html (accessed: May 5 2021).

[7] S. Luber, “Was ist das Border Gateway Protocol (BGP)?,” *IP-Insider*, 05 Mar., 2019. https://www.ip-insider.de/was-ist-das-border-gateway-protocol-bgp-a-804823/ (accessed: Sep. 17 2021).

[8] *ARTEMIS - an Open-Source Tool for Detecting BGP Prefix Hijacking in Real-Time*. [Online]. Available: https://bgpartemis.org/ (accessed: Jul. 2 2021).

[9] Cisco, *BGP-Algorithmus für die beste Pfadauswahl* (accessed: Oct. 23 2021).

[10] K. Butler, T. R. Farley, P. McDaniel, and J. Rexford, “A Survey of BGP Security Issues and Solutions,” *Proc. IEEE*, vol. 98, no. 1, pp. 100–122, 2010, doi: 10.1109/JPROC.2009.2034031.

[11] G. Huston, M. Rossi, and G. Armitage, “Securing BGP — A Literature Survey,” *IEEE Commun. Surv. Tutorials*, vol. 13, no. 2, pp. 199–222, 2011, doi: 10.1109/SURV.2011.041010.00041.

[12] *What is RPKI? — RIPE Network Coordination Centre.* [Online]. Available: https://www.ripe.net/manage-ips-and-asns/resource-management/rpki/what-is-rpki (accessed: Aug. 9 2021).

[13] R. Fedler, “Prefix Hijacking-Angriffe und Gegenmaßnahmen,” 2012. [Online]. Available: https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2012-08-1/NET-2012-08-1_01.pdf

[14] *Titel: Beta Version of the RPKI RTR Client C Library Released | RIPE Labs* (accessed: Oct. 20 2021).

[15] Internet Society, *BGPSec - A reality now | Internet Society.* [Online]. Available: https://www.internetsociety.org/blog/2017/10/bgpsec-reality-now/ (accessed: Aug. 9 2021).

[16] RIPE Labs, *BGP Communities - A Weapon for the Internet (Part 1).* [Online]. Available: https://labs.ripe.net/author/florian_streibelt/bgp-communities-a-weapon-for-the-internet-part-1/ (accessed: Aug. 3 2021).

[17] Noction, “The dark side of BGP community,” *Noction*, 30 Nov., 2020. https://www.noction.com/blog/bgp-community-attributes (accessed: Aug. 3 2021).

[18] R. Mangelmann, “DE-CIX Blackholing Service,” [Online]. Available: https://www.de-cix.net/_Resources/Persistent/4/d/5/f/4d5f5d57cb3a466d34ea4d640961353f309ca6b3/DE-CIX%20Blackholing%20service.pdf

[19] *Der Wilde Westen im Internet: BGP-Communities.* [Online]. Available: https://www.mpg.de/12592211/mpiinf_jb_2018 (accessed: Jul. 9 2021).

[20] *rfc8092.* [Online]. Available: https://datatracker.ietf.org/doc/html/rfc8092 (accessed: Aug. 2 2021).

[21] R. Mangelmann, “PowerPoint-Präsentation,” [Online]. Available: https://www.de-cix.net/_Resources/Persistent/4/d/5/f/4d5f5d57cb3a466d34ea4d640961353f309ca6b3/DE-CIX%20Blackholing%20service.pdf

[22] P. Sermpezis, V. Kotronis, A. Dainotti, and X. Dimitropoulos, “A Survey among Network Operators on BGP Prefix Hijacking,” Jan. 2018. [Online]. Available: http://arxiv.org/pdf/1801.02918v1

[23] *NIST RPKI Monitor.* [Online]. Available: https://rpki-monitor.antd.nist.gov/ROV (accessed: Sep. 8 2021).

[24] *Routing Information Service (RIS) — RIPE Network Coordination Centre.* [Online]. Available: https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris (accessed: May 13 2021).

[25] *RIS Raw Data — RIPE Network Coordination Centre.* [Online]. Available: https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-raw-data (accessed: May 13 2021).

[26] RIPE Labs, *RIPE NCC Technical Services 2017 - Part Three: Focus on Tools and Research.* [Online]. Available: https://labs.ripe.net/author/kranjbar/ripe-ncc-technical-services-2017-part-three-focus-on-tools-and-research/ (accessed: Jun. 15 2021).

[27] *RIS Live — RIPE Network Coordination Centre.* [Online]. Available: https://ris-live.ripe.net/ (accessed: May 17 2021).

[28] *FAQ – Routeviews.* [Online]. Available: http://www.routeviews.org/routeviews/index.php/faq/ (accessed: May 13 2021).

[29] *RouteViews Collector Map – Routeviews.* [Online]. Available: http://www.routeviews.org/routeviews/index.php/map/ (accessed: May 13 2021).

[30] *Route Views Archive Project Page.* [Online]. Available: http://archive.routeviews.org/ (accessed: Aug. 10 2021).

[31] C. Orsini, A. King, D. Giordano, V. Giotsas, and A. Dainotti, “BGPStream,” in *IMC'16: Proceedings of the 2016 Internet Measurement Conference : November 14-16, 2016, Santa Monica, CA, USA*, Santa Monica California USA, 2016, pp. 429–444. Accessed: May 18 2021. [Online]. Available: https://bgpstream.caida.org/bundles/caidabgpstreamwebhomepage/pubs/bgpstream.pdf

[32] GitHub, *Exa-Networks/exabgp.* [Online]. Available: https://github.com/Exa-Networks/exabgp (accessed: Jul. 7 2021).

[33] RIPE Labs, *ExaBGP - A new Tool to Interact with BGP.* [Online]. Available: https://labs.ripe.net/author/thomas_mangin/exabgp-a-new-tool-to-interact-with-bgp/ (accessed: Jul. 7 2021).

[34] Khin Thida Latt, Yasuhiro Ohara, Satoshi Uda and Yoichi Shinoda, “Analysis of IP Prefix Hijacking and Traffic Interception,” [Online]. Available: http://paper.ijcsns.org/07_book/201007/20100704.pdf

[35] P. Sermpezis *et al.*, “ARTEMIS: Neutralizing BGP Hijacking Within a Minute,” *IEEE/ACM Trans. Networking*, vol. 26, no. 6, pp. 2471–2486, 2018, doi: 10.1109/TNET.2018.2869798.

[36] *Hijack Information - Artemis Docs.* [Online]. Available: https://bgpartemis.readthedocs.io/en/latest/hijackinfo/ (accessed: Aug. 11 2021).

[37] “Stealing The Internet An Internet-Scale Man In The Middle Attack Defcon 16, Las Vegas, NV - August 10th, 2008,” [Online]. Available: https://we.riseup.net/assets/43591/defcon-16-pilosov-kapela.pdf

[38] *What is BGP hijacking? | Cloudflare.* [Online]. Available: https://www.cloudflare.com/de-de/learning/security/glossary/bgp-hijacking/ (accessed: Jul. 9 2021).

[39] *YouTube Hijacking: A RIPE NCC RIS case study — RIPE Network Coordination Centre.* [Online]. Available: https://www.ripe.net/publications/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study (accessed: Aug. 11 2021).

[40] P. Sermpezis, V. Kotronis, K. Arakadakis, and A. Vakali, “Estimating the Impact of BGP Prefix Hijacking,” May. 2021. [Online]. Available: http://arxiv.org/pdf/2105.02346v1

[41] *Was ist ein autonomes System? | Was sind ASNs? | Cloudflare.* [Online]. Available: https://www.cloudflare.com/de-de/learning/network-layer/what-is-an-autonomous-system/ (accessed: Oct. 23 2021).

[42] Johannes Zirngibl, Patrick Sattler, Markus Sosnowski, and Georg Carle, “HEAP BGP Observatory,” [Online]. Available: https://www.caida.org/workshops/kismet/2002/slides/kismet2002_jzirngibl.pdf

[43] *Coverage and Statistics | RIPE Atlas.* [Online]. Available: https://atlas.ripe.net/results/maps/network-coverage/ (accessed: Jun. 30 2021).

[44] *RIPE Atlas - The Credit System | Docs.* [Online]. Available: https://beta-docs.atlas.ripe.net/getting-started/credits.html (accessed: Jul. 14 2021).

[45] L. Mohit, M. Dan, P. Dan, W. Yiguo, Z. Beichuan, and Z. Lixia, “PHAS: A Prefix Hijack Alert System,” *15th USENIX Security Symposium.* [Online]. Available: https://www.usenix.org/legacy/events/sec06/tech/full_papers/lad/lad.pdf

[46] YouTube, *PHAS - A Prefix Hijack Alert System.* [Online]. Available: https://www.youtube.com/watch?v=fkcJDsG92UU (accessed: Sep. 16 2021).

[47] J. Schlamp, R. Holz, Q. Jacquemart, G. Carle, and E. W. Biersack, “HEAP: Reliable Assessment of BGP Hijacking Attacks,” *IEEE J. Select. Areas Commun.*, vol. 34, no. 6, pp. 1849–1861, 2016, doi: 10.1109/JSAC.2016.2558978.

[48] *Routenüberwachung | BGPmon.* [Online]. Available: https://bgpmon.net/services/route-monitoring/ (accessed: Oct. 23 2021).

[49] GitHub, *ANSSI-FR/tabi: BGP Hijack Detection.* [Online]. Available: https://github.com/ANSSI-FR/tabi (accessed: Sep. 2 2021).

[50] GitHub, *FORTH-ICS-INSPIRE/artemis.* [Online]. Available: https://github.com/FORTH-ICS-INSPIRE/artemis/tags (accessed: Jul. 13 2021).

[51] RIPE Labs, *ARTEMIS: an Open-source Tool for Detecting BGP Prefix Hijacking in Real Time.* [Online]. Available: https://labs.ripe.net/author/vasileios_kotronis/artemis-an-open-source-tool-for-detecting-bgp-prefix-hijacking-in-real-time/ (accessed: Jul. 5 2021).

[52] GitHub, *nttgin/BGPalerter.* [Online]. Available: https://github.com/nttgin/BGPalerter/blob/main/docs/installation.md (accessed: Jul. 6 2021).

[53] Massimo Candela, “Easy BGP monitoring with BGPalerter,” [Online]. Available: https://www.lacnic.net/innovaportal/file/4489/1/bgpalerter_lacnic33.pdf

[54] GitHub, *BGPalerter/configuration.md at main · nttgin/BGPalerter.* [Online]. Available: https://github.com/nttgin/BGPalerter/blob/main/docs/configuration.md#monitorhijack (accessed: Aug. 12 2021).

[55] GitHub, *BGPalerter/configuration.md at main · nttgin/BGPalerter.* [Online]. Available: https://github.com/nttgin/BGPalerter/blob/main/docs/configuration.md (accessed: Jul. 30 2021).

[56] GitHub, *FORTH-ICS-INSPIRE/artemis.* [Online]. Available: https://github.com/FORTH-ICS-INSPIRE/artemis#minimum-technical-requirements (accessed: Jul. 7 2021).

[57] *Community Annotations - Artemis Docs.* [Online]. Available: https://bgpartemis.readthedocs.io/en/latest/commannotations/ (accessed: Aug. 31 2021).

[58] Martin Holland, “BGP-Hijacking: Massive Internet-Störungen im Festnetz der Telekom,” [Online]. Available: https://www.heise.de/news/Internet-Massive-Probleme-im-Festnetz-der-Telekom-6150438.html

[59] *Collectors – Routeviews.* [Online]. Available: http://www.routeviews.org/routeviews/index.php/collectors/ (accessed: Jun. 15 2021).

Appendix

Table 10 Ripe RIS (Data access May 2021) [25]

RCC name	Data access	Total Peering	Recording start/end and exchange point	Location
RRC 00	http://data.ris.ripe.net/rrc00	135	Oct 1999 (RIPE Region)	Amsterdam, NL
RRC 01	http://data.ris.ripe.net/rrc01	149	July 2000 (LINX) Mar 2018 (LONAP)	London, GB
RRC 02	http://data.ris.ripe.net/rrc02	40	Mar 2001 until Oct 2008 (SFINX)	Paris, FR
RRC 03	http://data.ris.ripe.net/rrc03	151	Jan 2001 (AMS-IX and NL-IX) Jan 2001 until July 2015 (GN-IX)	Amsterdam, NL
RRC 04	http://data.ris.ripe.net/rrc04	20	Apr 2001 (CIXP)	Geneva, CH
RRC 05	http://data.ris.ripe.net/rrc05	62	June 2001 (VIX)	Vienna, AT
RRC 06	http://data.ris.ripe.net/rrc06	8	Aug 2001 (JPIX)	Otemachi, JP
RRC 07	http://data.ris.ripe.net/rrc07	36	Apr 2002 (NETNOD)	Stockholm, SE
RRC 08	http://data.ris.ripe.net/rrc08	----	May 2002 until Sep 2004 (MAE-WEST)	San Jose, US
RRC 09	http://data.ris.ripe.net/rrc09	----	May 2003 until Feb 2004 (TXI)	Zurich, CH
RRC 10	http://data.ris.ripe.net/rrc10/	67	Nov 2003 (MIX)	Milan, IT
RRC 11	http://data.ris.ripe.net/rrc11/	46	Feb 2004 (NYIIX)	New York, US

RRC 12	http://data.ris.ripe.net/rrc12/	154	Jul 2004 (DE-CIX)	Frankfurt, DE
RRC 13	http://data.ris.ripe.net/rrc13/	34	Apr 2005 (MSK-IX)	Moscow, RU
RRC 14	http://data.ris.ripe.net/rrc14/	30	Dec 2004 (PAIX)	Palo Alto, US
RRC 15	http://data.ris.ripe.net/rrc15/	63	Dec 2005 (PTTMetro-SP)	Sao Paulo, BR
RRC 16	http://data.ris.ripe.net/rrc16/	35	Feb 2008 (NOTA)	Miami, US
RRC 18	http://data.ris.ripe.net/rrc18/	26	Nov 2015 (CATNIX)	Barcelona, ES
RRC 19	http://data.ris.ripe.net/rrc19/	63	Jan 2016 (NAP Africa JB)	Johannesburg, ZA
RRC 20	http://data.ris.ripe.net/rrc20/	73	Nov 2015 (SwissIX)	Zurich, CH
RRC 21	http://data.ris.ripe.net/rrc21/	73	Nov 2015 (FranceIX)	Paris, FI
RRC 22	http://data.ris.ripe.net/rrc22/	38	Jan 2018 (Interlan)	Bucharest, RO
RRC 23	http://data.ris.ripe.net/rrc23/	39	Jan 2018 (Equinix SG)	Singapore, SG
RRC 24	http://data.ris.ripe.net/rrc24/	26	Feb 2019 (LACNIC)	Montevideo, UY
RRC 25	http://data.ris.ripe.net/rrc25/	135	Feb 2021 (RIPE NCC)	Amsterdam, NL

Table 11 Route Views Collectors (Data access May 2021) [59]

Host	Software	BGP Proto	Recording start/end	UI	Location and exchange point
route-views.routeviews.org	Cisco	IPv4 uni/multi-cast multi-hop	-----	telnet	U of Oregon, Eugene Oregon, USA
route-views2.routeviews.org	Quagga	IPv4 uni/multi-cast multi-hop	Oct 2001	telnet	U of Oregon, Eugene Oregon, USA
route-views3.routeviews.org	FRR	IPv4 uni/multi-cast multi-hop	Apr 2007	telnet	U of Oregon, Eugene Oregon, USA
route-views4.routeviews.org	Quagga	IPv4/IPv6 uni/multi-cast multi-hop	Nov 2008	telnet	U of Oregon, Eugene Oregon, USA
route-views6.routeviews.org	Zebra	IPv6 multi-hop	May 2003	telnet	U of Oregon, Eugene Oregon, USA
route-views.amsix.routeviews.org	FRR	IPv4/v6 uni/multi-cast non-multi-hop	Jul 2018	telnet	AMS-IX AM6 - Amsterdam IX
route-views.chicago.routeviews.org	FRR	IPv4/v6 uni/multi-cast non-multi-hop	Jun 2016	telnet	Equinix CH1 - Chicago, IL USA
route-views.chile.routeviews.org	FRR	IPv4/v6 uni/multi-cast non-multi-hop	Jan 2018	telnet	Santiago, Chile

route-views.eqix.routeviews.org	FRR	IPv4/v6 uni/multi-cast non-multi-hop	May 2004	telnet	Equinix, Ashburn, VA
route-views.flix.routeviews.org	FRR	IPv4/v6 uni/multi-cast non-multi-hop	Jan 2018	telnet	FL-IX, Atlanta, Georgia
route-views.fortaleza.routeviews.org	FRR	IPv4/v6 uni/multi-cast non-multi-hop	May 2019	telnet	IX.br (PTT.br), Fortaleza, Brazil
route-views.gixa.routeviews.org	FRR	IPv4/v6 uni/multi-cast non-multi-hop	May 2019 – Feb 2020 & Mar 2021 -	telnet	GIXA, Ghana, Africa
route-views.gorex.routeviews.org	FRR	IPv4/v6 uni/multi-cast non-multi-hop	Aug 2019	telnet	GOREX, Guam, US Territories
route-views.isc.routeviews.org	Zebra	IPv4/v6 uni/multi-cast non-multi-hop	Nov 2003	telnet	ISC (PAIX), Palo Alto CA, USA
route-views.jinx.routeviews.org	Quagga	IPv4/v6 uni/multi-cast non-multi-hop	Jul 2012 – Aug 2019	telnet	Johannesburg, South Africa
route-views.kixp.routeviews.org	Zebra	IPv4 uni/multi-cast non-multi-hop	Oct 2005	telnet	KIXP, Nairobi, Kenya

route-views.linx.routeviews.org	FRR	IPv4/v6 uni/multi-cast non-multi-hop	Mar 2004	telnet	LINX, London, GB
route-views.napafrica.routeviews.org	FRR	IPv4/v6 uni/multi-cast non-multi-hop	Feb 2018	telnet	NAPAfrica, Johannesburg, South Africa
route-views.nwax.routeviews.org	Zebra	IPv4/v6 uni/multi-cast non-multi-hop	Mar 2014	telnet	NWAX, Portland, Oregon
route-views.perth.routeviews.org	Quagga	IPv4/v6 uni/multi-cast non-multi-hop	Nov 2012	telnet	West Australian Internet Exchange, Perth
route-views.phoix.routeviews.org	FRR	IPv4/v6 uni/multi-cast non-multi-hop	Aug 2019	telnet	University of the Philippines, Diliman, Quezon City
route-views.sfmix.routeviews.org	FRR	IPv4/6 uni/multi-cast non-multi-hop	Apr 2015	telnet	San Francisco Metro IX - San Francisco, CA USA
route-views.rio.routeviews.org	FRR	IPv4/6 uni/multi-cast non-multi-hop	Apr 2019	telnet	IX.br (PTT.br), Rio de Janeiro, Brazil
route-views.sydney.routeviews.org	Quagga	IPv4/6 uni/multi-cast non-multi-hop	Aug 2010	telnet	SYDNEY (SYD1 Equinix), Sydney, Australia
route-views.soxrs.routeviews.org	Quagga	IPv4/6 uni/multi-cast non-multi-hop	Aug 2013	telnet	Serbia Open Exchange, Belgrade Serbia

route-views.sg.routeviews.org	Zebra	IPv4/6 uni/multi-cast non-multi-hop	May 2014	telnet	SG1 Equinix Singapore
route-views.saopaulo.routeviews.org	Zebra	IPv4/6 uni/multi-cast non-multi-hop	Mar 2011	telnet	SAOPAULO (PTT Metro, NIC.br), Sao Paulo, Brazil
route-views2.saopaulo.routeviews.org	FRR	IPv4/6 uni/multi-cast non-multi-hop	Apr 2018	telnet	SAOPAULO (PTT Metro, NIC.br), Sao Paulo, Brazil
route-views.telxatl.routeviews.org	Zebra	IPv4/6 uni/multi-cast non-multi-hop	Feb 2012	telnet	TELXATL (TELX Atlanta), Atlanta, Georgia
route-views.wide.routeviews.org	Zebra	IPv4/6 uni/multi-cast non-multi-hop	Jul 2003	telnet	DIXIE (NSPIXP), Tokyo, Japan
route-views.mwix.routeviews.org	FRR	IPv4/6 uni	Feb 2018	telent	Indiana, USA
route-views.bdix.routeviews.org	FRR		Apr 2021	telent	(BDIX) Bangladesh

Table 12 BGP hijacking from AS212416 at 29.07.2021

No.	Attack type	Origin-Prefix	AS-Origin	Prefix announce by AS-hijacker (AS212416)	RPKI	Peers infected	Contact Mail	Note
1	Super-Prefix			5.59.64.0/20	No	155	abuse@coprosys.cz	Super Prefix; 5.59.64.0/22 is announced by AS202813 no ROA
2	Super-Prefix			5.59.80.0/20	No	155	abuse@coprosys.cz	Super Prefix; 5.59.80.0/22 is announced by AS204004 no ROA
3	Exact-Prefix	5.182.56.0/23	35142	5.182.56.0/23	No	15	abuse-blx@betterlinx.co.il	
4	Exact-Prefix	46.11.0.0/16	15735	46.11.0.0/16	No	56	abuse@go.com.mt	
5	Sub-Prefix	46.11.0.0/16	15735	46.11.88.0/21	No	155	abuse@go.com.mt	
6	Sub-Prefix	79.158.0.0/16	3352	79.158.103.0/24	No	153	nemesys@telefonica.es	
7	Sub-Prefix	80.91.64.0/19	174	80.91.93.0/24	No	119	abuse@cogentco.com	

8	Exact-Prefix	80.128.0.0/11	3320	80.128.0.0/11	Yes	20	auftrag@nic.telekom.de	
9	Exact-Prefix	80.128.0.0/12	3320	80.128.0.0/12	Yes	20	auftrag@nic.telekom.de	
10	Sub-Prefix	81.35.0.0/16	3352	81.35.3.0/24	No	153	nemesys@telefonica.es	
11	Exact-Prefix	85.217.135.0/24	200845/43160	85.217.135.0/24	Yes	26	sistemas@avatel.es	
12	Exact-Prefix	85.217.138.0/24	200845/43160	85.217.138.0/24	Yes	26	sistemas@avatel.es	
13	Exact-Prefix	87.128.0.0/10	3320	87.128.0.0/10	Yes	20	auftrag@nic.telekom.de	
14	Exact-Prefix	87.228.144.0/20	6866	87.228.144.0/20	Yes	16	abuse@cytanet.com.cy	
15	Sub-Prefix	88.26.0.0/16	3352	88.26.195.0/24	No	153	nemesys@telefonica.es	
16	Exact-Prefix	88.98.100.0/22	200845	88.98.100.0/22	Yes	26	sistemas@avatel.es	

17	Sub-Prefix	88.98.96.0/20	43160	88.98.108.0/22	Yes	106	sistemas@avatel.es	
18	Exact-Prefix	88.98.112.0/20	202147	88.98.112.0/20	Yes	29	report@vozplus.com	
19	Exact-Prefix	88.98.120.0/21	202147	88.98.120.0/21	Yes	78	report@vozplus.com	
20	Sub-Prefix	90.192.0.0/11	5607	90.218.57.0/24	Yes	104	abuse@sky.uk	
21	Exact-Prefix	91.245.200.0/21	202147	91.245.200.0/21	Yes	29	report@vozplus.com	
22	Exact-Prefix	92.58.104.0/22	12479	92.58.104.0/22	No	66	abuse@orange.es	
23	Sub-Prefix	93.192.0.0/10	3320	93.254.66.0/24	Yes	104	abuse@telekom.de	
24	Sub-Prefix	93.192.0.0/10	3320	93.254.130.0/24	Yes	104	abuse@telekom.de	
25	Super-Prefix			109.67.0.0/16	No	155	abuse@bezeqint.net	Super Prefix; 109.67.0.0/18 is announced by AS8551 with ROA

26	Exact-Prefix	109.67.80.0/24	8551	109.67.80.0/24	Yes	30	abuse@bezeqint.net	
27	Exact-Prefix	109.110.243.0/24	35432	109.110.243.0/24	Yes	104	abuse@cablenetcy.net	
28	Sub-Prefix	137.74.0.0/16	16276	137.74.106.0/24	No	153	abuse@ovh.net	
29	Sub-Prefix	141.95.0.0/17	16276	141.95.2.0/24	No	153	abuse@ovh.net	
30	Exact-Prefix	141.237.0.0/16	3329	141.237.0.0/16	Yes	31	ettn_pbn.gr@vodafone.com	
31	Exact-Prefix	141.255.0.0/17	3329	141.255.0.0/17	Yes	31	ettn_pbn.gr@vodafone.com	
32	Exact-Prefix	154.43.167.0/24	174	154.43.167.0/24	No	119	abuse@cogentco.com	
33	Sub-Prefix	178.194.0.0/15	3303	178.195.240.0/24	Yes	104	abuse@bluewin.ch	

34	Exact-Prefix	178.208.192.0/19	8301	178.208.192.0/19	Yes	20	abuse@gibtele.com	
35	Super-Prefix			185.49.168.0/22	No	155	abuse@olivenet.es	Super Prefix; 185.49.168.0/24 is announced by AS201746 with ROA
36	Exact-Prefix	185.51.108.0/22	202147	185.51.108.0/22	Yes	29	report@vozplus.com	
37	Exact-Prefix	185.94.48.0/22	200845	185.94.48.0/22	Yes	26	sistemas@avatel.es	
38	Exact-Prefix	185.205.252.0/22	205262	185.205.252.0/22	Yes	10	abuse@conred.es	
39	Exact-Prefix	188.241.96.0/21	202147	188.241.96.0/21	Yes	31	report@vozplus.com	
40	Exact-Prefix	212.63.116.0/22	43160	212.63.116.0/22	Yes	31	sistemas@avatel.es	
41	Exact-Prefix	212.170.0.0/16	3352	212.170.0.0/16	No	56	nemesys@telefonica.es	

Table 13 Excerpt BGP test messages for Artemis

1. BGP hijack: exact prefix with fake origin AS
2. BGP hijack: sub prefix with fake origin AS
3. BGP hijack: exact prefix with fake first hop AS
4. BGP hijack: sub prefix with fake first hop AS
5. Announce an new prefix

```
1. 77.87.228.0/22|101|38883|38883 6939 680 101|routeviews|route-views4|A|"[{""asn"":38883,""value"":2005}]"|1605095273.0
2. 77.87.224.0/24|102|38883|38883 6939 680 102|routeviews|route-views4|A|"[{""asn"":38883,""value"":2005}]"|1605095273.0
3. 77.87.224.0/22|49234|38883|38883 6939 103 49234|routeviews|route-views4|A|"[{""asn"":38883,""value"":2005}]"|1605095273.0
4. 77.87.224.0/24|49234|328145|328145 37271 3356 680 104 49234|routeviews|route-views.napafrica|A|"[{""asn"":65214,""value"":4440}]"|1605096991.920742
5. 77.87.224.10/22|105|328145|328145 37271 3356 680 680 105|routeviews|route-views.napafrica|A|"[{""asn"":65214,""value"":4440}]"|1605096991.920742
```

Table 14 BGPstream data retrieval test series (from 09.06.2021)

Unix time	Date	Filter prefix	Filter Collector	Results
1583020800	1.3.'20	Any 77.87.224.0/21	all	no conspicuity
1584230400	15.3.'20	Any 77.87.224.0/21	all	no conspicuity
1584403200	17.3'20	Any 77.87.224.0/21	all	no conspicuity
1584489600	18.3'20	Any 77.87.224.0/21	all	Message with prefix: 0.0.0.0/0
1584576000	19.3.'20	Any 77.87.224.0/21	all	Message with prefix: 0.0.0.0/0
1584662400	20.3.'20	Any 77.87.224.0/21	all	Message with prefix: 0.0.0.0/0
1584748800	21.3.'20	Any 77.87.224.0/21	all	Message with prefix: 0.0.0.0/0
1585699200	1.4.'20	Any 77.87.224.0/21	all	Message with prefix: 0.0.0.0/0
1585699200	1.4.'20	More 77.87.224.0/21	all	no conspicuity
1588291200	1.5.'20	Any 77.87.224.0/21	all	Message with prefix: 0.0.0.0/0
1588291200	1.5.'20	More 77.87.224.0/21	all	no conspicuity
1589500800	15.5.'20	Any 77.87.224.0/21	all	Message with prefix: 0.0.0.0/0
1589500800	15.5.'20	More 77.87.224.0/21	all	no conspicuity
1589932800	20.5.'20	Any 77.87.224.0/21	all	Message with prefix: 0.0.0.0/0
1589932800	20.5.'20	Any 193.24.128.0/18	all	Message with prefix: 0.0.0.0/0
1589932800	20.5.'20	More 193.24.128.0/18	all	no conspicuity

1590019200	21.5.'20	Any 77.87.224.0/21	all	Error messages: invalid prefix; Message with prefix: 0.0.0.0/0
1590105600	22.5.'20	Any 77.87.224.0/21	all	Error message: invalid prefix; Message with prefix: 0.0.0.0/0
1590192000	23.5.'20	Any 77.87.224.0/21	all	Error messages: invalid prefix; Message with prefix: 0.0.0.0/0
1590364800	25.5.'20	Any 77.87.224.0/21	all	Error messages: invalid prefix; Message with prefix: 0.0.0.0/0
1590883200	31.5.'20	Any 77.87.224.0/21	all	Error message: invalid prefix; Message with prefix: 0.0.0.0/0
1590883200	31.5.'20	More 77.87.224.0/21	all	Error message: invalid prefix;
1590969600	1.6.'20	Any 77.87.224.0/21	all	Error messages: invalid prefix; Message with prefix: 0.0.0.0/0
1590969600	1.6.'20	More 77.87.224.0/21	all	Error message: invalid prefix;
1589500800	15.5.'20	Any 77.87.224.0/21	all	Message with prefix: 0.0.0.0/0