Université de Strasbourg

Master RISE
Réseaux informatiques et systèmes embarqués

Présenté par

Loïc Miller
loic.miller@etu.unistra.fr

DDoS, BGP leaks and hijack mitigation techniques

Encadré par

Cristel Pelsser, professeure des universités
Stéphane Cateloin, maître de conférences
{pelsser,cateloin}@unistra.fr

Au sein du laboratoire

Icube

Université de Strasbourg

Master RISE
Computer Networks and Embedded Systems

Presented by

Loïc Miller
loic.miller@etu.unistra.fr

DDoS, BGP leaks and hijack mitigation techniques

Mentored by

Prof. Cristel Pelsser
Associate Prof. Stéphane Cateloin
{pelsser,cateloin}@unistra.fr

Within

Icube

Résumé

Les cyberattaques sont de plus en plus présentes dans Internet. Avec la montée en puissance de l'Internet des Objets, le nombre d'objets connectés est en train d'exploser. La potentialité qu'un botnet puisse lancer des attaques de déni de service distribué prends des proportions effrayantes [92]. De nouveaux vecteurs d'attaques sont régulièrement découverts [1, 88], rendant possible des attaques d'une ampleur inégalée.

Même si des techniques de mitigation existent, leur efficacité et leur dynamique fonctionnement restent peu étudiées. Le blackholing est une de ces techniques, utilisant les communautés pour annoncer un besoin de mitigation. Si il n'est pas sécurisé, le blackholing peut devenir un vecteur d'attaques.

Dans ce rapport, nous détaillons plusieurs vecteurs d'attaques dans Internet, ainsi que les solutions pour prévenir ou mitiger les dégâts causés par ces attaques. Deuxièmement, nous déterminons une taxonomie des attaques utilisant le blackholing. Ensuite, nous décrivons de bonnes pratiques ainsi qu'une solution de sécurité contre ces attaques. De plus, nous concevons un outil qui permet de détecter les attaquants potentiels.

Abstract

Attacks are more present than ever in the Internet. With the rise of the Internet of Things (IoT), the number of connected devices is exploding. The potential of a botnet to launch massive Distributed Denial of Service attacks (DDoS) is taking scary proportions [92]. New attack vectors [1, 88] are being discovered and are enabling the largest attacks we have ever seen.

While techniques to mitigate DDoS attack exist, little is known about the efficiency of these techniques and their dynamics. Blackholing is one of them, using BGP communities as one of the vectors to announce the need for mitigation. If not secured, blackholing can be a vector of attacks.

In this report, we first review existing attack vectors in the Internet, and current solutions to prevent or mitigate damage caused by these attacks. In a second time, we determine a taxonomy of attacks using blackholing. Then, we describe good practices and a solution to secure against those attacks. In addition, we design a tool allowing the detection of potential attackers.

Table of contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The ICube laboratory

This internship was executed within the network research team of the ICube laboratory [58].

Created in 2013, ICube hosts around 650 members coming from the University of Strasbourg, the French National Center for Scientific Research (CNRS) [21], the ENGEES [31] and the INSA of Strasbourg [59], working together in the fields of engineering science and computer science.

The research component of ICube (Figure 1.1) is divided into four departments:

- The Computer science department
- The Imaging, robotics, remote sensing and biomedical department
- The Solid-state electronics, systems and photonics department
- The Department of mechanics

Those departments are themselves composed of multiple teams (16 overall) working on a wide variety of subjects, from computer science to civil engineering and medical imaging.

Managed by Prof. Pierre Gançarski, the Computer science department is split into six different teams:

- The Geometry and Computer Graphics teams (IGG)
- The Networks team
- The Scientific and Parallel Computing team (ICPS)
- The Data Science and Knowledge team (SDC)
- The Complex Systems and Translational Bioinformatics team (CSTB)
- The IMages, leArning, Geometry and Statistics team (IMAGeS)

1.1.1 The network research team

The network research team managed by Prof. Thomas Noel is composed of 11 faculty members, one engineer and six PhD students. It focuses on two research topics:

- The Internet of Things, which focuses on study and design of efficient algorithms and protocols for the interaction of constrained smart devices with the physical world.
- Core networks, which focuses on the study and design of topologies, algorithms and policies related to the way data is exchanged over computer networks.

The team is also involved in research projects such as FIT/IoT-LAB [36].

I was a part of the network research team for six months as an intern, as part of the final-year internship to complete my master's degree. In this context, I had the chance to work with Prof. Cristel Pelsser as well as Associate Prof. Stéphane Cateloin.

During this period, we focused our efforts on determining how blackholing, a technique used to mitigate Distributed Denial of Service attacks, could be used with malicious intent in routing attacks. We then focused on finding ways to avoid those attacks and building a tool capable of detecting potential attackers.

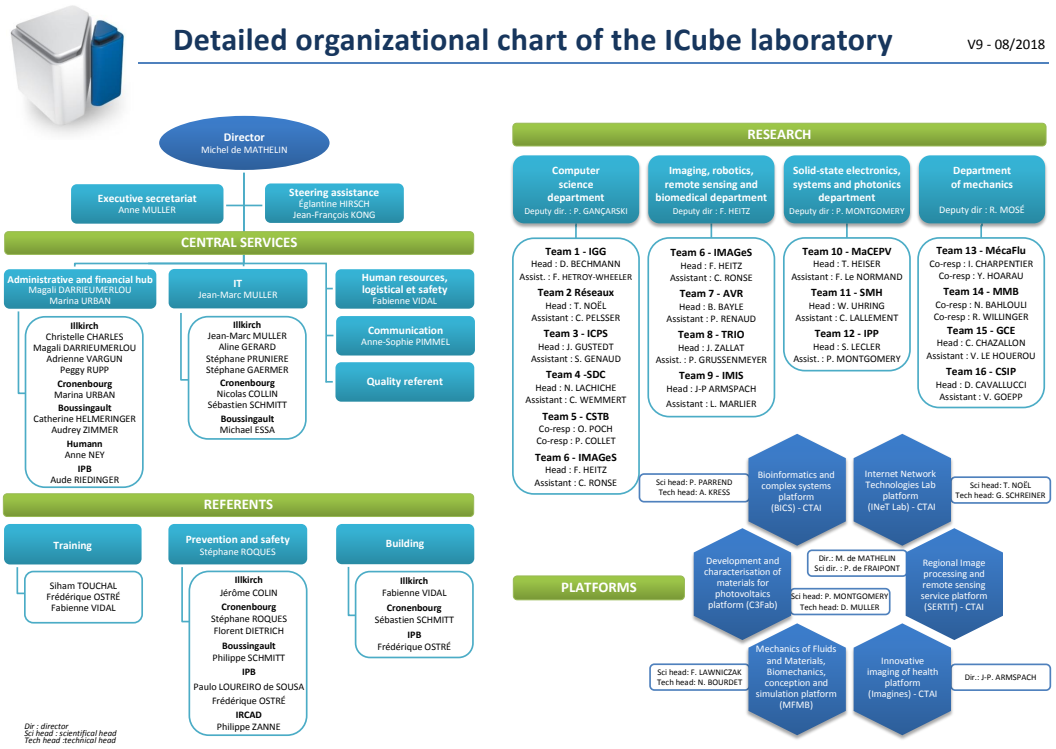


Figure 1.1: Detailed organizational chart of the ICube laboratory [57]

1.2 Routing in the Internet

The Internet is composed of multiple networks called **Autonomous Systems (ASes)**. These networks are interconnected by a routing protocol called the **Border Gateway Protocol (BGP)**, which provides and maintains reachability between hosts across those networks. In this section, we will present an overview of BGP and its intricacies, as well as the structure of the Internet in general.

1.2.1 The Border Gateway Protocol

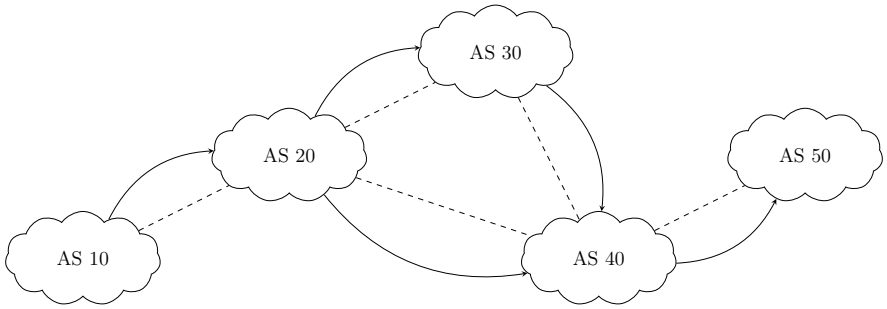
The Border Gateway Protocol [100] is the de-facto inter-domain routing protocol in the Internet. Its primary function is to allow ASes to communicate with others by exchanging reachability information. An AS has at least one **border router** (also called a **BGP speaker**) that can communicate with other ASes using BGP. Some ASes forward only their own traffic (stub ASes), while some provide services to other ASes to forward their traffic (transit ASes). Relationships between ASes will be detailed in Subsection 1.2.3.

BGP is a path vector protocol: every time a BGP speaker announces a **route** (i.e. a possible path to a given destination in the Internet), the advertisement saves the identification number (**AS Number (ASN)**) of each AS it went through in its **AS path** attribute.
Figure 1.2 illustrates this process. In Figure 1.2a, AS 10 announces a route to AS 20, which forwards it to AS 30 and AS 40 and so on. Table 1.2b shows the evolution of the AS path for this route. AS 40 must choose one of the two advertisements, and picks the one with the shortest AS path. In reality, this decision process is much more complex, and is detailed in Subsection 1.2.2. To join AS 10, AS 50 has a route with the AS path attribute set to "40 20 10". The rightmost AS in the AS path (here, AS 10) is the one who originated the **prefix**. This AS is called the **Origin AS**.

BGP messages

BGP speakers can send numerous types of messages to their peers:

- OPEN, used to establish a connection between two BGP speakers.
- UPDATE, used to exchange reachability information with another AS.
- KEEPALIVE, used to check if a BGP speaker is still reachable.
- NOTIFICATION, used to inform another BGP speaker that an error has occurred.



(a) Propagation of a BGP advertisement

AS Number	Received AS path	Sent AS path
AS 10	-	10
AS 20	10	20 10
AS 30	20 10	30 20 10
AS 40	20 10 30 20 10	40 20 10
AS 50	40 20 10	-

(b) Evolution of the AS path towards AS 10

Figure 1.2: BGP advertisement propagation and respective AS paths

UPDATE messages are used by BGP speakers to send and withdraw their routes. Each UPDATE message contains:

- A destination prefix (a block of **IP** addresses).
- BGP attributes bound to this prefix.

These BGP attributes include BGP **communities**.

BGP communities

BGP communities [17] are labels that can be used to induce a special behaviour for the announced routes, or pass additional information to BGP peers. The COMMUNITIES attribute is an optional and transitive set of four octet values, each of which specifies a community. Communities are usually encoded using an ASN in the first two octets, and a value in the last two octets, noted as *ASN:value*.

Some of the community values have global significance and should be treated appropriately as specified by [17]. Those communities are referred to as well-known communities:

- NO_EXPORT: Routes carrying this community must not be advertised outside a BGP **confederation**[1] boundary or outside a stand-alone AS.
- NO_ADVERTISE: Routes carrying this community must not be advertised to other BGP peers.
- NO_EXPORT_SUBCONFED: Routes carrying this community must not be advertised to **eBGP** peers (including inside a BGP confederation).

BGP speakers receiving a route may append or modify the COMMUNITIES attribute when propagating the advertisement to its peers. Extended communities [109] were subsequently defined to introduce an extended value range and provide a structure to communities with the addition of a Type field. Large communities [45] were then defined to account for four octet ASNs [122]. These communities are represented as an ASN and two operator-defined values, noted *ASN:value:value*.

A BGP speaker will send an UPDATE message every time it needs to announce a new route to a destination, if attributes bound to the route have changed, or if the BGP speaker can no longer reach a destination.

[1] AS divided into multiple internal sub-ASes to reduce **iBGP** mesh size, but still advertised as a single AS to external peers.

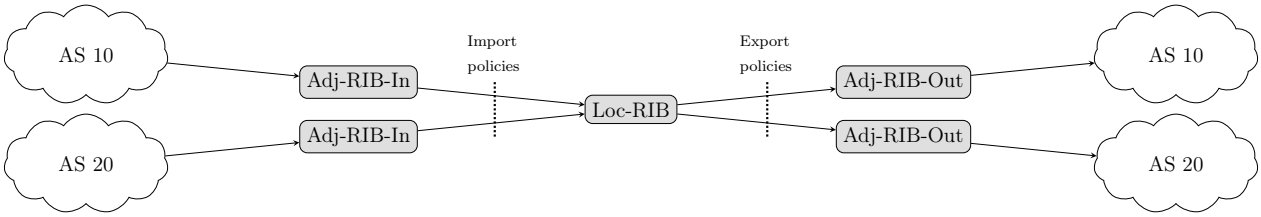


Figure 1.3: Routing information base of a BGP speaker

Step	Discriminant	
1	Highest local-pref	(economic considerations)
2	Shortest AS path	(performance)
3	Lowest origin type	(IGP < EGP < INCOMPLETE)
4	Lowest MED	(cold potato routing)
5	eBGP-learned over iBGP-learned	
6	Lowest IGP cost	(hot potato routing)
7	Oldest external path	(route-flap)
8	Lowest router-id	(arbitrary tie-break)

Table 1.1: Standard BGP route ranking process

1.2.2 BGP route selection

It is possible for an AS to receive multiple routes to the same destination, with different attributes. In that case, the AS must rank these routes and choose the best one according to its policies and the attributes of each route. This process takes several steps, which are described below.

A BGP speaker maintains all the reachability information in its **Routing Information Base (RIB)**, which is structured in three sets (Figure 1.3):

- **Adj-RIB-In**: routes learned from its neighbors.
- **Loc-RIB**: routes selected from Adj-RIB-In by applying import policies. This is the routing table used in the actual routing.
- **Adj-RIB-Out**: routes selected from Loc-RIB, which the router will announce to its neighbors. For each neighbor, the router creates a specific Adj-RIB-Out based on export policies.

If multiple routes to the same destination are validated by the routers' import policies, they will be ranked based on their attributes, and the best route will be stored in the Loc-RIB. A standard ranking process can be found in Table 1.1. The ranking process considers all routes step by step, and selects the first one being better than the others as best route. Once the best route has been determined, the router may export it to a peer based on its export policies (Figure 1.3). It is important to keep in mind that while an AS has fine control over routes it sends to its neighbors, it also has very few control over routes it receives (MED is the only discriminant and only indicates a preference).

It is also important to take into account the specificity of the prefix. Indeed, most ASes will not accept advertisments beyond a certain level of specificity (Section 6.1.3 of [30]). As of today, it seems that prefixes longer than /24 for **IPv4** and longer than /48 for **IPv6** are generally not announced nor accepted [113, 33].

1.2.3 AS relationships

As said earlier, the Internet is composed of multiple ASes (more than 60000 ASes advertised as of 2018-08-13 [112, 55]), which are interconnected by BGP. However, an AS cannot have a direct connection with every other AS. The Internet has a hierarchical structure. An AS can have peers, customers and providers.

Figure 1.4 shows a hierarchical representation of the Internet, with examples of some relationships between ASes. Relationships between ASes follow economical relationships: a stub AS may be the customer of a transit AS, which in turn may have a provider (customer-provider relationship). ASes can also transit each other's traffic for free, thus creating a peer-to-peer relationship. The ASes at the top of the hierarchy are called Tier one ASes. Tier one ASes do not have any providers, and are all peering with the other Tier one ASes in a full-mesh, to enable global connectivity in the Internet.

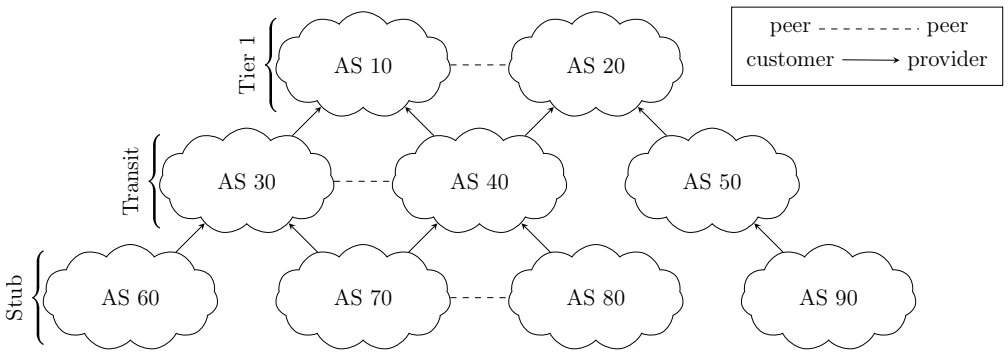


Figure 1.4: Hierarchical representation of the Internet

This now standard model of routing policies was developed by Gao and Rexford [37, 38]. The Gao-Rexford model supposes that all ASes apply the following:

- import: Prefer routes learned from a client (customer routes) over routers learned from a peer (peer routes). Prefer peer routes over routes learned from a provider (provider routes).
- export: Export customer routes to all neighboring ASes, and export peer or provider routes to customers only.

The import rule models ASes' incentives to send traffic along customer routes (which generate revenue), as opposed to peer routes (which do not generate revenue) or provider routes (which come at a monetary cost). The export rule models ASes' willingness to transit traffic from one neighbor to another only when paid to do so by a customer.

If every AS follows precisely those directives (and no AS is an indirect provider of itself), BGP convergence to a stable state is guaranteed. This set of policies induces a topological property on the AS path: after going through a provider-to-customer or a peer-to-peer link, the traffic cannot go through a customer-to-provider or a peer-to-peer link again. Those AS paths are called valley-free.

It is worth noting that while an AS is not obliged to strictly follow those rules, most of them do so [41] because they make economical sense: On Figure 1.4, if AS 30 exports routes it learned from its provider (AS 10) to its peer (AS 40), AS 30 takes the risk of AS 40 using those routes, thus paying for AS 40s' traffic.

1.2.4 Internet eXchange Points

Internet eXchange Points (IXPs) are infrastructures enabling ASes to exchange Internet traffic. Entities such as **Internet Service Providers (ISPs)** and **Content Delivery Networks (CDNs)** connect to IXPs to reduce transit costs and shorten distances to other networks. Instead of paying a provider for transit, a member can peer through an IXP, thus only paying the shared cost of the infrastructure.

While the most common type of peering at an IXP was bilateral (between two ASes), overhead and scaling issues led IXP members to multilateral interconnection. Multilateral interconnection (between three or more ASes) is achieved by means of a third-party brokering system: the **route server** [60]. Each IXP member announces its routes to the route server, which then forwards them to each other IXP member. The route server does not transit traffic itself, only reachability information.

1.3 Distributed Denial of Service attacks

A **Denial of Service (DoS)** attack is an attack trying to make a machine, service or network unavailable [44]. The attack works by flooding the target with requests in an attempt to overwhelm it, thus exhausting available resources and preventing legitimate requests from being fulfilled.

A **Distributed Denial of Service (DDoS)** attack is simply a DoS attack originating from multiple sources. The distributed nature of such attacks makes them all the more dangerous, as it is impossible to stop the attack by blocking a single source.

These attacks can be motivated by multiple reasons, including but not limited to revenge [62], activism [98], vandalism [99], financial reasons [72, 29] , political reasons [10], ...

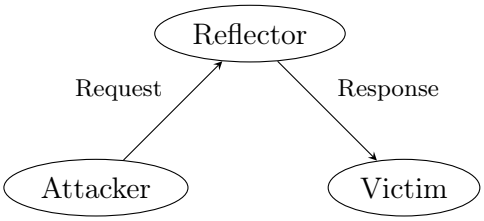


Figure 1.5: Reflection attack [108]

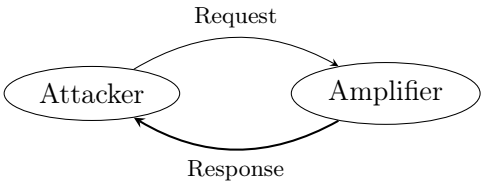


Figure 1.6: Amplification attack [108]

In this section, we will present an overview of DDoS attacks. DDoS attacks usually use two important mechanisms: reflection and amplification. Reflection is detailed in Subsection 1.3.1, while amplification is described in Subsection 1.3.2.

1.3.1 Reflection

In a reflection attack, the attacker spoofs the IP address of the victim in order to redirect the traffic they send to third parties. This serves many goals:

- Attackers effectively hide their identity from the victim, as all the attack traffic comes from benign third parties.
- Attackers can trigger attacks coming from different geographic or topological regions.
- Attackers do not receive responses, saving their available bandwidth.

Figure 1.5 portraits a reflection attack, where an attacker sends a spoofed request to a reflector, which directs the response to the victim.

Reflection attacks are often used in conjunction with amplification attacks.

1.3.2 Amplification

In an amplification attack, the attacker tries to exploit vulnerabilities in Internet protocols to amplify the amount of traffic they can use for an attack. Indeed, some protocols have a higher response payload size than the requests and thus can be used to amplify the attack volume.

Conducting an amplification attack can have multiple benefits:

- Attackers save bandwidth on their network uplink.
- Attackers obfuscate their identity, as amplified attack traffic comes from third parties.
- Attackers can trigger attacks coming from different geographic or topological regions.

Figure 1.6 depicts an amplification attack, where an attacker sends a request to an amplifier, and gets an amplified response. In this scenario, the victim of the attack is either the network between the attacker and the amplifier, or the amplifier itself. It is also important to note that in this scenario, the attacker must be able to withstand potential response traffic, or to make sure he does not receive it. The latter can be accomplished by address renewals, or simply as a result of a successful attack.

The amplification factor of an attack $X(P)$ for a protocol P is the ratio of the total number of bytes in the amplified traffic[2] and the number of bytes sent by the attacker[2], as shown in Equation 1.1 [108].

$$X(P) = \frac{number\_of\_response\_bytes(P)}{number\_of\_request\_bytes(P)} \tag{1.1}$$

Many protocols have been used in amplification attacks over the years. As of now, **CLDAP** is the most used protocol for this type of attack (29.4% of UDP reflection and amplification attacks in the last three months as of 2018-08-13 [91]).

While the surge in CLDAP attacks is relatively recent, **DNS** was the more popular choice for a long time. This is not surprising considering that at the time of writing, more than 10 million open **DNS resolvers**[3] currently "pose a significant threat" to the Internet [79], although this number

[2]Explicitly including headers, padding and payloads of all protocols involved, such as Ethernet, IP, **UDP** or the application-level protocol.

[3]Public DNS servers configured to respond to hosts outside of their domain.

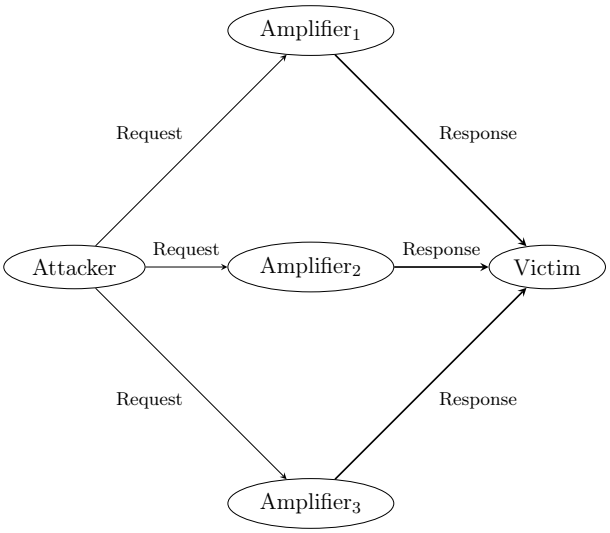


Figure 1.7: Combining reflection and amplification [108]

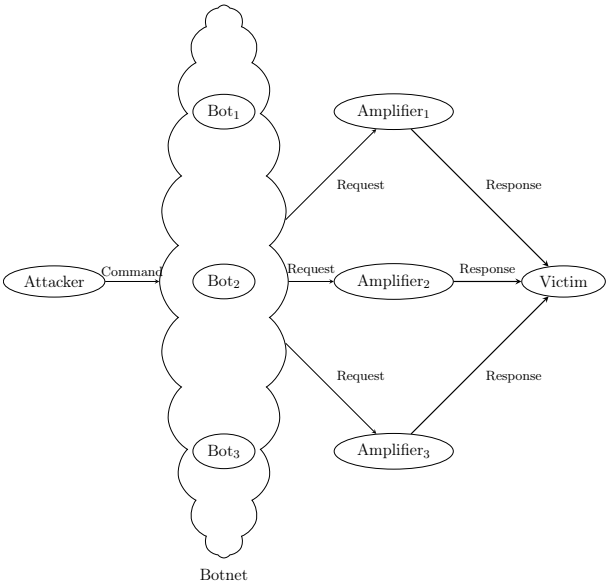


Figure 1.8: Botnet [108]

seems to be decreasing over time. Moreover, [106] finds that it only takes 92.5 seconds to get a list of 100,000 open DNS resolvers, making it quite easy to exploit them for DDoS attacks. [106] also finds that open DNS resolvers allow for an amplification factor of 64.1 (AF=64.1[4]).

UDP is often the preferred transport layer protocol in DDoS attacks (74.2% of DDoS attacks in the last three months as of 2018-08-13 [91]), as it offers the benefit of being stateless. It has no internal mechanism to verify the source of packets. Other UDP-based services have thus been used in amplification attacks, including but not limited to **SSDP** (AF=75.9[4]), **SNMP** (AF=11.3[4]) or even **NTP** (AF=4670[4]) [106].

There are other ways to amplify attack traffic volume. For example, an attacker may send requests (**ICMP** in the case of Smurf attacks; **CharGen** in the case of Fraggle attacks) to the broadcast address of networks whilst spoofing the IP address of victims. In this case, the routers receiving the requests act as amplifiers.

Amplification can also come from the amplified number of messages, rather than their amplified size. For example, networks running **NTP** can be subject to fork loops attacks, where requests are sent between SIP proxies indefinitely and at least one extra request is generated every iteration [108].

Amplification and reflection are very frequently used together in DDoS attacks (70% of all attacks as of 2018-08-13 [91]).

1.3.3 Combining Reflection and Amplification

Figure 1.7 shows a DDoS attack combining reflection and amplification. An attacker sends spoofed requests (reflection) to multiple amplifiers, which then direct their amplified responses to the victim. Such an attack is often called a **Distributed Reflective Denial of Service (DRDoS)**. This combination allows an attacker to benefit from the advantages of using both reflection and amplification.

Rather than sending himself all the requests to the amplifiers, an attacker can also make use of a **botnet**.

1.3.4 Botnets

When an attacker obtains access to a significant number of machines, by compromising them or by other methods, he can use them in coordination in what is called a botnet.

Figure 1.8 portraits such a botnet performing a DDoS attack. An attacker sends a command to all the bots under his control, making them send spoofed requests to multiple amplifiers, which

[4]Note that Rossow used a different equation for the amplification factor, only focusing on the number of UDP payload bytes.

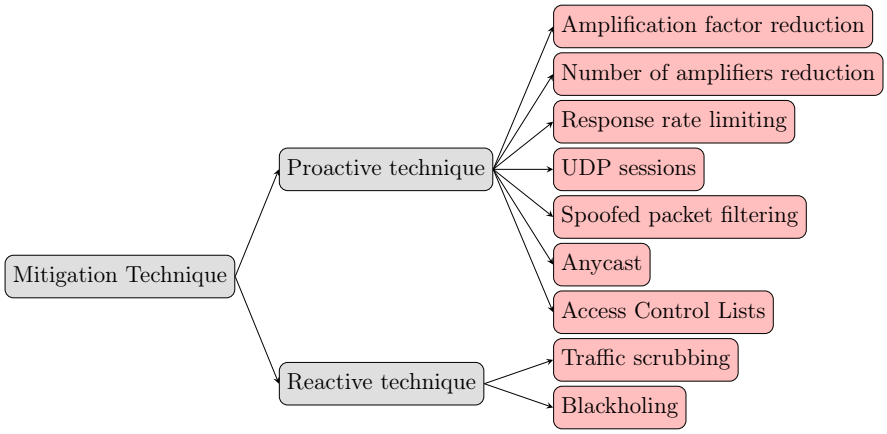


Figure 1.9: Mitigation taxonomy tree

direct their amplified responses to the victim.

This type of attack [92, 8, 126] is predicted to grow more lethal and prominent over time.

1.4 Blackholing: a DDoS mitigation technique

Fortunately, there are multiple ways to mitigate DDoS attacks [106, 108], some proactive and others reactive. This section lists some of the tools one can use to mitigate DDoS attacks, and details one in particular: **blackholing**. Figure 1.9 summarizes those tools.

Proactive techniques include:

- Designing protocols while taking into account its amplification factor, trying to reduce it.
- Reducing the number of amplifiers available to attackers.
- Implementing **Response Rate Limiting (RRL)**, reducing the rate at which servers respond to a high amount of forged queries.
- Using sessions for UDP, to be able to verify the source of packets.
- Filtering spoofed packets.
- Making use of anycast, spreading the load of the DDoS attack.
- Reduce the attack surface via **Access Control Lists (ACLs)**.

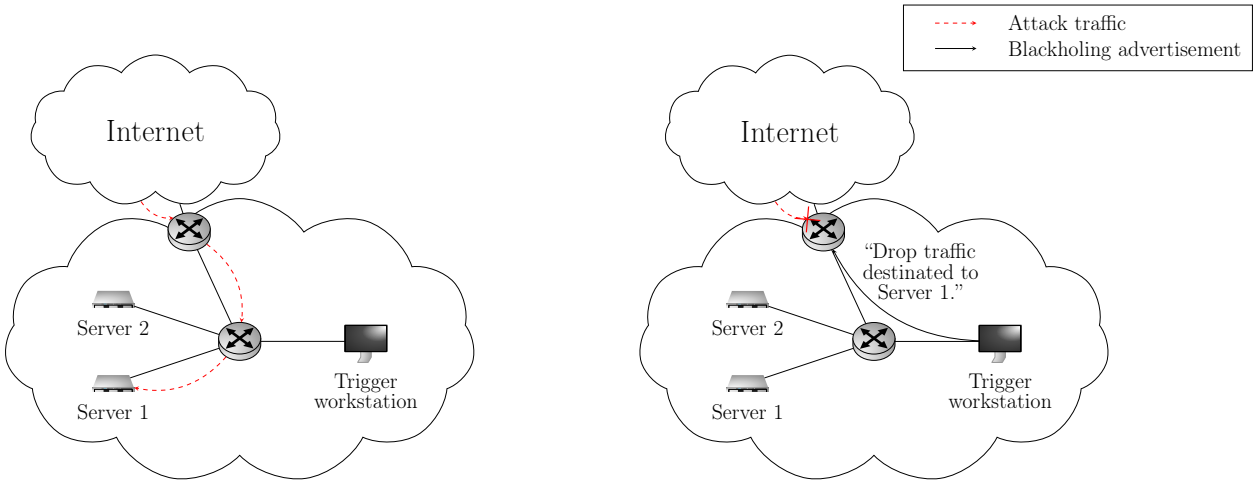
DDoS can also be dealt with in a reactive way. As soon as a DDoS attack is detected, a victim could activate its **traffic scrubbing**, or buy such a service from a third party. Traffic scrubbing is a (paying) service where a third party processes the victim's incoming traffic, detects and mitigates the attack, and then forwards the legitimate traffic to the victim. This is done via centralized data cleansing stations called scrubbing centers. Traffic scrubbing can be activated on demand, when an attack is detected. In this case, traffic is redirected to the scrubbing centers via DNS or BGP. Traffic scrubbing can also be activated by default. In this case, traffic always goes through the scrubbing centers before reaching the destination.

While filtering provides a great amount of flexibility, it runs into scalability issues in terms of the number of entries and the packet rate. Most routers are able to forward traffic at a much higher rate than they are able to filter, and are able to hold more forwarding table entries than filter entries [67]. A mitigation technique based on forwarding, such as blackholing, is thus much more scalable.

Blackholing [119, 67] allows an AS to specify that a set of routers or a network should discard any traffic destinated towards a specified IP prefix. This is also often referred to as **Remote Triggered Black Hole (RTBH)** filtering.

The primary use of blackholing is to discard all traffic destinated to a prefix under a DDoS attack. Dropping the traffic makes the specified prefix unreachable, thus fulfilling the goal of the attack, but the effects of the DDoS attack on the network infrastructure and other potential services are mitigated.

Figure 1.10 depicts a DDoS attack being stopped by blackholing. On the left (Subfigure 1.10a), Server 1 is receiving a DDoS attack. An advertisement is thus sent to the border router (Subfigure 1.10b), discarding all traffic destinated to Server 1, preventing collateral damages on the infrastructure and other potential services.



(a) Server 1 under a DDoS attack

(b) Blackholing in action

Figure 1.10: DDoS attack and mitigation by blackholing

Conceptually, the simplest way of requesting a blackhole is to call the network operator. After a phone call, the operator can manually activate blackholing on its routers. This method is highly unpractical and very slow, especially considering that the attack is ongoing even before phone calls are being made.

This is why automatic methods of blackholing are preferred. Blackholing is usually done through BGP, in two steps:

1. Send a trigger to the routers or networks that should blackhole the prefix.
2. Upon receiving a trigger, reroute the traffic destinated towards the specified prefix to a null interface.

1.4.1 Blackhole triggers

There are multiple ways to implement the trigger. [64] mentions the use of out-of-band BGP sessions with a special BGP speaker, but the two main methods are iBGP next hop triggers and BGP community triggers.

iBGP next hop trigger

Originally, a router of the AS wanting to blackhole a prefix would send a customized iBGP advertisement to other routers in the AS, modifying the next hop of the prefix to be blackholed.

The new next hop would often belong to private address space [101], as most routers in the Internet have static routes pointing those addresses to the null interface[5].

A router receiving this customized iBGP advertisement would update its forwarding table, resulting in the blackholed prefix being rerouted to the null interface of the router, thus being discarded. The scope of this trigger is local to the AS, as it propagates only through iBGP and does not cross AS borders.

Rerouting traffic to the null interface makes the blackholed prefix unreachable to the attacker and everyone else, but also has the undesirable side effect of making the blackholed prefix unreachable even to the local AS, as all the routers of the local AS will route traffic for this prefix to the null interface. To prevent this, [119] describes a new way to implement the trigger, to target only a set of routers for blackholing and maintain reachability of the blackholed prefix in the local AS, using BGP communities.

BGP community trigger

The operator of the local AS defines a unique community for each border router in its AS that could possibly receive attack traffic. The operator can also assign a community to a set of border routers, or all border routers.

[5] [67] also mentions the use of addresses in 192.0.2.0/24 [56], while [46] mentions use of address blocks reserved for documentation [4] and defines a specific discard prefix for IPv6, 0100::/64.

The community used for the trigger is defined by the operator of the network, but [64] defines a new well-known BGP community for operational ease, *BLACKHOLE*, which takes on the value ASN:666. [65] lists interesting information about the acceptance status of [64], namely supporting ISPs and IXPs, supporting software and supporting BGP speakers and manufacturers.

Then, border routers which will possibly blackhole traffic are configured with a static route pointing a private address (or equivalent) network to a null interface. Upon receiving an iBGP advertisement, a border router applies the following:

1. Match for a community value defined as a blackhole trigger.
2. Match the AS path to locally generated BGP advertisements.
3. Set the next hop to a private address (or equivalent) network.
4. Overwrite the trigger community with the community NO_ADVERTISE.

Blackholing is then triggered by sending a customized iBGP advertisement for the prefix to be blackholed and carrying the trigger community. The NO_EXPORT community is also added to this iBGP advertisement, to prevent it from crossing AS boundaries. The scope is thus local to the AS.

Using the match on blackhole communities, the next hop on non-triggering routers will be preserved, thus preventing traffic originating from the local AS and destinated to the affected prefix from being discarded. Matching locally generated advertisements prevents eBGP peers from misusing the community trigger to make the local AS blackhole any specified prefix. Overwriting the trigger community with the community NO_ADVERTISE prevents the router from propagating the advertisement to other routers.

If the operator of the local AS can determine from which routers the attack is coming from (Subsection 1.4.3), traffic destinated to the affected prefix and coming from routers not receiving attack traffic can also be routed appropriately by not triggering those routers.

The BGP community trigger is superior to the iBGP next hop trigger, but is harder to implement.

1.4.2 Source-based blackholing

Until now, traffic was always blackholed based on the destination of packets. A disadvantage of destination-based blackholing is that all traffic towards the blackholed prefix is discarded, including legitimate traffic.

Even though collateral damage to other systems and/or networks is reduced, one could argue that blackholing a prefix does more damage to this prefix than the DDoS attack itself, as it effectively takes the prefix offline, the goal of the DDoS attack in the first place. If a network operator could identify the sources of an attack, it would be much more convenient to discard traffic based on the source address of packets, as it would leave legitimate traffic unaffected.

[67] describes a way to blackhole traffic based on source address. This method makes use of **unicast Reverse Path Forwarding (uRPF)** [6] and will drop all traffic going to[6]/coming from the address. "uRPF performs a route lookup of the source address of the packet and checks to see if the ingress interface of the packet is a valid egress interface for the packet source address(strict mode) or if any route to the source address of the packet exists (loose mode)." [67]. If the check fails, the packet is dropped.

Strict mode will drop all ingress traffic if the best path back to the source is not the interface from which the traffic was received. Loose mode will drop all ingress traffic if no route back to the source exists. Some loose mode implementations also drop traffic if the route points to an invalid next hop (null interface), allowing source-based blackholing to work with the loose mode of uRPF.

The setup for source-based blackholing is the same as destination-based blackholing using communities, with the addition of enabling uRPF. Here as well, the blackholing is triggered by a BGP advertisement set with the appropriate community value and IP prefix, which will change the next hop of packets destinated to this prefix to the discard interface, thus causing the uRPF check to fail, and the traffic to/from this prefix to be dropped.

1.4.3 Blackholing analysis

A network operator may want to analyze the blackholed traffic, or have a way of knowing whether an attack is over or not. This can be achieved in multiple ways.

[6]Dropping traffic to the address is a side effect of uRPF, as there is no more valid route towards the source.

ICMP unreachable messages

As traffic is redirected to the null interface, routers should send ICMP unreachable messages to the source of the packets. To locate from which routers the attack traffic is coming from, the local AS can hijack one of the identified source addresses of the attack, announcing the address as its own into BGP. The device assigned with the address will receive the ICMP unreachable messages, which can reveal which routers are receiving attack traffic.

Note that this technique can only be used with destination-based blackholing and source-based blackholing with strict mode uRPF. Hijacking an address while using source-based blackholing with loose mode would cause the uRPF check to succeed, as a valid route towards the source exists, causing the attack traffic coming from this address to reach the victim.

Counts, sniffers and sinkholes

When a router blackholes traffic, it could count the number of drops, providing some information on the volume of the attack and reveal whether the attack is ongoing or over [67].

A way to capture the dropped traffic for further analysis is to install a sniffer on the spanned port of a switch [119].

Another way of capturing traffic is to redirect it to a sinkhole device, by reannouncing the attacked prefix, or a prefix that covers it into iBGP, with the sinkhole device as a next hop [119]. This has the same downside as the next hop blackhole trigger, as the traffic of the local AS will not be able to reach the legitimate destination and will be sucked into the sinkhole.

Sinkhole tunnels

A better alternative to sinkhole devices described in [119] is using sinkhole tunnels. A sinkhole tunnel is implemented on each border router that could possibly receive attack traffic. Using the same community trigger system, one could re-route traffic through the tunnel instead of to the null interface, where a sniffer could capture the traffic[7]. After exiting the tunnel, traffic can be routed to the legitimate destination, as the next hop won't be changed except for routers triggered by the blackhole community.

1.5 Problem statement

The objective of this internship is to review how blackholing can be misused as an attack vector, and what can be done to prevent or mitigate those attacks. Routing attacks in the Internet already exist, allowing attackers to block communications, redirect traffic, eavesdrop on communications or even send spam. The first goal of this internship is to determine in which ways blackholing can be abused, and thus construct an attack taxonomy of blackholing.

Fortunately, techniques to prevent or mitigate routing attacks already exist. Altough they are often lacking, they provide a basis on which we can improve to make blackholing and the Internet in general safer. The second goal of this internship is thus to determine good practices and security solutions to prevent the potential misuse of blackholing.

1.6 Outline

Considering routing attacks and defenses (Chapter 2), we construct an attack taxonomy using blackholing as an attack vector (Chapter 3). We detail good practices and implementations to protect against such attacks (Chapter 4), and present a tool to detect potential attackers (Chapter 5). Finally, Chapter 6 concludes this report by reviewing various contributions and describing the possible perspectives and areas of future work.

Appendix A gives more details on the inner workings of the Resource Public Key Infrastructure, while Appendix B gives more details on the behavior of the tool.

[7] As traffic passes through the tunnel, one could also implement rate-limiting policies, **Quality of Service (QoS)** policies or ACLs, to drop attack traffic (traffic scrubbing). This can be especially useful if those techniques cannot be implemented at border routers due to hardware or software limitations.

Chapter 2

Scientific context

In order to see how blackholing can be misused in routing attacks, we first need to look at already existing routing attacks, namely BGP **hijacking**.

2.1 BGP routing attacks

BGP was designed when security was of little concern and attacks were nowhere near as dangerous as today. As BGP is inherently based on trust, routing anomalies can occur [78], caused by misconfiguration or malicious intent.

In this section, we will talk about some of these routing anomalies, namely BGP **hijack**s and BGP **leak**s.

2.1.1 BGP hijacking

BGP is a distributed protocol, lacking authentication of routes. This allows ASes to advertise illegitimate routes for prefixes they do not own, attracting some or all of the traffic to these prefixes. Those advertisements propagate and pollute the Internet, affecting service availability, integrity, and confidentiality of communications [111]. This phenomenon is called BGP hijacking (also referred to as prefix hijacking or IP hijacking).

Hijacks can be caused by misconfiguration [104, 116, 24], or with malicious intent, possibly motivated by retaliation [114], information gathering [75], economical reasons [43, 42, 77] or political reasons [76]. When a hijack is caused by misconfiguration, the event is referred to as a route leak/BGP leak, but they are similar in form and effect to hijacks. One could argue that BGP hijacks and BGP leaks usually differ in size, as a leak usually involves the hijack of a significant amount of prefixes, whereas a hijack done with malicious intent tends to be smaller in size.

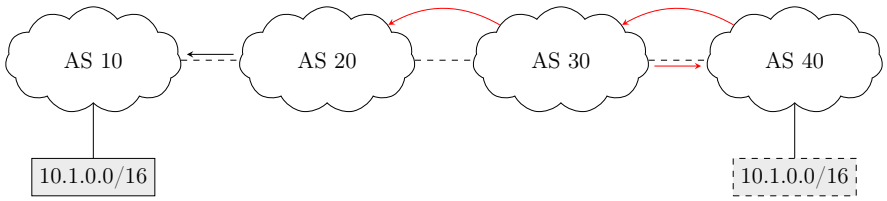
A hijacking example

On Figure 2.1, AS 10 (the victim) advertises a route for the prefix 10.1.0.0/16. The hijacker (here AS 40) can fake a direct connection to this network and advertise 10.1.0.0/16 to AS 30. Preferring the shorter AS path, AS 30 will choose a new best route going through AS 40, and forward the hijack to AS 20. AS 20's original route is already the best one, so it does not accept the hijack and does not forward the advertisement to AS 10.

2.1.2 Hijacking taxonomy

Hijacking can take many forms. In [111], the researchers develop an attack taxonomy classifying those hijacks. This taxonomy is based on a common and general hijacking threat model [110]. An attacker controls a single AS and its border routers. He also has full control of the control plane and the data plane within its own AS. The attacker can arbitrarily manipulate the advertisements that it sends to its neighboring ASes and the traffic that crosses its network. He has no control over advertisements and traffic exchanged between two other ASes. Their taxonomy is based on three dimensions:

- The manipulation of the AS path.
- The affected prefix.



(a) Prefix hijack with exact match

AS Number	AS path	
	Before Hijack	After Hijack
AS 10	-	-
AS 20	10	10
AS 30	20 10	40
AS 40	30 20 10	-

(b) Evolution of the AS path towards 10.1.0.0/16

Figure 2.1: Hijack message propagation and respective AS paths

- The way (hijacked) data traffic is treated.

To illustrate those hijack types, let us reconsider Figure 2.1, where AS 10 (the victim AS) owns and legitimately announces 10.1.0.0/16, and AS 40 is the hijacking AS. For the sake of simplicity, a BGP advertisement is noted as an announced prefix tagged with an AS path. For example, {AS20,AS10 - 10.1.0.0/16} is a BGP advertisement for prefix 10.1.0.0/16 with AS path {AS20,AS10}, originated by the legitimate AS (AS 10). In their paper, they first classify by AS path manipulation, creating three categories of hijacks:

- Origin AS (or Type-0) hijacking: The hijacker announces as its own a prefix that it is not authorized to originate (e.g. {AS40 - 10.1.0.0/16}). This type of hijack is sometimes called prefix re-origination, and is the most commonly observed type of hijack [111].
- Type-N hijacking (N $\geq$ 1): The hijacker announces an illegitimate path for a prefix it does not own, creating fake adjacencies between ASes. The path contains the ASN of the hijacker as the last hop (e.g. {AS40,AS20,AS10 - 10.1.0.0/16}). Here, AS 40 creates a fake adjacency between itself and AS 20. The position of the rightmost fake link in the forged advertisement determines the type. For example, {AS40,AS10 - 10.1.0.0/16} is a Type-1 hijacking, {AS40,AS20,AS10 - 10.1.0.0/16} is a Type-2 hijacking, etc.
- Type-U hijacking: The hijacker leaves the legitimate AS path unaltered.

The second discriminant is the affected prefix:

- Exact prefix hijacking: The hijacker announces a path for the exact same prefix that is announced by the legitimate AS. Since shortest AS paths are typically preferred, only part of the Internet that is close to the hijacker (in terms of AS hops) switches to route towards the hijacker.
- **Sub-prefix** hijacking: The hijacker announces a more specific prefix, i.e., a sub-prefix of the prefix of the legitimate AS. For example, AS 40 can announce {AS40 - 10.1.0.0/17} or {AS40,AS10 - 10.1.0.0/17}. Since routes to more specific prefixes are preferred, the entire Internet traffic is sent towards the hijacker to reach the announced sub-prefix. Note that since most routers do not accept BGP advertisements containing a prefix past a certain length (usually /24) to reduce routing table size, a sub-prefix hijack advertising a /25 or more may not be very effective, as the advertisements will be dropped.
- Squatting: The hijacker AS announces a prefix owned but not (currently) announced by the owner AS.

The last discriminant is the way the data-plane traffic is handled. Once the hijack is accomplished, the attacker attracts some or all of the traffic originally destinated to the hijacked prefix to his own AS. The attacker can then [128, 111]:

- Drop the packets (Blackholing).

- Impersonate the AS and its services by responding to the victims (Imposture).
- Eavesdrop on the traffic and forward it back to the victim (Interception/Man-in-the-Middle) [25].
- Send spam [121], and/or carry out other activities.

According to this taxonomy, the hijack depicted in Figure 2.1 is a Type-0 exact prefix hijack, as AS 40 re-originates 10.1.0.0/16.

Note that this taxonomy can be extended, as it does not cover cases where, for example, the hijacker possesses two or more ASes.

Given the size of the Internet, it is impossible to replace BGP with a more secure protocol. This has to be done incrementally by evolving BGP [83]. Fortunately, techniques exist to protect oneself against hijacks, and make sure received advertisements are legitimate.

2.2 AS path validation

When receiving an advertisement, a router might want to verify that the included AS path is legitimate. This process can be broken down in two validation steps:

- **Origin validation**: Does the Origin AS have a right to announce this prefix?
- **Path validation**: Does the sequence of ASes in the AS path reflect the sequence of ASes crossed by this advertisement?

2.2.1 Origin validation

Origin validation can be achieved through the **Internet Routing Registry (IRR)**[1] or the **Resource Public Key Infrastructure (RPKI)**[2].

The Internet Routing Registry

The IRR is a distributed routing database, which provides among other things a way to verify which prefix belongs to which AS. Other public registries of network routing information, like the **Routing Assets Database (RADb)** [81], provide this feature. Use of the IRR/RADb as a means of verifying the origin of an advertisement might be compromised by the fact that IRR/RADb records are not fully secure/reliable, contain numerous errors and are incomplete [40, 115, 83].

The Resource Public Key Infrastructure

The RPKI [69] is a distributed, hierarchic public key infrastructure. It allows **prefix holder**s (legitimate holders of IP address space) to emit digitally signed routing objects attesting that a given AS is authorized to originate routes to (a subset of) those prefixes. This way, a given AS can verify that the Origin AS present in a given advertisement is authorized to originate the prefix.

RPKI is independent of BGP: BGP does not need to be modified to allow RPKI to function. An advantage of RPKI over IRR/RADb is that the mapping of prefixes to origin ASes is formally verifiable [87].

RPKI is actually composed of three elements:

- The RPKI.
- Digitally signed routing objects.
- A distributed repository system to hold the objects.

Resource certificates

Certificates in the RPKI are called **resource certificate**s, and attest to IP address or ASN holdings. As such, when prefix holders in the RPKI want to delegate an address block to another entity, they can sign a resource certificate attesting of this delegation. This means every resource holder in the

[1] http://www.irr.net/

[2] Defined by the Internet Engineering Task Force (https://www.ietf.org/), and more specifically the Secure Inter-Domain Routing working group (https://datatracker.ietf.org/wg/sidr/charter/).

RPKI has a resource certificate attesting of such ownership, and assumes the role of **Certification Authority (CA)**.

It is worth noting that certificates do not attest to the identity of the prefix holder, RPKI is intended to provide authorization, not authentication.

CAs can then use their resource certificate (**CA certificate**) to sign routing objects. More specifically, when a CA wants to sign a routing object, it generates a new certificate, called an **End-Entity certificate (EE certificate)**. The private key corresponding to the public key in the EE certificate will be used to sign the routing object, and the EE certificate itself will be signed with the private key corresponding to the public key in the CA certificate. As EE certificates are only used to sign one routing object, and every routing object is signed by only one key, there is a one-to-one correspondence between EE certificates and routing objects. This means that to revoke a routing object, a CA must only revoke the associated EE certificate. Convenience aside, this one-to-one correspondence also means that the private key of an EE certificate is only used once, and can be destroyed after it has been used to sign the object, improving security and simplifying key management. For more details on certificates used in RPKI, see [52].

CA certificates are based on the X.509 certificate profile defined in [22] with the extensions for IP addresses and ASNs [74]. **Cryptographic Message Syntax (CMS)** [47] is used as the syntax for routing objects (See [68] for templates).

In a **Public Key Infrastructure (PKI)**, **Relying Parties (RPs)**, entities participating in the PKI, must each choose a set of **Trust Anchors (TAs)**. Those TAs are trusted by definition, trust in other entities is derived from those TAs via chains of trust. As the structure of the RPKI corresponds to the existing resource allocation structure (IP adresses and ASNs), with **Regional Internet Registries (RIRs)** assuming the role of roots, it is convenient that RIRs also assume the role of default TAs in the RPKI.

Route Origin Authorizations

Route Origin Authorizations (ROAs) are routing objects providing an authorization by a prefix holder that a given AS is permitted to originate routes to a set of prefixes. A ROA essentially contains an ASN, a set of prefixes and optionally, for each prefix, the maximum length of more specific prefixes that the AS is also authorized to advertise. For more details about the contents of ROAs, see [70].

For example, a ROA can have the meaning: "AS3356 is authorized to announce 104.37.56.0/22". By default, this authorization is strict: only advertisements with the same prefix will be viewed as valid. The advertisement of a more specific prefix, 104.37.56.0/24 for example, will be viewed as invalid. To have more flexibility, the AS can set a maximum length of 24, making the advertisement of 104.37.56.0/24 valid.

To forbid the origination of a prefix, a prefix holder can create a ROA associating this prefix to the AS 0 [51, 66]. Since no valid route can have an Origin AS of zero , no route can be matched by a ROA whose ASN is zero. This is useful either for reserved prefixes, or allocated prefixes which are not meant to be announced.

The example in Figure 2.2 depicts such a system. RIPE, the European RIR, uses its self-signed certificate to attest that ISP 1 is the legitimate holder of 10.0.0.0/8. ISP 1 then delegates portions of its address space to AS 2 (10.16.0.0/24) and AS 3 (10.0.0.0/12) using its CA certificate to sign CA certificates for AS 2 and AS 3. Those two entities then use their CA certificates to sign EE certificates which are used to sign ROAs authorizing a given AS to originate a given set of addresses.

Other routing objects exist, but are not needed to understand the concept of origin validation. Details concerning those objects can be found in Appendix A.

The distributed repository system

To make use of ROAs, an RP must acquire and validate all of them. This is what the distributed repository system is meant for. Every CA certificate is associated to a repository containing all the routing objects associated with this certificate (and not yet expired). This repository is called a **publication point**. One of the databases composing the distributed repository system can contain one or more publication points. This means a given database will most likely not have all of the RPKI objects, which is why publication points are organized in a hierarchy. Although roots of this hierarchy are arbitrarily chosen by each RP, today, the default five roots of the hierarchy are the five RIRs. To recuperate all of the RPKI objects, each CA certificate contains two fields:

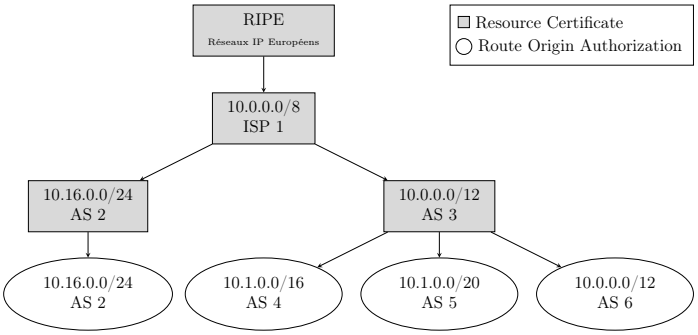


Figure 2.2: RPKI allocation example

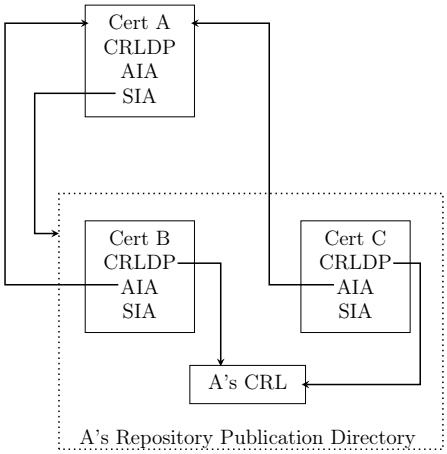


Figure 2.3: RPKI repository example (taken from Section 4.2 of [69])

- **Subject Information Access (SIA)**: **Uniform Resource Identifier (URI)** that points to the repository associated with this CA certificate.
- **Authority Information Access (AIA)**: URI referencing the location of the CA certificate used to issue this one.

Thus, to recuperate all the routing objects, an RP starts with the five root publication points and follows SIAs recursively, pulling all objects from the directories.

Figure 2.3 depicts the usage of those fields. the CA A issued two CA certificates. Those certificates are in A's publication point. The SIA of A's certificate points to this publication point. The AIAs of CA certificates B and C point to A's CA certificate. Since B and C have certificates, they can either have their publication point (or even their **Certificate Revocation List (CRL)**) hosted in the same place as A, or they can host their publication point somewhere else. In any case, the SIA in their CA certificates will point to those directories.

CRLs and **CRL Distribution Points (CRLDPs)** are part of the certificate revocation mechanism. More details on revocation can be found in Appendix A.

Publications points also need to provide several other functionalities, namely downloading, uploading , modifying and deleting. More details on those functionalities can be found in Appendix A.

Local caches

As said earlier, routers do not interact with the repositories directly, but via a local cache. Section 6 of [69] details how RPs populate and validate their caches. This can be done via rsync [107] or any other download protocol, like the **RPKI Repository Delta Protocol (RRDP)** [11].

The RPKI to Router Protocol

Once a cache is populated and objects within it are validated, how does an RP make use of this information? [14] (updated by [15]) introduces the RPKI to Router Protocol to allow routers to receive validated prefix origin data from one or multiple caches.

The objects loaded have the content (IP address, prefix length, maximum length, **Origin ASN**): it is a **Validated ROA Payload (VRP)** [87].

Route Origin Validation

Once the router has loaded all prefix origin data from the cache, how does it use that information to validate (or invalidate) BGP advertisements? [51] describes this process. When the router receives an advertisement, it needs to match the route with one or more (syntaxically valid and correctly signed) candidate ROAs. Depending on the matched ROAs, the route is classified as either "valid", "invalid", or "unknown"[3]. This validity state then determines the actions to perform on the route.

[3]The "unknown" validity state is needed to account for partial RPKI deployment scenarios, where only a subset of ASes have deployed RPKI.

Table 2.1 outlines the route's validity state when comparing the prefix and the Origin AS of the route with a **single** ROA.

Prefix / AS	matching AS	non-matching AS
Non-Intersecting	unknown	unknown
Covering Aggregate	unknown	unknown
match ROA prefix	valid	invalid
More Specific than ROA	invalid	invalid

Table 2.1: Route's Validity State (taken from Section 2 of [51])

If the route's Origin AS matches the one in the ROA (VRP ASN), and the route's prefix matches the one in the ROA[4] (VRP prefix), then the route is marked as "valid".

If the route's Origin AS does not match the VRP ASN, and the route's prefix matches the VRP prefix, then the route is marked as "invalid". If the route's prefix is more specific than the VRP prefix, the route is marked as "invalid", as the ROA encompasses all address prefixes that are more specific than the one in the ROA.

If the ROA is a covering aggregate of the route's prefix, or if the ROA does not intersect the route's prefix, the route has an "unknown" state as it cannot be reliably classified as "invalid" in a partial deployment scenario.

In a more realistic context where multiple (valid) ROAs exist:

- If any ROA provides a "valid" outcome, the route is considered to be "valid".
- If no ROA provides a "valid" outcome, and at least one ROA provides an "invalid" outcome, the route is considered to be "invalid".
- If no ROA provides either a "valid" or "invalid" outcome, the route validity state is considered to be "unknown".

[87] has another way to make this distinction, but it equates to the same result:

- NotFound: No VRP Covers[5] the route prefix.
- Valid: At least one VRP Matches[6] the route prefix.
- Invalid: At least one VRP Covers[5] the route prefix, but no VRP Matches[6] it.

The validity state of a route is determined by first selecting all ROAs which have a prefix that matches or covers the route's prefix: those are the "candidate ROAs". If this set is empty, the route's validity state is set to "unknown". If any of the ROAs in this set matches[4] the route's Origin AS and the route's prefix, the route's validity state is set to "valid". Otherwise, the route's validity state is set to "invalid". Figure 2.4 depicts this procedure.

Even though we use RPKI to determine a validity state, it is possible to use any other database mapping prefixes to their Origin ASes.

The outcome of this procedure can be used in BGP's decision process (see Subsection 1.2.2). This means the validity state of a route can affect local preference, in which case "valid" is to be preferred over "unknown" which is to preferred over "invalid". The actions to apply in each case are a matter of local routing policy. Considering partial deployment scenarios, RPs might not want to reject "unknown" routes, as it would seriously impact connectivity in the Internet. Even "invalid" routes might not be worth rejecting as long as the adoption rate of RPKI is low.

Entities participating in the RPKI must be careful when issuing ROAs. Issuing a ROAs implicitly invalidates all routes that have more specific prefixes (with a prefix length greater than maxLength), and all Origin ASes other than the AS in the ROA. Thus, [51] suggests a conservative operational practice: ensure the issuing of ROAs for all more specfic prefixes with distinct Origin ASes prior to issuing ROAs for the covering aggregate.

RPs should also take into consideration that "like the DNS, the global RPKI presents only a loosely consistent view, depending on timing, updating, fetching, etc. Thus, one cache or router

[4]Here, "match" is defined as either the route's prefix is matching the ROA's prefix exacly, or the route's prefix is more specific than the ROA's prefix and its length is no longer than maxLength.

[5]The route prefix is either identical to the VRP prefix or more specific than the VRP prefix.

[6]The route prefix is Covered by the VRP, the route prefix is no longer than the VRP maximum length, and the route Origin ASN is equal to the VRP ASN.

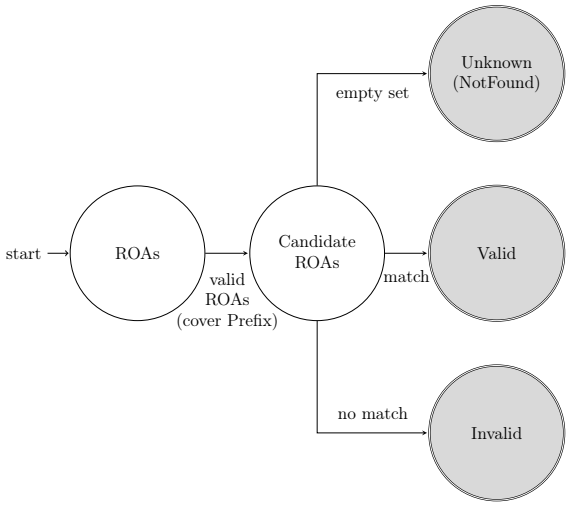


Figure 2.4: Procedure to determine a route's validity state

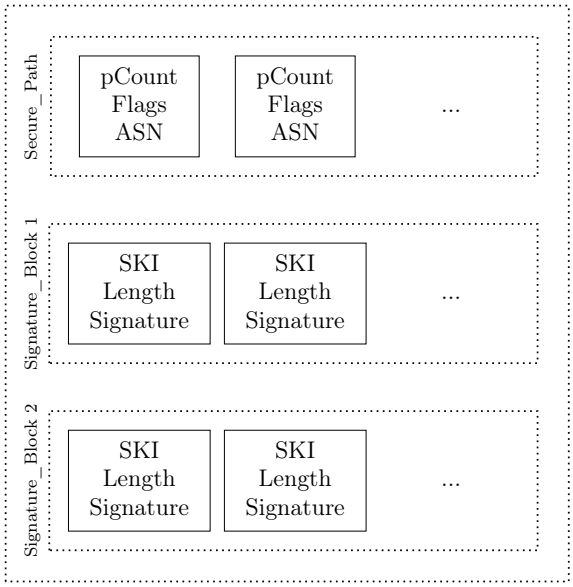


Figure 2.5: BGPsec_PATH attribute

may have different data about a particular prefix than another cache or router. There is no 'fix' for this; it is the nature of distributed data with distributed caches." [87].

RPs might need to propagate the validity state of a route into iBGP for routing correctness. This can be accomplished by using communities (or extended communities [86]).

"If "invalid" routes are blocked as this specification suggests, the overall system provides a possible denial-of-service vector; it could lead an otherwise "valid" route to be marked "invalid"." [87].

When fully deployed, RPKI only protects against Type-0 hijacks, but all type of sub-prefix hijacks.

2.2.2 Path validation

Path validation can be achieved through **BGPsec**[7] [71]. BGPsec relies on RPKI (Subsection 2.2.1) as it makes use of certificates.

To secure the path attribute, BGPsec relies on an new optional non-transitive BGP path attribute which replaces the AS_PATH attribute: BGPsec_PATH. The attribute carries digital signatures providing cryptographic assurance that every AS on the path of ASes listed in the advertisement has explicitly authorized the advertisement of the route. BGPsec-compliant BGP speakers (**BGPsec speakers**) wishing to send BGPsec advertisements to eBGP peers need to possess a private key associated with an RPKI router certificate [103] that corresponds to the BGPsec speaker's ASNs. Traditional BGP advertisements may still be sent between BGPsec speakers.

The BGPsec_PATH attribute is made of a Secure_Path and one or two Signature_Blocks (see Figure 2.5).

The Secure_Path attribute

The Secure_Path contains AS path information: for each AS in the path, the Secure_Path will contain a Secure_Path Segment. Each Secure_Path Segment is itself composed of a pCount field, flags, and an ASN. The pCount fields indicates the number of times the ASN is present in the path, allowing ASes to prepend their ASN multiple times to the path if they so desire. The pCount field also comes into use for AS confederations [118], managing route servers and ASN migrations [39]. In particular, route servers, which are usually "invisible" (they do not add their ASN to the path attribute of an advertisement) can add their ASN with the pCount field set to zero, to still benefit from BGPsec security. The Confed_Segment flag is the only flag defined as of today. It indicates that the BGPsec speaker that constructed this Segment is sending the advertisement to an AS within the same confederation.

[7] Defined by the Internet Engineering Task Force (https://www.ietf.org/), and more specifically the Secure Inter-Domain Routing working group (https://datatracker.ietf.org/wg/sidr/charter/).

The Signature_Block attribute

The Signature_Blocks contain a Signature Segment for each AS in the Secure_Path. Each Signature Segment contains a **Subject Key Identifier (SKI)**, a Signature Length and a Signature. The SKI is used to identify the router certificate which generated the Signature. The Signature protects among other things the **Network Layer Reachability Information (NLRI)**, the BGPsec_PATH attribute and the ASN of the peer to whom the advertisement is being sent. The Signature_Blocks also identify the used digest algorithm and digital signature algorithm via an Algorithm Suite Identifier field (see [120] for more details). The possible presence of another Signature_Block is by design, to be able to transition algorithm suites while retaining backwards compatibility (see [71] for more details).

BGPsec advertisement generation

To originate a prefix using BGPsec, an AS must first generate the BGPsec_PATH attribute. First, the AS generates a Secure_Path Segment:

- Set the pCount value to the appropriate value (usually to one).
- Set the Confed_Segment flag if appropriate.
- Set the ASN to the number of the AS generating the Segment.

Then, the AS generates the corresponding Signature_Block Segment. The SKI field is populated by the SKI of the router certificate used to verify the signature. The Signature field is populated by applying the digest algorithm to a specific sequence of elements (Table 2.2), and then applying the signature algorithm to the obtained digest value.

Target AS Number	
Signature Segment	: N-1
Secure_Path Segment	: N
...	
Signature Segment	: 1
Secure_Path Segment	: 2
Secure_Path Segment	: 1
Algorithm Suite Identifier	
AFI	
SAFI	
NLRI	

Table 2.2: Sequence of Octets to Be Hashed (Figure 8 of [71])

As mentioned earlier, the element sequence to be signed contains among other things the NLRI, the Secure_Path Segment and the ASN of the peer to whom the advertisement is being sent. The AS is then ready to send the BGPsec advertisement.

An AS receiving a BGPsec advertisement and wanting to forward it goes through the same process, prepending blocks to the ones already present in the advertisement.

As it is required to specify the ASN of the destination of the advertisement, BGPsec speakers wishing to send a BGPsec advertisement to multiple peers must generate a separate advertisement for every one of them. BGPsec speakers must also advertise a route to only a single prefix to be able to construct a valid BGPsec advertisement. If a BGPsec speaker wishes to advertise routes for multiple prefixes, it must generate a separate advertisement for each prefix it wants to advertise. Note that the BGPsec_PATH and AS_PATH attributes are mutually exclusive, there can be only one in an advertisement.

Of course, a BGPsec speaker cannot send a BGPsec advertisement to a non-BGPsec speaker. In case a BGPsec speaker wants to propagate an advertisement to a non-BGPsec speaker, the BGPsec

speaker needs to "downgrade" the advertisement to a normal BGP advertisement. To this end, Section 4.4 of [71] specifies a way to reconstruct the AS_PATH attribute from the BGPsec_PATH attribute. Doing this has “significant security ramifications”, as an attacker could for example try to downgrade an advertisement to make a route insecure [73].

An AS receiving a BGPsec advertisement performs basic integrity and syntax checks. It notably checks (like the unsecured version of BGP) if the leftmost AS in the path attribute matches the ASN of the peer that sent the advertisement. Any errors in those checks is handled using the "treat-as-withdraw" rule [18]. The AS can then choose to validate the advertisement to determine the authenticity of the path.

Path validation

[71] expects most RPs using BGPsec to use the RPKI for origin validation, and recommends that a BGPsec speaker should only send a BGPsec advertisement if there exists a valid ROA for the prefix it wants to originate.

The path validation procedure either ends in a "Valid" or "Not Valid" state. Like with origin validation, the outcome of this procedure can be used in BGP’s decision process (see Subsection 1.2.2) to lower route preference or reject routes, in which case the "Valid" state is obviously preferred over the "Not Valid" state. This is once again a matter of local policy. To validate the path, the BGPsec speaker iterates through the Signature_Block Segments, starting with the most recently added one. For each Segment, the BGPsec speaker finds the certificate used for the Signature field, runs the digest algorithm on the same element sequence the Signature was generated from, and uses the signature validation algorithm. If one of the Signature_Block Segments is marked as "Not Valid", the advertisement is "Not Valid". Otherwise, the advertisement is considered "Valid".

Like with RPKI adoption, the transition period from BGP to BGPsec will be long. It is assumed BGPsec speakers will form small contiguous groups, that will grow and merge over time. Only routes originated and propagated within those groups will get the security benefits of BGPsec.

Operational considerations for BGPsec are found in [13]. A threat model and security consideration are found in [63].

Security concerns

As the path validation procedure performs expensive checks, it is a potential target for DoS attacks (Sending many advertisements, or advertisements with long path attributes). It is therefore advised to perform expensive checks after less expensive checks, and to stop the validation procedure as soon as a "Not Valid" Signature_Block Segment is found. Another approach is to only try to validate advertisements that would be chosen as best path if they were found "Valid".

Another security concern is that ASes (besides route servers, confederations or migrating ASes) might try to use the pCount field to attract more traffic (by setting pCount to zero, shortening the path). Section 7.2 of [71] advises ASes to perform a check on the pCount field of the leftmost Secure_Path Segment of a received advertisement. If the field is set to zero when it should not, the AS should treat the advertisement as an error (using the "treat-as-withdraw" rule). Note that an AS can only verify the pCount of the last hop, and cannot verify if other pCount fields in the advertisement were set to their respective values legitimately.

There is also the possibility to pass a BGPsec advertisement through a tunnel between colluding ASes, faking a link between them. Detecting such behaviour is beyond the scope of BGPsec, as it is only meant to secure the path of BGP advertisements, not the bypass of the control plane.

Furthermore, BGPsec does not provide transport layer protection. BGPsec sessions should thus be secured using appropriate mechanisms such as the **TCP Authentication Option (TCP AO)** [117] or **IP Security (IPsec)** [35].

Finally, an adversary on the path between a BGPsec speaker and its peer is able to modify valid BGPsec advertisements to cause them to fail validation, inject BGP advertisements without BGPsec_PATH, inject BGP advertisements with BGPsec_PATH failing validation, or causing the peer to tear down the BGP session, or tamper with other BGP attributes. An on path adversary cannot make a BGPsec speaker believe a "Not Valid" route is "Valid".

A malicious BGPsec speaker can also make use of replay attacks, by suppressing a prefix withdrawal (implicit or explicit), or replaying a previously received advertisement which has since been withdrawn. This attack vector can be mitigated by doing a periodic rollover of router certificates

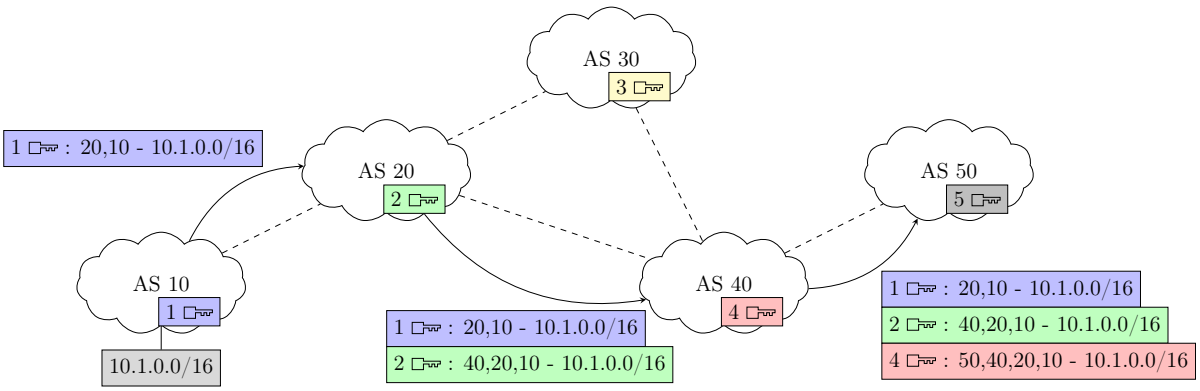


Figure 2.6: BGPsec advertisement propagation

Security guarantees when using BGPsec along origin validation are as follows:

- The Origin AS of the advertisement is authorized to announce the prefix.
- The advertisement was propagated along the path in the path attribute of the advertisement.

BGPsec does not guarantee that the data plane will follow the control plane.

To validate the path, each AS on the AS path, starting with the Origin AS, writes in the advertisement which AS it is sending it to. The AS then signs this field along with the AS path. This way, every AS receiving the advertisement can verify that the path is legitimate by verifying the signatures present in the advertisement.

This process is illustrated in Figure 2.6. AS 10 originates 10.1.0.0/16, and as such signs a BGP advertisement containing this prefix, the AS path (10) and the ASN of the recipient of the advertisement (20) with its private key. When receiving the advertisement, AS 20 can then verify the information present in this advertisement by using AS 10's public key. To forward the advertisement to other ASes, AS 20 adds itself to the AS path and specifies it wants to send the advertisement to AS 40. After signing the appropriate sequence of elements, AS 20 can send the advertisement to AS 40, which will go through the same steps to forward it to AS 50.

If AS 20 wanted to forward the advertisement to AS 30, it would have needed to generate another advertisement specific to AS 30.

Unfortunately, BGPsec is of no use against attacks using communities: "attacks that modify (or strip) these other attributes are not prevented/detected by PATHSEC" [63] (PATHSEC referring to "any BGP path security technology that makes use of the RPKI").

Note that RPKI provides origin authentication, while BGPsec provides authentication and integrity of the path attribute.

2.3 Good practices

Blackholing is a very effective tool [28], but it can be quite dangerous if not used appropriately. This section describes current general and specific good practices for blackholing. Subsection 2.3.1 describes general operational guidelines while Subsection 2.3.2 describes general blackholing good practices. Subsection 2.3.3 describes blackholing good practices specific to whether blackholing is used locally or between ASes.

2.3.1 General operational guidelines

First of all, it is recommended to follow the Gao-Rexford model. I emphasize once again that ASes are not obliged to follow those rules, but doing so makes economical sense, and helps BGP converge to a stable state.

Second of all, it is important to follow the guidelines of [100] and [30]. [30] describes several protection techniques and filters, that an operator should follow in most cases. This includes securing BGP speakers, securing BGP sessions between them and regulating which routing information is exchanged. The latter is comprised of:

- Filtering special-purpose prefixes.
- Filtering unallocated prefixes.

- Filtering prefixes that are too specific.
- Filtering prefixes belonging to the local AS and downstreams.
- Filtering prefixes based on IRRs or the RPKI/BGPsec.
- Setting a limit on the number of routes accepted from a peer.
- AS path and next-hop filtering.
- Community scrubbing.

[30] also recommends which inbound and outbound filters should be used, depending on the type of the peering session.

ASes can of course agree on exceptions to generic guidelines, but those can impact the entire Internet, so they must be made with caution.

2.3.2 Blackholing operational guidelines

Blackholing should be implemented by following the planning and guidelines of [67]. Notably, [67] introduces several interesting ideas:

- Using multiple discard addresses, allowing an operator to configure one address per attack, which makes the analysis of each separate attack easier.
- Using multiple trigger communities to control the scope of blackholing (filtering on all routers, on data centers, on peering routers, ...).

Of course, a network operator should always log BGP advertisements carrying communities used for blackholing for long-term analysis [64], via one of the methods described in Subsection 1.4.3.

As said earlier, network operators generally do not accept BGP advertisements longer than /24. However, in the case of blackholing, an operator wants to blackhole the longest prefix possible, to limit the impact of discarding traffic for adjacent IP space that is not attacked. Thus, an operator should accept blackholing advertisements up to /32 for IPv4, and /128 for IPv6 [64].

To the best of our knowledge, there are no considerations for blackholing in the literature concerning BGPsec.

2.3.3 The scope of blackholing

In Section 1.4, I explained how blackholing was used within an AS, but blackholing can also be used between BGP peers.

An AS might want another AS to drop traffic for multiple reasons. For example, a stub AS being the victim of a DDoS attack might want its provider to blackhole the traffic, because it is receiving a 100 Gb/s DDoS attack and its link to the provider is only 10 Gb/s. In this case, the stub AS can mitigate the attack by blackholing at its edge of the network, but the attack still consumes all the bandwidth on the router's uplink.
It is much more efficient to make the provider blackhole the traffic, so that no attack traffic goes down the link to the stub AS. This also has the advantage of blackholing traffic closer to the attack source, and will potentially drop more attack traffic than if the stub AS was blackholing.

Figures 2.7 and 2.8 demonstrate such mechanism. Figure 2.7 depicts a server under a DDoS attack. To counter this attack, the operator can tell its provider to blackhole the attack (Figure 2.8), specifying the scope by either adding NO_EXPORT or NO_ADVERTISE to the blackholing advertisement.
If NO_ADVERTISE is added (Figure 2.8a), only the router receiving the advertisement will blackhole the traffic, as it will not propagate the advertisement even within the AS.
If NO_EXPORT is added (Figure 2.8b), the router receiving the advertisement will propagate the advertisement, but the advertisement will not be propagated to other ASes (AS Y). In this case, the attack is mitigated closer to the source when using NO_EXPORT.

Of course, accepting blackhole advertisements from other ASes introduces the risk that malicious ASes use this service for malevolent purposes, by sending illegitimate blackhole advertisements. Operators thus need to be cautious when accepting blackhole advertisements from BGP peers.

This subsection will describe operational considerations when blackholing, first within the local AS, and then between BGP peers.

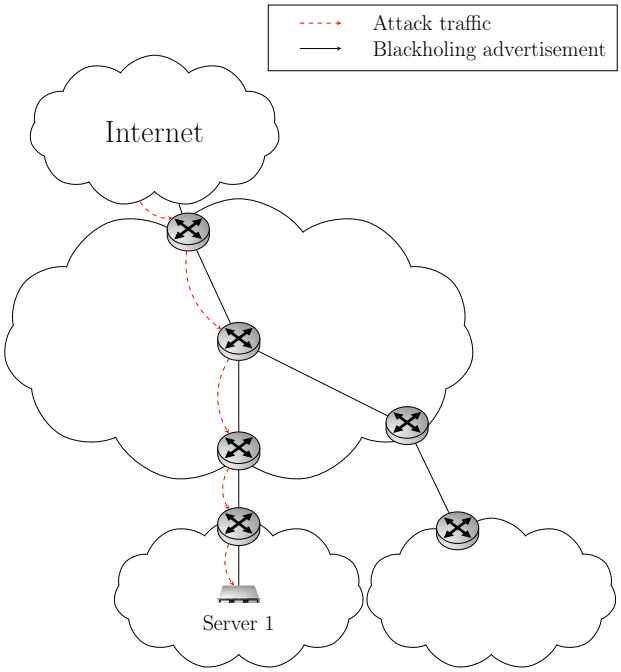
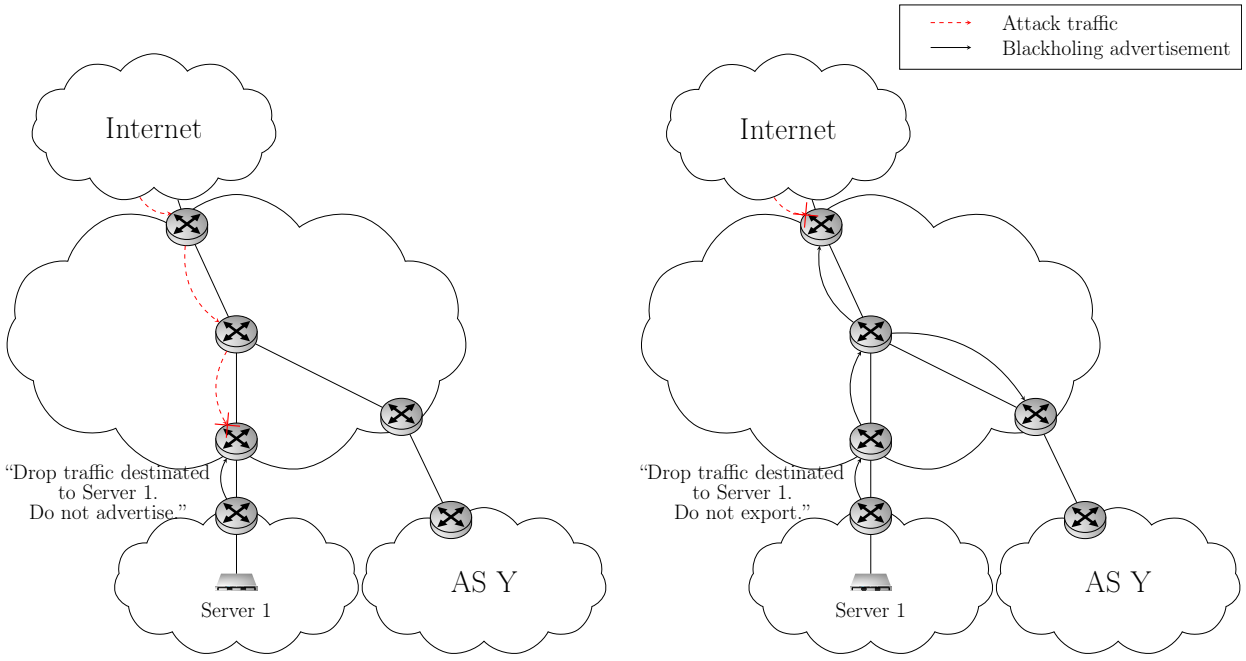


Figure 2.7: Server 1 under a DDoS attack



(a) Blackholing with NO_ADVERTISE (b) Blackholing with NO_EXPORT

Figure 2.8: Mitigation by blackholing at another AS

Blackholing within the local AS

When sending the BGP trigger, the NO_EXPORT community must be added, to ensure the advertisement does not cross AS boundaries [119, 67, 64].

An inbound filter must be set on routers receiving blackhole advertisements to only accept blackhole advertisements which were locally originated (Step 2 of Subsubsection 1.4.1), to ensure eBGP peers cannot activate blackholing in the local AS [119].

An outbound filter should be set on eBGP peering sessions to scrub communities used internally for blackholing (as well as the BLACKHOLE community), to ensure no blackholing is triggered by accident [67]. It is recommended to deny all prefixes longer than the longest prefix expected to be announced for the same reason [67]. The NO_EXPORT community included in blackholing advertisements should already ensure that blackholing is not triggered over eBGP sessions, those filters act as safety measures.

Note that with the match on locally generated advertisements, the local AS cannot be attacked by BGP peers using blackholing as an attack vector, but can still be vulnerable to other attackers combining blackholing with other attack vectors, such as the insertion of an advertisement on a BGP session [90, 7]. Those attacks are out of the scope of this document.

Blackholing across AS boundaries

The same good practices apply for peering with customers, peers or providers, but accepting blackhole advertisements from more peers also means being vulnerable to attacks from more peers.

[119] advises to either match only locally generated BGP advertisements or to accept only blackhole requests from customers. [67] advises the same rule, but specifies that the local AS must only accept blackhole requests from customers if the customer is authorized[8] to advertise that prefix. [64] does not give any particular restriction concerning from who blackhole advertisements are accepted, as long as the peer is authorized[8] to advertise the prefix. Ultimately, the decision is left up to the network operator, which needs to make the best compromise between flexibility and security.

In addition, [67] recommends to discard source-based RTBH requests coming from peers or customers, and only use source-based blackholing locally.

When sending the BGP trigger to an eBGP neighbor, the NO_EXPORT community must be added, to ensure the advertisement does not cross AS boundaries [119, 67, 64].

An inbound filter must be set on routers receiving blackhole advertisements, verifying multiple elements:

1. If the NO_EXPORT community is not present in the blackhole advertisement, the router must add the NO_EXPORT or the NO_ADVERTISE community, depending on whether the operator wants to keep blackholing local to the router, or propagate the advertisement to other routers in the network [64]. This choice is left up to the operator.

2. If the leftmost AS in the AS path does not correspond to the ASN specified in the BGP OPEN message that created the session, the BGP speaker should handle the route using the "treat-as-withdraw" approach, and may reset the BGP session if configured to do so [100, 18]. This mostly helps with Type-U hijacks when the attacker is not in the AS path.

3. The router needs to check if the prefix to be blackholed belongs to the AS sending the blackhole advertisement[9]. This can be done either with IRRs or with the RPKI. [64] does not give any details on how to treat validation information on a blackhole advertisement, specifying only that origin validation techniques must not block legitimate blackhole advertisements.

When peering with a route server, an AS needs to drop verification step 2, as the leftmost AS of the AS path will be the AS of the peer which sent the advertisement, not the AS of the router server [60]. This is not optimal from a security point of view as trust needs to be outsourced to the IXP in order to verify this step, even if IXP configurations are usually made public and thus enable peers to check that the IXP indeed verifies the legitimity of the peer.

In the same way, an IXP member wanting to blackhole a prefix needs to send its advertisement without the NO_EXPORT or NO_ADVERTISE community, as the router server will honor those communities and will not propagate the advertisement to other IXP members. The route server will set the NO_EXPORT community itself once it receives a blackhole advertisement.

An outbound filter should be set on eBGP peering sessions to scrub communities used internally for blackholing (as well as the BLACKHOLE community), unless explicitly configured to do so [67].

[8] This is verified using IRRs or the RPKI.

[9] This obviously cannot be checked in the case of source-based blackholing. Thus, if receiving a source-based blackhole advertisement from a peer, [67] advises to discard it.

Chapter 3

An attack taxonomy for blackholing

As we saw in Section 2.2, solutions to secure BGP do not secure communities. The IETF is aware of this (see Section 6 of [64]), warning the reader that *"a forwarding agent can alter, add or remove BGP communities"*. This is especially problematic in the case of the BLACKHOLE community, its misuse being capable of causing a "Denial of Service attack based on denial of reachability".

The IETF recommendations for limiting the impact of such unauthorized advertisements is to apply strict filtering (see Section 6.2.1.1.2 of [30]) to verify that the peer announcing the prefix is authorized to do so. Unfortunately, this solution is based on IRRs, which do not seem to be satisfying considering their issues (Subsection 2.2.1), and the fact that more sophisticated attacks cannot be detected by simply validating the origin or the AS path attribute of an advertisement.

This chapter presents a threat model for attacks using blackholing. We will go over different security deployments, and see which attacks are possible in each of them.

3.1 Threat model

This section is dedicated to the elaboration of a threat model. We consider a common and general hijacking threat model [110, 111]. An attacker controls a single AS and its border routers. He also has full control of the control plane and the data plane within its own AS. The attacker can arbitrarily manipulate the advertisements that it sends to its neighboring ASes and the traffic that crosses its network. He has no control over advertisements and traffic exchanged between two other ASes. The hijack taxonomy used in this section is defined in Section 2.1.1.

Tables 3.1 and 3.2 summarize security of BGP communities under different security deployments, the former against exact prefix hijacks and the latter against sub-prefix hijacks. Each line of the table details a security deployment scenario, and each column details an attack vector. Thus, the intersection of a line and a column shows how a particular security deployment fares against a given attack vector:

- ✓ - The security deployment is resistant to the attack vector
- - - The security deployment is not resistant to the attack vector
- ✓ / - - The resistance of the security deployment to the attack vector is determined by other factors (network topology, where security is deployed, ...)

Security Deployment	Hijack			On Path
	Type-0	Type-N	Type-U	
BGPsec (full)	✓	✓	✓	-
BGPsec (partial)	✓ / -	✓ / -	✓ / -	-
RPKI (full)	✓	-	-	-
RPKI (partial)	✓ / -	-	-	-
No security	-	-	-	-

Table 3.1: Security of communities against exact prefix hijacks

The next sections go over different deployments of BGPsec and RPKI, and describe the attacks possible in each context.

Security Deployment	Hijack		
	Type-0	Type-N	Type-U
BGPsec (full)	✓	✓	✓
BGPsec (partial)	✓ / -	✓ / -	✓ / -
RPKI (full)	✓	✓	✓
RPKI (partial)	✓ / -	✓ / -	✓ / -
No security	-	-	-

Table 3.2: Security of communities against sub-prefix hijacks

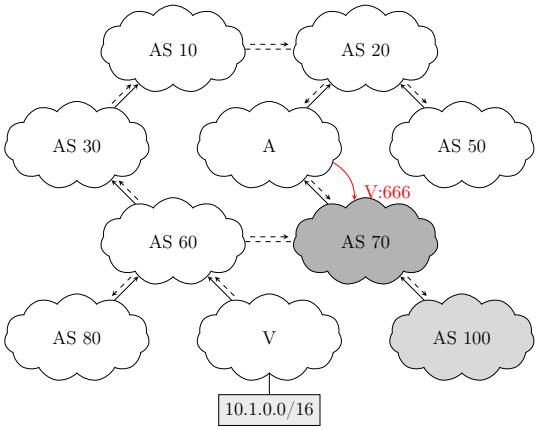


Figure 3.1: On Path blackholing attack

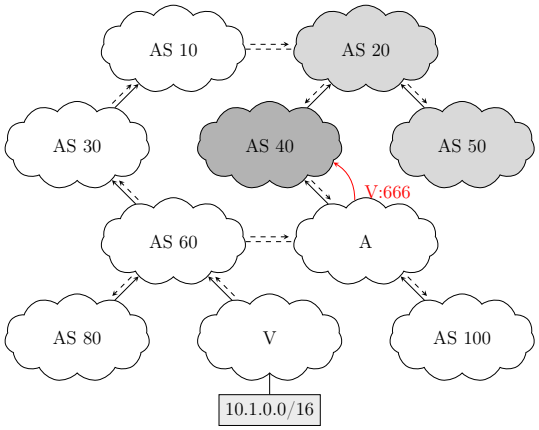


Figure 3.2: On Path blackholing attack with Gao-Rexford break

3.2 Fully deployed BGPsec

In this section, we consider a full deployment of BGPsec:

Assumption 1. Every AS has deployed, and uses, BGPsec according to best practices [71, 13].

If BGPsec is fully deployed, every AS can be assured that the AS path attribute is protected and legitimate in every advertisement they receive. An attacker thus cannot make use of exact prefix hijacks or sub-prefix hijacks, and must be directly on the path of an advertisement to conduct an attack via communities.

3.2.1 On Path blackholing

In the example in Figure 3.1, AS V advertises a route for 10.1.0.0/16. Full arrows represent customer-to-provider relationships. Dashed lines represent peer-to-peer relationships. Dashed arrows represent the propagation of this advertisement in the AS graph. We assume that every AS uses the Gao-Rexford routing policy model, detailed in Subsection 1.2.3:

Assumption 2. Every AS uses the Gao-Rexford routing policy model.

We also assume that ASes follow the best practices when receiving a blackholing request defined by Cisco [19]:

- Set local-preference to 200
- Set origin-type to IGP
- Add the NO_EXPORT community to the advertisement

Assumption 3. Every AS follows the best practices for RTBH defined by Cisco.

This assures that the blackholed route is preferred over other routes and that blackholing is done at every edge of the network.

Now, let us look at AS 70. AS 70 receives two routes to 10.1.0.0/16, one from its peer AS 60, and one from its provider AS A. AS 70 is following the Gao-Rexford routing policy model, and as such chooses the route going through its peer, directing traffic destinated to 10.1.0.0/16 from itself and AS 100 to AS 60.

AS A can decide to add a BLACKHOLE community to its advertisement for AS 70. Considering Assumption 3, AS 70 will choose the route coming from AS A over the route coming from AS 60, blackholing the traffic destinated to 10.1.0.0/16[1].

This attack has two main advantages over dropping only the traffic going through the attacker:

- Reach: The attacker can potentially drop more traffic by sending blackholing advertisements to its neighbors than by blackholing the traffic himself. In our example, AS A could not have dropped any traffic from other ASes, had he not performed the attack, because no traffic was going through it to begin with.
- Stealth: As the attacker is not the one dropping the traffic, he effectively obfuscates the source of the attack. Note that it may still be possible to retrieve the source of the attack by looking at the advertisements received by the relevant routers at the time of the attack. It is also worth noting that an even stealthier attack is possible, if the AS blackholing the traffic is at multiple hops from the attacker. If the attacker manages to target this AS only, not only will the attacker not blackhole the traffic himself, but he will also potentially not be the only one that could have added the BLACKHOLE community, as every AS between the attacker and the blackholer was capable of adding this community. In our example, AS 70 is the one blackholing the traffic, even though it was AS A that performed the attack.

Note that the attacker may break the Gao-Rexford export rule and propagate the advertisement in ways forbidden by those rules, which brings us to the second attack in this deployment.

3.2.2 On Path blackholing with Gao-Rexford break

Figure 3.2 has the same setting as the precedent attack, with only a change in the position of the attacker. We keep the same assumptions, but now also consider the fact that the attacker can break the Gao-Rexford export rule.

When receiving the route to 10.1.0.0/16 from AS 60, AS A can decide to forward this advertisement to its provider, AS 40, thus breaking the Gao-Rexford export rule. In doing so, AS A will attract the traffic having for destination 10.1.0.0/16 and coming from AS 40, AS 20 and AS 50, AS 20 and 40 preferring the route through their respective customers by following the Gao-Rexford import rule.

In a similar manner, AS A can then forward the advertisement to its provider but this time add a BLACKHOLE community. In this configuration, AS 40 will blackhole traffic coming from AS 20 and AS 50.

This attack which is in reality a combination of two attacks, the community blackhole attack and a break of the Gao-Rexford export rule, has the same advantages as the previous attack, only that it can reach much more ASes. Technically, the attacker could only break the Gao-Rexford export rule and drop traffic at his network, but he would then lose the stealth component of the attack.

Note that On Path attacks are a subtype of Type-U hijacks, where the attacker is in the AS path.

3.3 Partially deployed BGPsec

In this section, we drop Assumption 1 and consider instead a partial deployment of BGPsec:

Assumption 4. A subset of ASes have deployed, and use, BGPsec according to best practices [71, 13].

Depending on which ASes deployed BGPsec and which ASes use RPKI and **Route Origin Validation (ROV)**, multiple cases arise. An attacker may be able to use Type-0 hijacks if no one on the path of the advertisement is performing ROV, or if the prefix is not in the RPKI.

If at least an AS on the path of the advertisement is using ROV and the prefix is in the RPKI, but no AS on the path is using BGPsec, a potential attacker cannot use Type-0 hijacks, but can carry out Type-N and Type-U hijacks.

Once again, an attacker directly on the path of the advertisement can alter the communities of this advertisement, thus both On Path attacks are possible in this deployment.

[1] Note that this also blackholes traffic coming from AS 100.

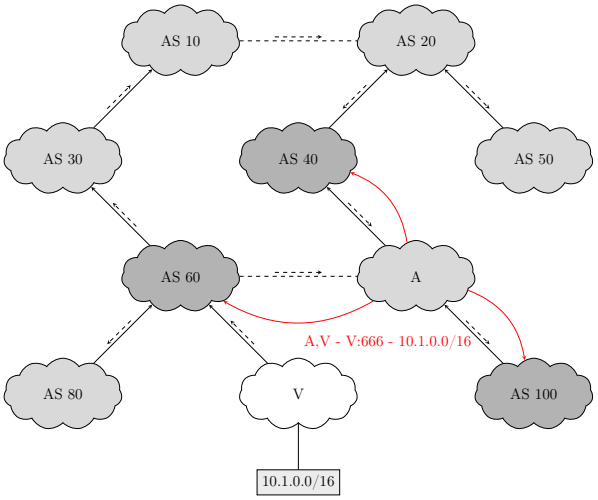


Figure 3.3: Type-N hijack blackholing attack

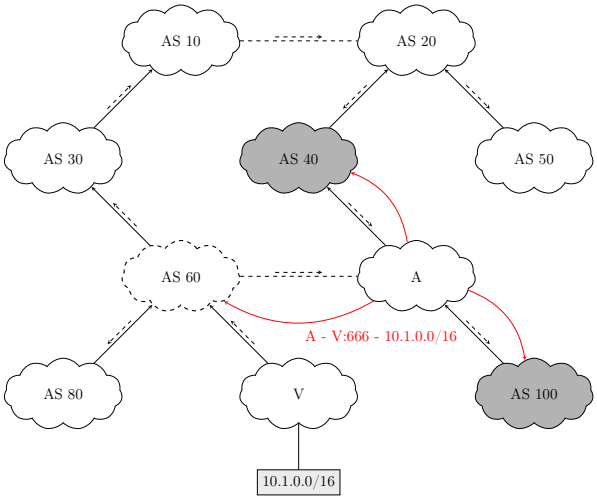


Figure 3.4: Type-0 hijack blackholing attack

3.4 Fully deployed RPKI

In this section, we drop Assumption 4 and consider instead a full deployment of RPKI and ROV:

Assumption 5. Every AS has deployed, and uses, RPKI and ROV according to best practices [69, 51, 87].

If RPKI and ROV are fully deployed, every AS can verify the association of the advertised prefix and the AS originating it. In this setting, an attacker can obviously carry out both On Path attacks, in the same way as with the BGPsec deployment, but he can do more. An attacker can indeed carry out a Type-U or Type-N hijack blackholing, which will not be detected by ROV, as the Origin AS is left unaltered.

3.4.1 Type-N hijack blackholing

In Figure 3.3, the attacker can perform a Type-N hijack blackholing, by advertising a false connection to AS V. As the origin is "valid", ROV will not detect this kind of attack. Interestingly, as the preference for blackholing is higher than normal advertisements [19], AS 60 prefers the blackholing advertisement, where it would not prefer a simple Type-N hijack because of the longer AS path, and thus also blackholes the traffic, preventing all other ASes except AS V from reaching 10.1.0.0/16.

3.4.2 Type-U hijack blackholing

An attacker can also perform a Type-U hijack[2]. The only difference with Figure 3.3 is that the attacker advertises the AS path {ASV} instead of {ASA,ASV} This attack has the same effects as a Type-N hijack, but it is stealthier, as the AS of the attacker is not in the AS path. It also might have more reach due to the shorter AS path.

Since everyone is using RPKI and ROV according to best practices (from Assumption 5), Type-0 hijacks and sub-prefix hijacks are not possible.

3.5 Partially deployed RPKI

In this section, we drop Assumption 5 and consider instead a partial deployment of RPKI and ROV:

Assumption 6. A subset of ASes have deployed, and use, RPKI and ROV according to best practices [69, 51, 87].

In the same way as if RPKI is fully deployed, to tamper with the communities in an advertisement, an attacker can be on the path of the advertisement, or can carry out either a Type-U or a Type-N hijack, which will not be detected by ROV, as the Origin AS is left unaltered.

[2]Here, we consider Type-U hijacks where the attacker is not in the AS path, those being the On Path attacks discussed in Section 3.2.

3.5.1 Type-0 hijack blackholing

The attacker can also make use of Type-0 hijacks in certain cases. If ASes receiving the forged advertisement do not perform ROV, they accept the advertisement (no ROV = no security).

Figure 3.4 depicts such an attack. AS A sends a forged advertisement (A - V:666 - 10.1.0.0/16) to its neighbors. AS 60 is performing ROV, and thus classifies the advertisement as "invalid", rejecting it. AS 40 and AS 100 do not use ROV, and accept the forged advertisement, blackholing traffic towards 10.1.0.0/16. Depending on who implemented ROV, this attack vector has more or less reach, as each AS can accept or drop the advertisement depending on its configuration. This is the reason why the intersection of RPKI (partial) and Type-0 hijack in Table 3.1 is ✓/ -.

If RPKI is only partially deployed, other cases arise. What is possible to achieve depends on three factors:

- The presence (or absence) of the prefix to be attacked in the RPKI.
- If the prefix is in the RPKI, the fact that the ROA for the prefix is loose[3] or not.
- The presence (or absence) of ROV at the AS receiving the advertisement.

Table 3.3 details those possibilities.

	P in RPKI		P not in RPKI
	ROA is loose	ROA is not loose	
ROV	Sub-prefix HJ (Type-N and Type-U)	✓	AS policy
no ROV	-	-	-

Table 3.3: Detail of security of communities against hijacks

As you can see in Table 3.3, if the AS receiving the forged advertisement does not enforce ROV, it is the same as if there is no security.

Second, if the AS receiving the forged advertisement enforces ROV, and P is not in the RPKI, it is up to the AS receiving the advertisement (RPKI validation state = "unknown") to decide what to do. The AS can either diminish its preference of the route, or drop the route. In the former case, exact prefix hijacks (of all types) will be possible as the AS classifies all routes to this prefix as "unknown", even the one from the legitimate AS: Hijacks can win the BGP decision process. Sub-prefix hijacks are also possible, and are not even penalized by a diminished preference, as they are more specific than the legitimate advertised prefix. All in all, for prefixes not in the RPKI, an AS enforcing ROV and lowering preferences for "unknown" route validity states behaves in the same way as an AS not enforcing ROV. In this case, attack vectors are the same as if there was no security (see Section 3.6). In the latter case, those attacks are no longer possible. Note that in the current deployment state of RPKI, dropping "unknown" routes equates to dropping routes to 95% of the Internet, so for now, a compromise between reachability and security must be made.

Third, if the AS receiving the forged advertisement enforce ROV, and P is in the RPKI, two cases arise. Either the ROA for the prefix is loose, or it is not. If the ROA is not loose, there are no new attack vectors (compared to a full deployment of RPKI). If the ROA is loose, an attacker can perform Type-N and Type-U sub-prefix hijacks within the range of maxLength, as the Origin AS will match the asID in the ROA, and the sub-prefix is within the range of maxLength.

3.6 No security

In this section, we drop Assumption 6 and consider instead that no security mechanisms are deployed at all:

Assumption 7. ASes do not use any of the aforementioned security mechanisms.

If neither BGPsec nor RPKI and ROV are deployed, an attacker can perform all the attacks of the taxonomy.

[3]Not all sub-prefixes of the maximum length allowed by the ROA are advertised in BGP [40].

Chapter 4

Securing Blackholing

As we saw in Chapter 3, allowing blackholing between ASes can lead to serious damage. Good practices already in the literature (2.3) can prevent some, but not all of the attacks in the taxonomy. Additional mechanisms are thus necessary to protect against the rest of these attacks.

In this chapter, we first present how good blackholing practices impact the attack taxonomy (Section 4.1). We then detail additional good blackholing practices (Section 4.2) necessary to protect against the rest of the attack taxonomy, and a simple integration of communities into BGPsec (Section 4.3) to protect against such attacks.

4.1 Protection assured by good blackholing practices

The three items having an influence on preventing attacks from the taxonomy are as follows:

- Legitimate peer: The peer sending the blackholing advertisement is legitimate if the leftmost AS in the AS path is the ASN specified in the BGP OPEN message that created the session[1].
- Authorized origin: The origin is authorized if the association between the Origin AS and the prefix is "valid" according to IRRs or the RPKI[2].
- Valid path: The path is considered "Valid" if the AS path reflects the actual path the advertisement went through. This can be verified using BGPsec.

Concerning the security benefits provided by those rules:

- The "Legitimate peer" rule protects against Type-U hijacks because the leftmost AS of a Type-U hijack is not legitimate by definition (see Section 2.1.2).
- The "Authorized origin" equates to ROV, and protects against Type-0 hijacks (see Section 3.4).
- The "Valid path" rule equates to path validation, thus protecting against all but On Path attacks and Type-0 hijacks (see Section 2.2.2).

The first part of Table 4.1 summarizes the protection provided by those good practices, depending on which of them are implemented. Note that since the "Valid path" rule is basically an extension of the "Legitimate peer" rule, verifying all the AS path instead of just the leftmost one, combinations including "Legitimate peer" and "Valid path" do not benefit from the "Legitimate peer" rule.

As the first part of table 4.1 highlights, even with all three good practices implemented, and in accordance with the explanations of Section 3.1, On Path attacks are still not prevented. It is also worth noting that, acknowledging the deployment state of RPKI, an AS peering through an IXP virtually has no protection against the attacks, as it must trust the IXP to verify the "Legitimate peer" rule and the route server will most likely not perform ROV[3].

4.2 Additional good blackholing practices

As demonstrated, not all attacks are prevented by following even all good practices present in the literature, to the best of our knowledge. New rules thus need to be put in place.

[1] Reminder: ASes peering with a route server cannot verify this rule, they need to trust the IXP.
[2] While an AS can accept "unknown" origins, they do not count as authorized, and do not verify this rule.
[3] This might be changing as several IXPs now seem to implement ROV [3].

4.2.1 An additional rule to prevent attacks

In addition to the three rules of Section 4.1, we propose to introduce a new rule:

- Direct connection: The AS sending the blackholing advertisement is directly connected to the local AS. This can be verified by making sure there is only one AS in the AS path.

With this rule, Type-N hijacks are not possible by definition, because it would mean the AS path contains at least two ASes. On path attacks are also not possible, or rather, they are reduced to Type-U hijacks.

The second part of Table 4.1 provides additional combinations with the new rule. Looking at Table 4.1, it appears the minimal set of rules to follow in order to be protected against all the attacks of the taxonomy is {"Legitimate peer"; "Authorized origin"; "Direct connection" }. An AS following at least those rules should prevent any of the attacks in the taxonomy. Note that the "Direct connection" rule is mandatory to be fully protected: As soon as there is more than one AS in the AS path, one cannot determine which AS attached the blackholing community, and other mechanisms are needed (Section 4.3).

	On Path	On Path GR break	Hijack		
			Type-N	Type-U	Type-0
No rule	-	-	-	-	-
Legitimate peer	-	-	-	✓	-
Authorized origin	-	-	-	-	✓
Valid path	-	-	✓	✓	-
Legitimate peer Authorized origin	-	-	-	✓	✓
Legitimate peer Valid path	-	-	✓	✓	-
Authorized origin Valid path	-	-	✓	✓	✓
Legitimate peer Authorized origin Valid path	-	-	✓	✓	✓
Direct connection	✓	✓	✓	-	-
Legitimate peer Direct connection	✓	✓	✓	✓	-
Authorized origin Direct connection	✓	✓	✓	-	✓
Valid path Direct connection	✓	✓	✓	✓	-
Legitimate peer Authorized origin Direct connection	✓	✓	✓	✓	✓
Legitimate peer Valid path Direct connection	✓	✓	✓	✓	-
Authorized origin Valid path Direct connection	✓	✓	✓	✓	✓
Legitimate peer Authorized origin Valid path Direct connection	✓	✓	✓	✓	✓

Table 4.1: Protection assured by good practices

4.2.2 Additional good blackholing practices

In addition, other good practices can be put in place. Those good practices help to limit the possible damage caused by an inadvertant blackholing advertisement.

An outbound filter for more specific blackholing advertisements

When using blackholing across AS boundaries, an outbound filter should be set on eBGP peering sessions to deny all prefixes longer than the longest prefix expected to be announced, unless that prefix is tagged with a blackhole community. This does not really help with accidental blackholing directly, but prevents an AS from advertising more specific prefixes inadvertently.

A filter for less specific blackholing advertisements

The literature specifies that operators should accept blackholing advertisements up to /32 for IPv4, and /128 for IPv6, but does not specify a limit on prefixes which are less specific. We propose that operators reject blackholing advertisements if they are not specific enough, in order to avoid accidental blackholing of large IP blocks.

Some operators put this limit at /8 [26, 34], while others are more restrictive and put the limit at /24 [49] or /25 [20, 89, 32]. As per usual, the operator decides on the value the limit should take, but acknowledging the distribution of blackholing prefix length [28], we advise to set it to /24, thus only accepting blackholing advertisements from /24 up to /32.

Concerning IPv6, observed IXPs put the limit at /19 [26, 34]. The literature does not have any specific information enabling us to determine a good limit for IPv6 blackholing prefix specificity, more research needs to be done.

This filter can be applied as both an inbound and outbound filter.

An inbound filter on the outcome of ROV

Subsection 2.3.3 outlines the need, when blackholing across AS boundaries, to check if the prefix to be blackholed belongs to the AS sending the blackhole advertisement. This can be done either with IRRs or with the RPKI, but [64] does not give any details on how to treat validation information on a blackhole advertisement, specifying only that origin validation techniques must not block legitimate blackhole advertisements.

Acknowledging the literature, the current deployment of RPKI and the state of IRRs, we recommend to follow this set of rules when considering the outcome of route origin validation:

- If the outcome is "valid", the blackhole request can be accepted.
- If the outcome is "unknown", the blackhole request can be accepted.
- If the outcome is "invalid", the blackhole request must be discarded.

Note that accepting "unknown" blackholing advertisements exposes the AS to potential Type-0 hijacks.

An inbound filter on the outcome of BGPsec validation

When using BGPsec, IXPs are visible in the AS path, which means ASes can use the "Legitimate peer" rule even if the peer is a route server.

To the best of our knowledge, there are no considerations for blackholing in the literature concerning BGPsec. When considering the outcome of the BGPsec validation process, we advise to:

- Accept the advertisement if the outcome is "Valid".
- Discard the advertisement if the outcome is "Not Valid".

While certain combinations of good practices can provide sufficient protection against the attacks of the taxonomy, some of these good practices might not be applicable depending on the situation[4] (See remarks at the end of Section 4.1). Considering this, one might consider to add an extension to BGPsec as an alternative to protect against attacks of the taxonomy.

[4]For example, it cannot be assumed that the AS path will only contain one AS.

4.3 A BGPsec solution

Instead of relying on good practices, one could extend BGPsec to secure communities as well. If BGPsec is used with RPKI, only On Path attacks are still possible. Thus, the goal of integrating communities to BGPsec is to be able to attribute the changes made to communities to an AS. This attribution is crucial for blackholing, because it allows an AS to accept or reject a blackhole request based on the identity of the AS requesting the blackhole. A blackholing advertisement can then be analyzed, to determine the source of the request, and a decision can be made based on whether or not this AS has a right to blackhole this prefix. Moreover, given an unwanted blackholing event, those responsible for it can be held accountable.

To be able to add communities to BGPsec, we need to modify the BGPsec_PATH attribute to create a new attribute we call the BGPsec_PATH_COMMUNITIES attribute.

The main reason to secure the AS path and communities together is that secure communities are not useful if they cannot be reliably attributed to an AS. Evidently, if communities are secured, but the AS path is not, there is not guarantee that communities can be attributed to an AS, thus failing to fulfill the purpose of secure communities in the first place. Hence, secure communities only make sense if the AS path is also secured.

4.3.1 A new attribute for BGPsec

The BGPsec_PATH_COMMUNITIES attribute will replace the AS_PATH attribute. This attribute also replaces the BGPsec_PATH attribute. That is, UPDATE messages that contain the BGPsec_PATH_COMMUNITIES attribute must not contain the BGPsec_PATH attribute or the AS_PATH attribute.

The BGPsec_PATH_COMMUNITIES attribute works in the same way as the BGPsec_PATH attribute. Figure 4.3.1 describes the BGPsec_PATH_COMMUNITIES attribute. As you can see, the only change to the BGPsec_PATH attribute (Figure 2.5) is the addition of a Secure_Communities field. There needs to be one Secure_Communities Segment per Secure_Path Segment. Any deviation means the attribute is malformed and needs to be discarded.

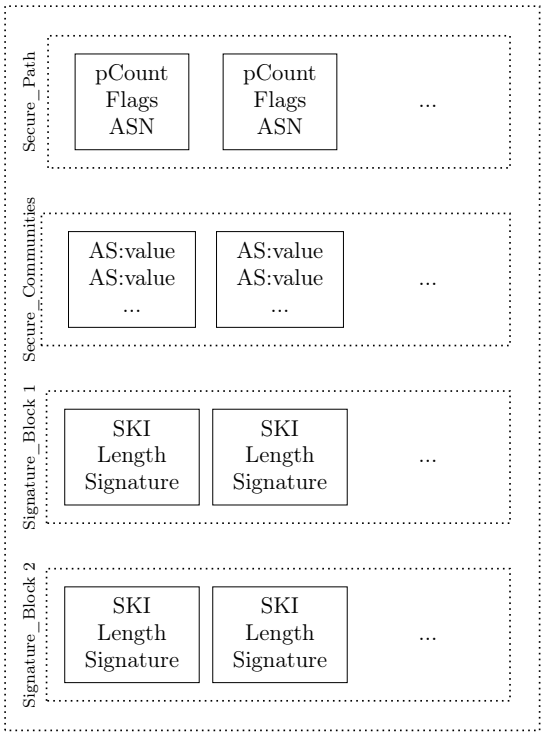


Figure 4.1: The BGPsec_PATH_COMMUNITIES attribute

Target AS Number	
Signature Segment	: N-1
Secure_Communities Segment	: N
Secure_Path Segment	: N
...	
Signature Segment	: 1
Secure_Communities Segment	: 2
Secure_Path Segment	: 2
Secure_Communities Segment	: 1
Secure_Path Segment	: 1
Algorithm Suite Identifier	
AFI	
SAFI	
NLRI	

Table 4.2: Sequence of Octets to Be Hashed

The Secure_Communities field (Table 4.3) is composed of a 2 octet Secure_Communities Length field indicating the total length of the Secure_Communities attribute (in octets), followed by one

or more Secure_Communities Segments, each of which composed by a set of **type-length-value (TLV)** fields. The type field indicates the type of communities (regular, extended or large) contained in the value field whereas the length field indicates the length of the value field. The value field is thus a set of communities.

Secure_Communities Length	(2 octets)
One or more Secure_Communities Segments	(variable)

Table 4.3: Secure_Communities Format

Now that the attribute is defined, we need to know how it will be constructed.

4.3.2 Constructing the BGPsec_PATH_COMMUNITIES attribute

Constructing the attribute implies we need to define how to generate it, how to update it when an AS wants to forward an advertisement containing such an attribute, and how to leverage the information provided by the attribute in the decision process for blackholing.

The process to generate a BGPsec_PATH_COMMUNITIES attribute is the same as for the BGPsec_PATH attribute, except that it also creates a Secure_Communities Segment when creating the Secure_Path Segment. The sequence of elements to be hashed (Table 2.2) must also be modified to take the Secure_Communities Segment into account, resulting in the sequence depicted in Table 4.3.1. This way, an AS signing a BGPsec advertisement asserts that it received the indicated communities, and chose to propagate the advertisement to the peer specified by the Target AS Number, with the indicated communities.

The same sequence of elements is reconstructed to verify a received BGPsec advertisement. If a BGPsec_PATH_COMMUNITIES attribute already exists in a received advertisement, the AS receiving it prepends its new Secure_Communities Segment onto the existing Secure_Communities. This means the set of communities to consider when receiving an advertisement is thus the communities present in the leftmost Secure_Communities Segment. This mimics the current behavior of communities in BGP and BGPsec, but one could also consider communities in other Secure_Communities Segments.

Now that we defined how to construct the attribute, let us see how it would work in an example.

4.3.3 A working example

Figure 4.2 depicts how such an extension would work. AS 10 originates 10.1.0.0/16, and as such signs a BGP advertisement containing this prefix, the AS path (10), the communities (C1) and the ASN of the recipient of the advertisement (20) with its private key. When receiving the advertisement, AS 20 can then verify the information present in this advertisement by using AS 10's public key. To forward the advertisement to other ASes, AS 20 adds itself to the AS path, adds its communities and specifies it wants to send the advertisement to AS 40. After signing the appropriate sequence of elements, AS 20 can send the advertisement to AS 40, which will go through the same steps to forward it to AS 50.

Again, if AS 20 wanted to forward the advertisement to AS 30, it would have needed to generate another advertisement specific to AS 30.

Blackholing

With such an extension to BGPsec, accepting blackhole advertisements from BGP peers could be a lot simpler. As we know which AS introduced which communities, an AS could simply generate a table associating an ASN to a set of prefixes this AS is authorized to blackhole. This table could be populated by RPKI/IRR data, but also manually with trusted peers, or other associations the operator deems relevant.

Then, this AS could accept a blackholing request if the AS requesting the blackhole and the prefix in the advertisement matches an association in the table. With this method, the AS performing the check can consider all Secure_Communities Segments, not just the leftmost one. If a blackhole community present at some point in the Secure_Communities Segments is removed in newer

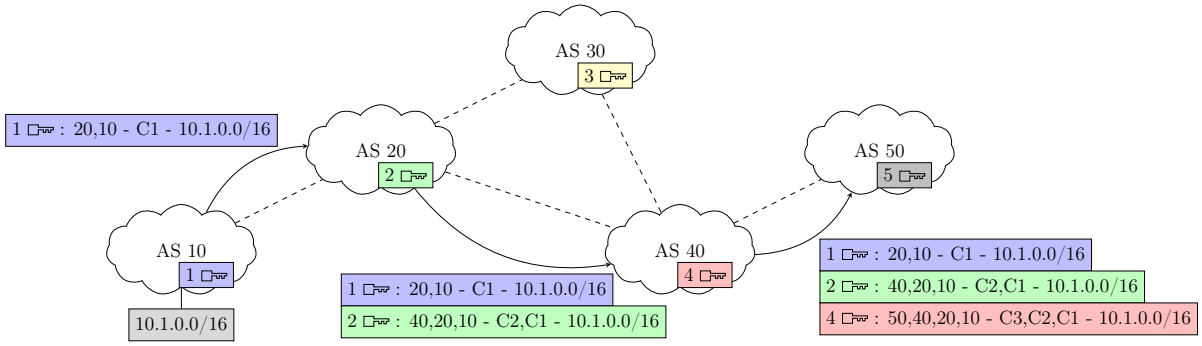


Figure 4.2: Modified BGPsec advertisement propagation

Secure_Communities Segments, the AS could still accept the blackholing request, unless specified otherwise.

Note that with this method, an AS could consider accepting blackholing requests originating more than one AS hop away.

Of course, filters preventing accidental blackholing should not be neglected. Even if we know which AS introduced which communities, errors and misconfigurations can occur.

4.3.4 Additional considerations

Reconstructing the COMMUNITIES attribute

As mentioned previously, if a BGPsec speaker wants to forward a BGPsec advertisement to a non-BGPsec speaker, it needs to "downgrade" the BGPsec advertisement to a normal BGP advertisement. To do this with our modified version of BGPsec, an AS needs to be able to reconstruct the COMMUNITIES attribute (in addition to the AS path [71]). This is very straightforward, as we only need to consider the leftmost Secure_Communities Segment of the advertisement: The appropriate set of communities corresponds exactly to the value field of the TLV whose type field indicates the use of regular communities. The AS can then modify the newly generated COMMUNITIES attribute before forwarding it to the non-BGPsec speaker.

A similar reconstruction process can be applied for extended and large communities.

Note that as this extension to BGPsec behaves in the same way, it is vulnerable to the same attacks. As such, this extension is still vulnerable to downgrade attacks [73].

Using the COMMUNITIES attribute

Operators may still use the COMMUNITIES attribute when some of the communities they wish to send are meant to be used by the direct peer only. This helps with reducing the size of the BGPsec_PATH_COMMUNITIES attribute, and also provides some level of opacity, as communities transmitted via the BGPsec_PATH_COMMUNITIES attribute will be seen by all the ASes receiving the advertisement containing this attribute. Note that communities transmitted using the standard COMMUNITIES attribute will not benefit from the security provided by the BGPsec_PATH_COMMUNITIES attribute.

Similar considerations apply to extended and large communities.

Residual vulnerabilities

With BGPsec and RPKI, the only attacks still possible in our attack taxonomy are On Path attacks. Subsubsection 4.3.3 gives an idea of how one could defend against those remaining attacks, but more importantly, adding communities to BGPsec allows an operator to identify a responsible AS in case of an unwanted blackholing event. This way, those responsible for the unwanted blackholing event can be held accountable.

On the viability of this extension to BGPsec

With the addition of communities to BGPsec, our modified version most probably performs worse than BGPsec.

BGPsec performs worse than BGP overall. Advertisements are larger due to the added security information. Advertisements are more numerous because each of them only carries a single prefix. Processing advertisements is slower because of the validation process.

Our modified version of BGPsec will make advertisements a lot larger due to the added community information (One set of communities per AS in the AS path). Processing advertisements will also probably be slower than standard BGPsec.

To mitigate slow processing speeds, it is likely that BGPsec speakers will incorporate dedicated cryptographic hardware to take care of the validation process.

Even though performance seems to be an issue, performance measurements suggest the increase in processing time is not an issue [80]. This will need to be confirmed once BGPsec is actually deployed.

In any case, security always comes at a price, and it is up to the operators to decide if the increase in security BGPsec provides is worth a slight decrease in performance.

An alternative implementation

When considering our proposed implementation, one could wonder why we modified the BGPsec_PATH attribute to include communities, instead of creating a new attribute altogether. Creating a new attribute, BGPsec_COMMUNITIES consisting only of Secure_Communities Segments would mean that we would need to add two additional Signature_Blocks, in order to be able to verify the authenticity and integrity of the attribute, increasing the size of our already large advertisements.

Another problem arising is that to make use of the BGPsec_COMMUNITIES attribute, we need to correlate Secure_Communities Segments with Secure_Path Segments, which is not as trivial as with our BGPsec_PATH_COMMUNITIES attribute. A way of verifying this correlation could be by checking that each Secure_Communities Segment corresponding to its Secure_Path Segment have the same SKI.

A problem with the BGPsec_PATH_COMMUNITIES implementation is that it is not compatible with BGPsec. An AS cannot forward an advertisement containing a BGPsec_PATH_COMMUNITIES attribute to a BGPsec speaker which does not handle BGPsec_PATH_COMMUNITIES. The AS cannot reconstruct the BGPsec_PATH attribute either, as that would require all the keys needs to generate the Signature_Blocks again. Thus, an advantage of creating a new attribute is that the BGPsec_PATH attribute is not modified, enabling compatibility with BGPsec.

Chapter 5

A tool to detect potential attackers

In this chapter, we develop a tool to detect potential attackers using the attack taxonomy defined in Chapter 3.

5.1 Goal of the tool

Given a topology (ASes, their security deployment and the relationships between them), an AS path and a detector AS receiving said AS path, the tool tries to classify which ASes in the AS path might be responsible for a blackholing attack, and of which type.

This tool can prove to be especially useful when operators keeps logs of blackholing advertisements they received, as the tool can essentially be fed the topology known to the operator, and the AS path contained in the advertisement, and give the operator a list of potential entities responsible for the unwanted blackholing.

As we use the attack taxonomy defined in Chapter 3, the tool assumes the same threat model and inherits the same limitations. For example, the tool does not take into account cases where the attacker possesses two or more ASes.

5.2 Methodology

To guarantee the tool gives accurate results, we make three assumptions:

Assumption 8. The detector knows the topology.

Assumption 9. The detector knows the relationships between ASes.

Assumption 10. The attacker is in the AS path.

Those assumptions might not prove to be right in real life situations, as an AS wanting to detect blackholing attacks might not know the full topology and all the relationships between ASes, but only a part of it. Furthermore, the attacker might not be on the AS path at all, performing for example a Type-U hijack. Further iterations of this tool will need to take those considerations into account when making predictions, so that our assumptions can be relaxed.

For each security deployment, the tool is given a topology file, a security deployment file, a detector AS and an AS path.

The topology file contains topology information. Each line consists of a triplet {AS, AS, relationship}, where the ASes are nodes and the relationship is the peering type between those ASes. The relationship can either be provider-to-customer, in which case the relationship field will be set to 1, or it can be a peer-to-peer relationship, in which case the relationship field is set to 2.

The security deployment file contains information about the security level of each AS. Each line consists of a pair {AS, security level} associating an AS with its security level:

- 0: No security.
- 1: RPKI and ROV.
- 2: RPKI, ROV and BGPsec.

The following section gives an example of how the tool works.

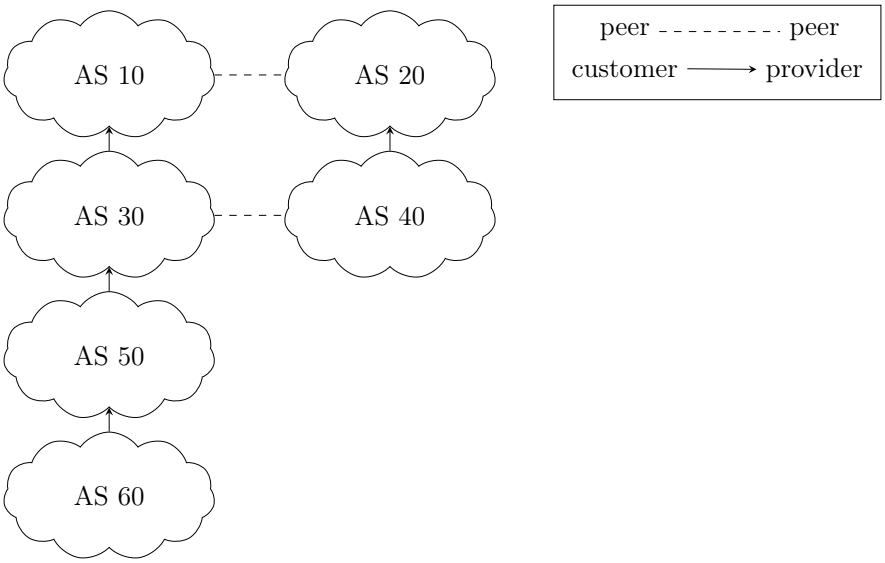


Figure 5.1: Topology used to demonstrate the tool

5.3 Example

Figure 5.1 depicts the topology we will use in our example. All ASes in the topology have a security level of 0, meaning they use none of the security mechanisms described earlier.

Now, let us imagine that AS 20, the AS we choose as our detector, receives the blackholing advertisement {AS40,AS30,AS50 - P}, where P is the prefix and {AS40,AS30,AS50} is the AS path.

We can give our tool this information via this command:

```
$ python attack_detection_tool.py topology.dat security_deployment.dat "20" "40␣
    ↪ 30␣50"
```

Given this input, our tool classifies AS 50 as a potential Type-0 attacker because the detector cannot verify that the Origin AS is authorized to advertise P. Our tool also classifies AS 30 as a potential On Path attacker, and AS 40 as a potential On Path with Gao-Rexford break attacker (because AS 40 forwarded the advertisement coming from a peer to a provider).

Appendix B gives more details on how the tool behaves in different security deployments.

5.4 Additional details

The tool can display the topology in matplotlib using the "-c" option, and has different verbosity levels via the "-v" option.

For the deployments using RPKI and ROV (full BGPsec, full RPKI and partial RPKI), we need to provide the tool with a list of the ASes authorized to advertise the prefix. This is done via the "-r" option. Omitting this option will make the tool consider that the prefix is not present in the RPKI at all.

Chapter 6

Conclusion

In this chapter, we will go over the main contributions of this thesis. We will then try to give some perspectives of future work.

6.1 Contributions

The purpose of this thesis was to study how blackholing, a technique used to mitigate DDoS attacks, could be used with malicious intent in routing attacks. After proposing an attack taxonomy based on a common and general hijacking threat model [110, 111], we tried to define additional good practices, as those present in the literature could not prevent all of the attacks. We then proposed an extension to the BGPsec protocol, introducing a way to attribute changes in the communities attribute to ASes. We also prototyped a tool whose goal was to detect potential attackers, which can be useful to determine which ASes might be responsible for an unwanted blackholing event.

6.2 Perspectives

The main contributions of this thesis are theoretical. Although they are based on best practices, standards and scientific literature, experiments must be made to confirm our theoretical contributions. The next logical step should then be to confirm that the attacks of Chapter 3 are possible in a real setting, using for example a similar methodology to that of [102].

Another way of expanding our work is to expand the threat model [110, 111] our work is based upon. As we have already pointed out, the threat model can be extended to account for attackers possessing two or more ASes. A way to extend our attack taxonomy is to consider attacks which try to remove blackholing communities from advertisements, denying legitimate blackhole request.

In Chapter 4, we proposed an extension to the BGPsec protocol. There is room for improvement, as we only specified the format of our attribute, its basic behavior as well as the way to handle it. To complete this specification, we need to take into account operational and management considerations, in order to determine undefined behavior; for example, the way confederations must process the attribute is to be defined. As this is an extension to BGPsec, a lot of the relevant considerations for our proposed implementation might already be present in the BGPsec literature.

Aside from completing the specification, using such an extension to BGPsec opens up other possibilities, such as accepting blackholing requests originating more than one AS hop away. Considerations for accepting such requests are still to be defined.

Moreover, the tool we designed is only a prototype. Once again, there is still plenty of room for improvement, accounting for more cases and more realistic scenarios. Possible improvements include taking into account a partial deployment of BGPsec, simulating a more realistic topology with routers, instead of considering an AS as a simple node in the graph, or even relaxing our assumptions by taking into account the local view of the detector, or considering that the attacker might not be present in the AS path. More generally, everything making the simulation more realistic, like simulating good practices and taking them into account to find potential attackers, can be an improvement to the tool. An extension of the threat model can also lead to an extension of the tool.

During this internship, I had the opportunity to work on a convoluted but interesting subject. Studying the security aspects of BGP requires taking into account a lot of different elements, namely

networking, attacks vectors and security solutions, and considering how those elements interact with one another.

Over the course of this intership, I gained a deeper understanding of the way trust works in the Internet, and of the current state of Internet security. For a given AS to be able to trust BGP advertisements, it relies on trusted authorities. While this may simplify the problem of trust in BGP, relying on a trusted authority creates a whole other set of attack vectors [23]. With the ultimate goal of securing Internet communications in mind, perhaps it is better to focus our efforts or securing data delivery [125].

I think I also grasped how difficult it was to make good and accurate measurements in the domain of networking, as well as the need for controlled experiments. Public datasets only depict an incomplete view of the Internet, and it can sometimes be impossible to determine the cause of an effect with uncontrolled experiments.

To conclude, I would like to thank Prof. Cristel Pelsser and Associate Prof. Stéphane Cateloin for their help and guidance, as well as all the people who provided me with their valuable comments, this work would not have been possible without them.

Appendix A

The Resource Public Key Infrastructure

A.1 Ghostbuster Records

Ghostbuster Records [12] are optional routing objects. As it was said earlier, the RPKI does not contain any information concerning the identity of the replying parties. How does one contact a RP if a problem arises (expired certificate, malformed ROA, ...)? This is what Ghostbuster Records are for. A Ghostbuster Record is essentially a simplified vCard [93], containing a name, an organisation, and at least one of the following: postal address, voice and/or fax phone, Email address.

A.2 Certificate Revocation Lists

RPs also need to be able to revoke certificates and routing objects. As it was said earlier, they do so by directly revoking the certificate in the case of CA certificates, or by revoking the EE certificate associated with the object in the case of routing objects.

To keep track of revoked objects, RPs use a CRL. There is only one CRL by Certification Authority. The URI of the CRL is indicated in the CA certificate of the entity issuing the CRL as the CRLDP.

For more details on how to publish objects in the RPKI, see [123]. For more details about the file naming scheme of routing objects, the contents of a publication point or a structure for local caches, see [50].

A.3 Publication points

To manage and maintain each publication point, several functionalities must be put in place, namely downloading, uploading , modifying and deleting. Functionalities modifying the repository (addition, deletion, modification) must support verification of the authorization of the entity performing said modification. publication points must be accessible via rsync, although other download protocols may also be supported. RPs can pull data from the publication points at whatever frequency they deem appropriate.

Of course, to provide acceptable service, publication points must guarantee availability and integrity. Availability is assured through replication of the databases. It is worth noting that except the verification of entities modifying objects in the repository, publication points do not offer any security. Instead, integrity is verified by local caches.

Indeed, the validation chain provided by CA certificate and EE certificates protect an RP against addition and modification of routing objects. The validation chain cannot protect against the deletion of routing objects, or the substitution of valid objects by their older non-expired non-revoked version. This explains the creation of a new routing object: the manifest.

A.4 Manifests

“A manifest is a signed object listing of all of the signed objects (except for the manifest itself) issued by an authority responsible for a publication in the repository system. For each unexpired certificate, CRL or ROA issued by the authority, the manifest contains both the name of the file

containing the object, and a hash of the file content" - Section 5 of [69]. This guarantees the integrity of routing objects.

A manifest missing, expired, or not matching objects in the repository can indicate either an error or an attack. In this case, it is up to the RP to decide whether to download routing objects or not (the RP can still use its cache or contact the relevant entities to sort the problem out). For more details about manifests, see [5].

A.5 Trust Anchor Locators

How does one bootstrap a local cache? That is, how can an RP trust that it has acquired the correct CA certificates for the TAs it chose? Distribution of those certificates can be done online or out-of-band, but doing so over insecure communication channels would defeat the purpose of RPKI.

Trust Anchor Locators (TALs) were introduced to simplify CA certificate distribution. A TAL contains an rsync URI [124] and the public key [22] of the CA certificate encoded in Base64 (Section 4 of [61]). A TAL can be distributed by mail, by HTTPs or any other alternate channel to ensure correct distribution. Once an RP has downloaded the root certificate via the rsync URI, it can compare the public key of the certificate with the public key in the TAL, to verify that the downloaded certificate is indeed the one specified in the TAL. For more details about TALs, see [53] (obsoleted by [54]).

Appendix B

Tool behavior

Different security deployments leverage different methods to find potential attackers. The following sections go over different security deployments and specifies those methods.

B.1 Fully deployed BGPsec

In this deployment, the tool only needs to account for On Path attacks, as all other attacks are prevented by the security deployment (Section 3.2). Thus, to find potential attackers, we iterate on the AS path, considering ASes two by two[1].

If forwarding the advertisement on the link between the considered ASes does not respect Gao-Rexford rules, we add the AS to the list of potential On Path with Gao-Rexford break attackers. If this is not the case, we add the AS to the list of potential On Path attackers.

B.2 Fully deployed RPKI

In this deployment, the tool needs to account for On Path attacks, as well as Type-N and Type-U hijacks. Taking into account Assumption 10, Type-U hijacks are reduced to either an On Path attack or an On Path with Gao-Rexford break attack, leaving Type-N hijacks to handle.

The method to find potential attackers is similar to the one in the full BGPsec deployment, but here, we also check if the link between the considered ASes exists or not. If the link does not exist, this means the AS claiming to have received an advertisement on this link lied, so we can add it to the list of potential Type-N hijackers, relative to their position in the AS path.

B.3 Partially deployed RPKI

In this deployment, the tool needs to account for On Path attacks, as well as Type-0, Type-N and Type-U hijacks. Like with the full RPKI deployment, taking into account Assumption 10, Type-U hijacks are reduced to either an On Path attack or an On Path with Gao-Rexford break attack.

Unlike with the full RPKI deployment, the detector AS might not enforce ROV, and the prefix might not be in the RPKI. If either one of those is true, the detector cannot perform ROV, and the method used to determine potential attackers is the same as the method used when no security mechanisms are deployed.

If the prefix is in the RPKI, and the detector performs ROV, we use the same method as in the full RPKI deployment, except we also verify that the Origin AS is authorized to advertise the prefix.

B.4 No security

In this deployment, the tool needs to account for On Path attacks, as well as Type-0, Type-N and Type-U hijacks. Like with the full RPKI deployment, taking into account Assumption 10, Type-U hijacks are reduced to either an On Path attack or an On Path with Gao-Rexford break attack.

[1] In our example, the AS path is {AS40,AS30,AS50}, so the iteration would first consider {AS30,AS50}, then {AS40,AS30}

The method to find potential attackers is similar to the one in the full RPKI deployment, but as the detector cannot perform ROV, the Origin AS cannot be trusted as is thus classified as a potential Type-0 hijacker, provided that the link between the Origin AS and the second AS of the AS path exists.

Acronyms

ACL Access Control List. 8, 11, *Glossary:* ACL

AIA Authority Information Access. 16, *Glossary:* AIA

AS Autonomous System. 2–5, 8–40, 43, 44, *Glossary:* AS

ASN AS Number. 2, 3, 10, 13–15, 17–21, 24, 30, 34, *Glossary:* ASN

BGP Border Gateway Protocol. 2–5, 8–14, 16–18, 20–25, 29, 30, 34–36, 39, 40, *Glossary:* BGP

CA Certification Authority. 15, 16, *Glossary:* CA

CDN Content Delivery Network. 5, *Glossary:* CDN

CharGen Character Generator Protocol. 7, *Glossary:* CharGen

CLDAP Connectionless Lightweight Directory Access Protocol. 6, *Glossary:* CLDAP

CMS Cryptographic Message Syntax. 15, *Glossary:* CMS

CRL Certificate Revocation List. 16, 41, *Glossary:* CRL

CRLDP CRL Distribution Point. 16, 41, *Glossary:* CRLDP

DDoS Distributed Denial of Service. 5–8, 10, 22, 39, *Glossary:* DDoS

DNS Domain Name System. 6, 8, 17, *Glossary:* DNS

DoS Denial of Service. 5, 20, *Glossary:* DoS

DRDoS Distributed Reflective Denial of Service. 7, *Glossary:* DRDoS

eBGP External BGP; Exterior Border Gateway Protocol. 3, 10, 18, 23, 24, 32, *Glossary:* eBGP

EE certificate End-Entity certificate. 15, 41, *Glossary:* EE certificate

iBGP Internal BGP; Interior Border Gateway Protocol. 3, 9–11, 18, *Glossary:* iBGP

ICMP Internet Control Message Protocol. 7, 11, *Glossary:* ICMP

IP Internet Protocol. 3, 6–8, 10, 12, 14–16, 22, 32, *Glossary:* IP

IPsec IP Security. 20, *Glossary:* IPsec

IPv4 Internet Protocol version 4. 4, 22, 32, *Glossary:* IPv4

IPv6 Internet Protocol version 6. 4, 9, 22, 32, *Glossary:* IPv6

IRR Internet Routing Registry. 14, 22, 24, 25, 30, 32, 34, *Glossary:* IRR

ISP Internet Service Provider. 5, 10, 15, *Glossary:* ISP

IXP Internet eXchange Point. 5, 10, 24, 30, 32, *Glossary:* IXP

NLRI Network Layer Reachability Information. 19, *Glossary:* NLRI

NTP Network Time Protocol. 7, *Glossary:* NTP

PKI Public Key Infrastructure. 15, *Glossary:* PKI

QoS Quality of Service. 11, *Glossary:* QoS

RADb Routing Assets Database. 14, *Glossary:* RADb

RIB Routing Information Base. 4, *Glossary:* RIB

RIR Regional Internet Registry. 15, *Glossary:* RIR

ROA Route Origin Authorization. 15–17, 20, 29, 41, *Glossary:* ROA

ROV Route Origin Validation. 27–30, 37, 38, 43, 44, *Glossary:* ROV

RP Relying Party. 15–18, 20, 41, 42, *Glossary:* RP

RPKI Resource Public Key Infrastructure. 14–18, 20–22, 24, 25, 27–30, 32–35, 37, 38, 41–44, *Glossary:* RPKI

RRDP RPKI Repository Delta Protocol. 16, *Glossary:* RRDP

RRL Response Rate Limiting. 8, *Glossary:* RRL

RTBH Remote Triggered Black Hole. 8, 24, 26, *Glossary:* RTBH

SIA Subject Information Access. 16, *Glossary:* SIA

SIP Session Initiation Protocol. 7, *Glossary:* SIP

SKI Subject Key Identifier. 19, 36, *Glossary:* SKI

SNMP Simple Network Management Protocol. 7, *Glossary:* SNMP

SSDP Simple Service Discovery Protocol. 7, *Glossary:* SSDP

TA Trust Anchor. 15, 42, *Glossary:* TA

TAL Trust Anchor Locator. 42, *Glossary:* TAL

TCP Transmission Control Protocol. *Glossary:* TCP

TCP AO TCP Authentication Option. 20, *Glossary:* TCP AO

TLV type-length-value. 34, 35, *Glossary:* TLV

UDP User Datagram Protocol. 6–8, *Glossary:* UDP

URI Uniform Resource Identifier. 16, 41, 42, *Glossary:* URI

uRPF unicast Reverse Path Forwarding. 10, 11, *Glossary:* uRPF

VRP Validated ROA Payload. 16, 17, *Glossary:* VRP

Glossary

ACL List of rules acting as filters to control inbound and outbound traffic. 8

Adj-RIB-In Data table containing routes learned from neighbors. 4

Adj-RIB-Out Data table containing routes selected from Loc-RIB, which the router will announce to its neighbors. 4

AIA URI referencing the location of the parent CA certificate. 16

AS Network or collection of networks under the control of a single entity. 2

AS path Sequence of ASes traversed to reach a destination. 2, 5, 10, 12–14, 18, 21, 22, 24–28, 30–39, 43, 44

ASN Unique AS identifier. 2

BGP The de-facto inter-domain routing protocol in the Internet. 2

BGP speaker Router capable of using BGP to communicate with other BGP speakers. 2–4, 9, 10, 18, 21, 24

BGPsec BGP extension providing security for the AS path attribute. 18–22, 25–30, 32–39, 43

BGPsec speaker Router capable of using BGPsec to communicate with other BGPsec speakers. 18–20, 35, 36

Blackholing (also **RTBH**) DDoS mitigation technique consisting of specifying that a set of routers or a network should discard any traffic destinated towards a specified IP prefix. 8–13, 21–35, 37–39, 49

Border router Router located at the border of an AS, communicating with routers in other ASes. 2, 8–12, 25

Botnet Group of Internet devices used together towards a specific goal. 7

CA Entity holding resources in the RPKI. 15

CA certificate (also **Resource certificate**) Certificate attesting to the ownership of resources to a given CA. 15, 16, 41, 42, 49

CDN Geographically distributed group of servers providing fast delivery of content. 5

CharGen Protocol sending random characters to connecting hosts. Used for testing, debugging and measurement purposes [96]. 7

CLDAP Protocol used to lookup small amounts of information held in a directory [127]. 6

CMS Syntax used to digitally sign, digest, authenticate, or encrypt arbitrary message content [47]. 15

Community Group of destinations which share some common property [17]. 3, 9–11, 18, 21–28, 30–36, 39

Confederation AS divided into multiple internal sub-ASes to reduce iBGP mesh size, but still advertised as a single AS to external peers. 3, 18, 20, 39

CRL List of revoked certificates. 16

CRLDP Distribution point of CRLs. 16

DDoS DoS attack originating from multiple sources. 5

DNS Decentralized naming system for Internet resources. Mostly known for translating domain names to IP addresses [84, 85]. 6

DNS resolver Public DNS server configured to respond to hosts outside of their domain. 6, 7

DoS Attack trying to make a machine, service or network unavailable by flooding the target with requests. 5

DRDoS DDoS attack combining reflection and amplification. 7

eBGP BGP running between two routers in different ASes. 3

EE certificate Certificate used by a CA to sign routing objects. 15

Hijack Illegitimate advertisement of a route. 12–14, 18, 24–31, 37, 43, 48

Hijacking *See* hijack. 12, 13, 25, 39

iBGP BGP running between two routers in the same AS. 3

ICMP Protocol used to send error messages and operational information in IP networks [95]. 7

IP Communication protocol designed for use in packet-switched computer networks [97]. 3

IPsec Suite of protocols providing security to Internet communications at the IP layer [35]. 20

IPv4 Fourth version of the Internet Protocol [97]. 4

IPv6 Most recent version of the Internet Protocol [27]. 4

IRR Distributed routing database capable of origin validation. 14

ISP Entity providing access to the Internet. 5

IXP Infrastructure enabling ASes to exchange Internet traffic in a multilateral interconnection. 5

Leak Accidental hijack of a route. 12

Loc-RIB Data table containing routes selected from Adj-RIB-In by applying import policies. 4

NLRI Field of a BGP UPDATE message containing prefix information. 19

NTP Protocol used for clock synchronization between IP devices [82]. 7

Origin AS Rightmost AS in the AS path attribute; AS originating the prefix. 2, 13–15, 17, 21, 28–30, 38, 43, 44

Origin ASN ASN of the Origin AS. 16, 17

Origin validation (also **ROV**) Process of validating the origin of a route. 14, 15, 20, 21, 24, 49

Path validation Process of validating the AS path of a route. 14, 18, 20

PKI Hierarchic infrastructure used to manage public keys. 15

Prefix IP prefix; Block of IP addresses. 2–4, 8–26, 28–30, 32–34, 36, 38, 43

Prefix holder Legitimate holder of IP address space. 14, 15

Publication point Repository containing all the routing objects associated with a CA certificate. 15, 16, 41

QoS Set of techniques used to manage resources, in order to meet certain requirements. 11

RADb Distributed routing database capable of origin validation. 14

Resource certificate *See* CA certificate. 14, 15

RIB Data table listing routes to available destinations. 4

RIR Entity managing Internet number resources. 15

ROA Routing object providing an authorization by a prefix holder that a given AS is permitted to originate routes to a set of prefixes. 15

Route Unit of information that associates a set of destinations described by an IP address prefix with a set of attributes of a path to those destinations [100]. 2–5, 9–20, 22, 24, 26, 27, 29

Route server Third-party brokering system. Used at IXPs to manage BGP sessions. 5, 18, 20, 24, 30, 32

ROV *See* origin validation. 27

RP Entity participating in the PKI. 15

RPKI Distributed hierarchic public key infrastructure capable of origin validation. 14

RRDP Protocol used to collect RPKI information from publication points [11]. 16

RRL Mitigation technique consisting of reducing the rate at which servers respond to a high amount of forged queries. 8

RTBH *See* blackholing. 8

SIA URI referencing the repository associated with a CA certificate. 16

SIP Protocol used to manage real-time communications (voice, video, ...) [105]. 7

SKI Identifier of the certificate used to generate a given Signature. 19

SNMP Protocol used for monitoring devices on IP networks [16]. 7

SSDP Protocol enabling the advertisement and discovery of network services and presence information [2]. 7

Sub-prefix More specific prefix in relation to a given prefix. 13, 18, 25, 26, 28, 29

TA Authoritative entity represented by a public key and associated data [48]. 15

TAL Format used to distribute TA material [53]. 42

TCP AO TCP option to authenticate TCP segments [117]. 20

TLV Encoding format indicating the type, the length and the value of a given element. 34

Traffic scrubbing The action of processing traffic, detecting malicious traffic and dropping it, then forwarding the "cleaned" traffic to the destination. 8, 11

UDP Communication protocol built on top of the Internet Protocol [94]. 6

URI Compact sequence of characters that identifies an abstract or physical resource [9]. 16

uRPF Technique used to discard malicious traffic, dropping packets if their source address is unreachable. 10

VRP Payload containing prefix origin data (IP address, prefix length, maximum length, Origin ASN) [87]. 16

Bibliography

[1] Akamai. Memcached-fueled 1.3 Tbps attacks. https://blogs.akamai.com/2018/03/memcached-fueled-13-tbps-attacks.html, March 2018. [Online; accessed 7-March-2018].

[2] Shivaun Albright, Paul J. Leach, Ye Gu, Yaron Y. Goland, and Ting Cai. Simple Service Discovery Protocol/1.0. Internet-Draft draft-cai-ssdp-v1-03, Internet Engineering Task Force, November 1999. Work in Progress.

[3] Andreas Reuter and Randy Bush and Ethan Katz-Bassett and Italo Cunha and Thomas C. Schmidt and Matthias Wählisch. Measuring Adoption of RPKI Route Origin Validation and Filtering. https://ripe76.ripe.net/presentations/63-rov_filtering_update.pdf, May 2018. [Online; accessed 4-August-2018].

[4] J. Arkko, M. Cotton, and L. Vegoda. Ipv4 address blocks reserved for documentation. RFC 5737, RFC Editor, January 2010.

[5] R. Austein, G. Huston, S. Kent, and M. Lepinski. Manifests for the resource public key infrastructure (rpki). RFC 6486, RFC Editor, February 2012.

[6] F. Baker and P. Savola. Ingress filtering for multihomed networks. BCP 84, RFC Editor, March 2004. http://www.rfc-editor.org/rfc/rfc3704.txt.

[7] A. Barbir, S. Murphy, and Y. Yang. Generic threats to routing protocols. RFC 4593, RFC Editor, October 2006.

[8] Dima Bekerman, Avishay Zawoznik, and Igal Zeifman. New Variant of Mirai Embeds Itself in TalkTalk Home Routers. https://www.incapsula.com/blog/new-variant-mirai-embeds-talktalk-home-routers.html, December 2016. [Online; accessed 10-April-2018].

[9] Tim Berners-Lee, Roy T. Fielding, and Larry Masinter. Uniform resource identifier (uri): Generic syntax. STD 66, RFC Editor, January 2005. http://www.rfc-editor.org/rfc/rfc3986.txt.

[10] Thomas Brewster. Cyber Attacks Strike Zimbabweans Around Controversial Election. http://www.silicon.co.uk/workspace/zimbabwe-election-cyber-attacks-123938, August 2013. [Online; accessed 15-March-2018].

[11] T. Bruijnzeels, O. Muravskiy, B. Weber, and R. Austein. The rpki repository delta protocol (rrdp). RFC 8182, RFC Editor, July 2017.

[12] R. Bush. The resource public key infrastructure (rpki) ghostbusters record. RFC 6493, RFC Editor, February 2012.

[13] R. Bush. Bgpsec operational considerations. BCP 211, RFC Editor, September 2017.

[14] R. Bush and R. Austein. The resource public key infrastructure (rpki) to router protocol. RFC 6810, RFC Editor, January 2013.

[15] R. Bush and R. Austein. The resource public key infrastructure (rpki) to router protocol, version 1. RFC 8210, RFC Editor, September 2017.

[16] Jeffrey D. Case, Mark Fedor, Martin Lee Schoffstall, and James R. Davin. Simple network management protocol (snmp). STD 15, RFC Editor, May 1990. http://www.rfc-editor.org/rfc/rfc1157.txt.

[17] R. Chandra, P. Traina, and T. Li. BGP Communities Attribute. RFC 1997, RFC Editor, August 1996.

[18] E. Chen, J. Scudder, P. Mohapatra, and K. Patel. Revised error handling for bgp update messages. RFC 7606, RFC Editor, August 2015.

[19] Cisco. Remotely Triggered Black Hole Filtering - Destination Based and Source Based. https://www.cisco.com/c/dam/en/us/products/collateral/security/ios-network-foundation-protection-nfp/prod_white_paper0900aecd80313fac.pdf, 2005. [Online; accessed 17-May-2018].

[20] CIX. CIX Blackholing Service. https://www.cix.hr/en/about-cix/infrastructure/route-server, July 2018. [Online; accessed 16-July-2018].

[21] CNRS. Centre National de la Recherche Scientifique. http://www.cnrs.fr/index.html, August 2018. [Online; accessed 8-August-2018].

[22] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile. RFC 5280, RFC Editor, May 2008. http://www.rfc-editor.org/rfc/rfc5280.txt.

[23] Danny Cooper, Ethan Heilman, Kyle Brogle, Leonid Reyzin, and Sharon Goldberg. On the risk of misbehaving rpki authorities. In *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, page 16. ACM, 2013.

[24] Jim Cowie. China's 18-Minute Mystery. https://dyn.com/blog/chinas-18-minute-mystery/, November 2010. [Online; accessed 15-May-2018].

[25] Jim Cowie. The New Threat: Targeted Internet Traffic Misdirection. https://dyn.com/blog/mitm-internet-hijacking/, November 2013. [Online; accessed 16-May-2018].

[26] DE-CIX. DE-CIX Blackholing Service. https://www.de-cix.net/_-Resources/Persistent/4277e7d4867a78ae923c0f5b3b66d7ff6aeb61f8/DE-CIX-Blackholing-Service.pdf, July 2018. [Online; accessed 16-July-2018; Slide 3].

[27] S. Deering and R. Hinden. Internet protocol, version 6 (ipv6) specification. STD 86, RFC Editor, July 2017.

[28] Christoph Dietzel, Anja Feldmann, and Thomas King. Blackholing at ixps: On the effectiveness of ddos mitigation in the wild. In Thomas Karagiannis and Xenofontas Dimitropoulos, editors, *Passive and Active Measurement*, pages 319–332, Cham, 2016. Springer International Publishing.

[29] Michael E. Donner. Prolexic Stops Largest Ever DNS Reflection DDoS Attack - 167 Gbps Attack Targets Real-Time Financial Exchange Platform. http://www.prweb.com/releases/prolexic/dos-ddos-mitigation/prweb10782467.htm, May 2013. [Online; accessed 15-March-2018].

[30] J. Durand, I. Pepelnjak, and G. Doering. Bgp operations and security. BCP 194, RFC Editor, February 2015.

[31] ENGEES. Ecole Nationale du Génie de l'Eau et de l'Environnement de Strasbourg. https://engees.unistra.fr/en/, August 2018. [Online; accessed 8-August-2018].

[32] Equinix. Equinix Blackholing Service. http://www.sanog.org/resources/sanog28/SANOG28-Conference_RTBH-Safiudeen.pdf, August 2016. [Online; accessed 16-July-2018; Slide 15].

[33] Rob Evans and Philip Smith. RIPE Routing Working Group Recommendations on IPv6 Route Aggregation. https://www.ripe.net/publications/docs/ripe-532, 2011. [Online; accessed 22-May-2018].

[34] France-IX. France-IX Blackholing Service. https://www.franceix.net/fr/technical/blackholing/, July 2018. [Online; accessed 16-July-2018].

[35] S. Frankel and S. Krishnan. Ip security (ipsec) and internet key exchange (ike) document roadmap. RFC 6071, RFC Editor, February 2011. http://www.rfc-editor.org/rfc/rfc6071.txt.

[36] Future Internet of the Things. IoT-LAB. https://www.iot-lab.info/, August 2018. [Online; accessed 8-August-2018].

[37] Lixin Gao, Timothy G Griffin, and Jennifer Rexford. Inherently safe backup routing with bgp. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 547–556. IEEE, 2001.

[38] Lixin Gao and Jennifer Rexford. Stable internet routing without global coordination. *IEEE/ACM Transactions on Networking (TON)*, 9(6):681–692, 2001.

[39] W. George and S. Murphy. Bgpsec considerations for autonomous system (as) migration. RFC 8206, RFC Editor, September 2017.

[40] Yossi Gilad, Avichai Cohen, Amir Herzberg, Michael Schapira, and Haya Shulman. Are we there yet? on rpki's deployment and security. *IACR Cryptology ePrint Archive*, 2016:1010, 2016.

[41] Phillipa Gill, Michael Schapira, and Sharon Goldberg. A survey of interdomain routing policies. *ACM SIGCOMM Computer Communication Review*, 44(1):28–34, 2013.

[42] Dan Goodin. Russian-controlled telecom hijacks financial services' Internet traffic. https://arstechnica.com/information-technology/2017/04/russian-controlled-telecom-hijacks-financial-services-internet-traffic/, April 2017. [Online; accessed 15-May-2018].

[43] Andy Greenberg. Hacker Redirects Traffic From 19 Internet Providers to Steal Bitcoins. https://www.wired.com/2014/08/isp-bitcoin-theft/, August 2014. [Online; accessed 15-May-2018].

[44] M. Handley and E. Rescorla. Internet Denial-of-Service Considerations. RFC 4732, RFC Editor, December 2006.

[45] J. Heitz, J. Snijders, K. Patel, I. Bagdonas, and N. Hilliard. Bgp large communities attribute. RFC 8092, RFC Editor, February 2017.

[46] N. Hilliard and D. Freedman. A discard prefix for ipv6. RFC 6666, RFC Editor, August 2012.

[47] R. Housley. Cryptographic message syntax (cms). STD 70, RFC Editor, September 2009. http://www.rfc-editor.org/rfc/rfc5652.txt.

[48] R. Housley, S. Ashmore, and C. Wallace. Trust anchor format. RFC 5914, RFC Editor, June 2010.

[49] Hurricane Electric. Hurricane Electric Blackholing Service. https://www.he.net/adm/blackhole.html, July 2018. [Online; accessed 17-July-2018].

[50] G. Huston, R. Loomans, and G. Michaelson. A profile for resource certificate repository structure. RFC 6481, RFC Editor, February 2012.

[51] G. Huston and G. Michaelson. Validation of route origination using the resource certificate public key infrastructure (pki) and route origin authorizations (roas). RFC 6483, RFC Editor, February 2012.

[52] G. Huston, G. Michaelson, and R. Loomans. A profile for x.509 pkix resource certificates. RFC 6487, RFC Editor, February 2012.

[53] G. Huston, S. Weiler, G. Michaelson, and S. Kent. Resource public key infrastructure (rpki) trust anchor locator. RFC 6490, RFC Editor, February 2012.

[54] G. Huston, S. Weiler, G. Michaelson, and S. Kent. Resource public key infrastructure (rpki) trust anchor locator. RFC 7730, RFC Editor, January 2016.

[55] Geoff Huston. AS Number Report. http://www.potaroo.net/tools/asn32/, March 2018. [Online; accessed 9-July-2018].

[56] IANA. Special-use ipv4 addresses. RFC 3330, RFC Editor, September 2002.

[57] ICube. Detailed organizational chart of the ICube laboratory. https://icube.unistra.fr/uploads/media/ORGA-Global-082018-EN.pdf, August 2018. [Online; accessed 2-August-2018].

[58] ICube. ICube Laboratory. https://icube.unistra.fr/en/, August 2018. [Online; accessed 8-August-2018].

[59] INSA. Institut National des Sciences Appliquées. http://www.insa-strasbourg.fr/en/, August 2018. [Online; accessed 8-August-2018].

[60] E. Jasinska, N. Hilliard, R. Raszuk, and N. Bakker. Internet exchange bgp route server. RFC 7947, RFC Editor, September 2016.

[61] S. Josefsson. The base16, base32, and base64 data encodings. RFC 4648, RFC Editor, October 2006. http://www.rfc-editor.org/rfc/rfc4648.txt.

[62] Nishitha Kandagatla. Disgruntled ex-employees, DDoS attacks and the revenge of the nerds. https://www.wittysparks.com/disgruntled-ex-employees-ddos-attacks-and-the-revenge-of-the-nerds/, November 2017. [Online; accessed 9-April-2018].

[63] S. Kent and A. Chi. Threat model for bgp path security. RFC 7132, RFC Editor, February 2014.

[64] T. King, C. Dietzel, J. Snijders, G. Doering, and G. Hankins. Blackhole community. RFC 7999, RFC Editor, October 2016.

[65] Thomas King. Blackhole bgp community. https://github.com/tking/BLACKHOLE-BGP-Community, June 2017.

[66] W. Kumari, R. Bush, H. Schiller, and K. Patel. Codification of as 0 processing. RFC 7607, RFC Editor, August 2015.

[67] W. Kumari and D. McPherson. Remote triggered black hole filtering with unicast reverse path forwarding (urpf). RFC 5635, RFC Editor, August 2009.

[68] M. Lepinski, A. Chi, and S. Kent. Signed object template for the resource public key infrastructure (rpki). RFC 6488, RFC Editor, February 2012.

[69] M. Lepinski and S. Kent. An infrastructure to support secure internet routing. RFC 6480, RFC Editor, February 2012. http://www.rfc-editor.org/rfc/rfc6480.txt.

[70] M. Lepinski, S. Kent, and D. Kong. A profile for route origin authorizations (roas). RFC 6482, RFC Editor, February 2012.

[71] M. Lepinski and K. Sriram. Bgpsec protocol specification. RFC 8205, RFC Editor, September 2017.

[72] John Leyden. US credit card firm fights DDoS attack. http://www.theregister.co.uk/2004/09/23/authorize_ddos_attack/, September 2004. [Online; accessed 15-March-2018].

[73] Robert Lychev, Sharon Goldberg, and Michael Schapira. Bgp security in partial deployment: Is the juice worth the squeeze? *SIGCOMM Comput. Commun. Rev.*, 43(4):171–182, August 2013.

[74] C. Lynn, S. Kent, and K. Seo. X.509 extensions for ip addresses and as identifiers. RFC 3779, RFC Editor, June 2004.

[75] Doug Madory. BackConnect's Suspicious BGP Hijacks. https://dyn.com/blog/backconnects-suspicious-bgp-hijacks/, September 2016. [Online; accessed 16-May-2018].

[76] Doug Madory. Iran Leaks Censorship via BGP Hijacks. https://dyn.com/blog/iran-leaks-censorship-via-bgp-hijacks/, January 2017. [Online; accessed 15-May-2018].

[77] Doug Madory. BGP Hijack of Amazon DNS to Steal Crypto Currency. https://dyn.com/blog/bgp-hijack-of-amazon-dns-to-steal-crypto-currency/, April 2018. [Online; accessed 15-May-2018].

[78] Pratyusa Manadhata and Vyas Sekar. Understanding bgp anomalies: Detection, analysis, and prevention. *15-744 Class Project, CMU*, 2014.

[79] Jared Mauch. Open Resolver Project. http://openresolverproject.org/, April 2018. [Online; accessed 9-April-2018].

[80] Mehmet Adalier and Kotikalapudi Sriram and Oliver Borchert and Kyehwan Lee and Doug Montgomery. High Performance BGP Security: Algorithms and Architectures. https://www.nanog.org/sites/default/files/1_Sriram_High_Performance_Bgp_-v1.pdf, February 2017. [Online; accessed 30-July-2018].

[81] Merit RADb. Merit RADb. http://www.radb.net/, May 2018. [Online; accessed 11-May-2018].

[82] D. Mills, J. Martin, J. Burbank, and W. Kasch. Network time protocol version 4: Protocol and algorithms specification. RFC 5905, RFC Editor, June 2010. http://www.rfc-editor.org/rfc/rfc5905.txt.

[83] Asya Mitseva, Andriy Panchenko, and Thomas Engel. The state of affairs in bgp security: A survey of attacks and defenses. *Computer Communications*, 2018.

[84] P. Mockapetris. Domain names - concepts and facilities. STD 13, RFC Editor, November 1987. http://www.rfc-editor.org/rfc/rfc1034.txt.

[85] P. Mockapetris. Domain names - implementation and specification. STD 13, RFC Editor, November 1987. http://www.rfc-editor.org/rfc/rfc1035.txt.

[86] P. Mohapatra, K. Patel, J. Scudder, D. Ward, and R. Bush. Bgp prefix origin validation state extended community. RFC 8097, RFC Editor, March 2017.

[87] P. Mohapatra, J. Scudder, D. Ward, R. Bush, and R. Austein. Bgp prefix origin validation. RFC 6811, RFC Editor, January 2013. http://www.rfc-editor.org/rfc/rfc6811.txt.

[88] Carlos Morales. NETSCOUT Arbor Confirms 1.7 Tbps DDoS Attack; The Terabit Attack Era Is Upon Us. https://www.arbornetworks.com/blog/asert/netscout-arbor-confirms-1-7-tbps-ddos-attack-terabit-attack-era-upon-us/, March 2018. [Online; accessed 7-March-2018].

[89] MSK-IX. MSK-IX Blackholing Service. https://kb.msk-ix.ru/en/ix/services/route-server/, July 2018. [Online; accessed 16-July-2018].

[90] S. Murphy. Bgp security vulnerabilities analysis. RFC 4272, RFC Editor, January 2006.

[91] Netlab. Insight into Global DDoS Threat Landscape. https://ddosmon.net/insight/, April 2018. [Online; accessed 9-April-2018].

[92] Lily Hay Newman. The Botnet That Broke the Internet Isn't Going Away. https://www.wired.com/2016/12/botnet-broke-internet-isnt-going-away/, September 2016. [Online; accessed 7-March-2018].

[93] S. Perreault. vcard format specification. RFC 6350, RFC Editor, August 2011.

[94] J. Postel. User datagram protocol. STD 6, RFC Editor, August 1980. http://www.rfc-editor.org/rfc/rfc768.txt.

[95] J. Postel. Internet control message protocol. STD 5, RFC Editor, September 1981. http://www.rfc-editor.org/rfc/rfc792.txt.

[96] J. Postel. Character generator protocol. STD 22, RFC Editor, May 1983.

[97] Jon Postel. Internet protocol. STD 5, RFC Editor, September 1981. http://www.rfc-editor.org/rfc/rfc791.txt.

[98] Aiko Pras, Anna Sperotto, Giovane C. M. Moura, Idilio Drago, Rafael Barbosa, Ramin Sadre, Ricardo Schmidt, and Rick Hofstede. Attacks by "anonymous " wikileaks proponents not anonymous, 2010.

[99] Matthew Prince. The DDoS That Almost Broke the Internet. https://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet/, March 2013. [Online; accessed 15-March-2018].

[100] Y. Rekhter, T. Li, and S. Hares. A border gateway protocol 4 (bgp-4). RFC 4271, RFC Editor, January 2006. http://www.rfc-editor.org/rfc/rfc4271.txt.

[101] Yakov Rekhter, Robert G. Moskowitz, Daniel Karrenberg, Geert Jan de Groot, and Eliot Lear. Address allocation for private internets. BCP 5, RFC Editor, February 1996. http://www.rfc-editor.org/rfc/rfc1918.txt.

[102] Andreas Reuter, Randy Bush, Italo Cunha, Ethan Katz-Bassett, Thomas C Schmidt, and Matthias Wählisch. Towards a rigorous methodology for measuring adoption of rpki route validation and filtering. *ACM SIGCOMM Computer Communication Review*, 48(1):19–27, 2018.

[103] M. Reynolds, S. Turner, and S. Kent. A profile for bgpsec router certificates, certificate revocation lists, and certification requests. RFC 8209, RFC Editor, September 2017.

[104] RIPE NCC. YouTube Hijacking: A RIPE NCC RIS case study. https://www.ripe.net/publications/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study, March 2008. [Online; accessed 15-May-2018].

[105] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. Sip: Session initiation protocol. RFC 3261, RFC Editor, June 2002. http://www.rfc-editor.org/rfc/rfc3261.txt.

[106] Christian Rossow. Amplification hell: Revisiting network protocols for ddos abuse. In *NDSS*, 2014.

[107] rsync. rsync web pages. https://rsync.samba.org/, 2018. [Online; accessed 6-June-2018].

[108] Fabrice J Ryba, Matthew Orlinski, Matthias Wählisch, Christian Rossow, and Thomas C Schmidt. Amplification and drdos attack defense–a survey and new perspectives. *arXiv preprint arXiv:1505.07892*, 2015.

[109] S. Sangli, D. Tappan, and Y. Rekhter. Bgp extended communities attribute. RFC 4360, RFC Editor, February 2006.

[110] Johann Schlamp, Ralph Holz, Quentin Jacquemart, Georg Carle, and Ernst W Biersack. Heap: reliable assessment of bgp hijacking attacks. *IEEE Journal on Selected Areas in Communications*, 34(6):1849–1861, 2016.

[111] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti. Artemis: Neutralizing bgp hijacking within a minute. *arXiv preprint arXiv:1801.01085*, 2018.

[112] Philip Smith. BGP Routing Table Analysis - DIX-IE Data. http://thyme.rand.apnic.net/current/data-summary, March 2018. [Online; accessed 9-July-2018].

[113] Philip Smith, Rob Evans, and Mike Hughes. RIPE Routing Working Group Recommendations on Route Aggregation. https://www.ripe.net/publications/docs/ripe-399, 2006. [Online; accessed 22-May-2018].

[114] Kerry Tomlinson. Cyber battle rages on Internet after arrest of cyber crime suspects. http://www.archersecuritygroup.com/cyber-battle-rages-internet-arrest-cyber-crime-suspects/, September 2016. [Online; accessed 16-May-2018].

[115] Andree Toonk. How accurate are the Internet Route Registries (IRR). https://bgpmon.net/how-accurate-are-the-internet-route-registries-irr/, March 2009. [Online; accessed 20-April-2018].

[116] Andree Toonk. Chinese ISP hijacks the Internet. https://bgpmon.net/chinese-isp-hijacked-10-of-the-internet/, April 2010. [Online; accessed 15-May-2018].

[117] J. Touch, A. Mankin, and R. Bonica. The tcp authentication option. RFC 5925, RFC Editor, June 2010.

[118] P. Traina, D. McPherson, and J. Scudder. Autonomous system confederations for bgp. RFC 5065, RFC Editor, August 2007.

[119] D. Turk. Configuring bgp to block denial-of-service attacks. RFC 3882, RFC Editor, September 2004.

[120] S. Turner and O. Borchert. Bgpsec algorithms, key formats, and signature formats. RFC 8208, RFC Editor, September 2017.

[121] Pierre-Antoine Vervier, Olivier Thonnard, and Marc Dacier. Mind your blocks: On the stealthiness of malicious bgp hijacks. In *NDSS*, 2015.

[122] Q. Vohra and E. Chen. Bgp support for four-octet autonomous system (as) number space. RFC 6793, RFC Editor, December 2012. http://www.rfc-editor.org/rfc/rfc6793.txt.

[123] S. Weiler, A. Sonalker, and R. Austein. A publication protocol for the resource public key infrastructure (rpki). RFC 8181, RFC Editor, July 2017.

[124] S. Weiler, D. Ward, and R. Housley. The rsync uri scheme. RFC 5781, RFC Editor, February 2010.

[125] Dan Wendlandt, Ioannis C. Avramopoulos, David G. Andersen, and Jennifer Rexford. Don't secure routing protocols, secure data delivery. In *HotNets*, 2006.

[126] Avishay Zawoznik and Dima Bekerman. 650Gbps DDoS Attack from the Leet Botnet. https://www.incapsula.com/blog/650gbps-ddos-attack-leet-botnet.html, December 2016. [Online; accessed 10-April-2018].

[127] K. Zeilenga. Connection-less lightweight directory access protocol (cldap) to historic status. RFC 3352, RFC Editor, March 2003.

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