

# BGP Path Hijacking

NHU DUONG, Oregon State University

BROGAN MINER, Oregon State University

MICHAEL WAGONER, Oregon State University

The BGP Protocol is the backbone of the internet. Over the past 30 years it has grown to be a key protocol for exchanging routing and reachability information across autonomous systems on the internet. However, BGP was not designed with security in mind. This report delves into the security of BGP and shows a path hijacking attacking using Mininet. Lastly, BGPsec and the Mutually Agreed Norms of Routing Security are discussed as ways to improve BGP's Security moving forward.

CCS Concepts: • **Networks** → Network Security.

Additional Key Words and Phrases: BGP, BGP Security, BGP Path Hijacking, MANRS, BGPsec

## 1 INTRODUCTION

The internet is a well-known service being used for multiple facets of life. Whether the use of the internet is personal or commercial, the use of such a service is now a necessity in the modern world. This has opened up a large attack surface for people to expose, manipulate, and benefit off of the lax security and need for the internet. In this document, we will explore one method of attack. The attack method explored in this document is the BGP path hijacking attack. This attack can lead to other exploits and snooping of web traffic. Recently this type of attack has been surfacing and been prevalent all over the internet. [Doffman 2020]

## 2 BGP OVERVIEW

BGP, Border Gateway Protocol, is designed to exchange routing and reachability information among autonomous systems (AS) on the Internet. BGP is a path-vector routing protocol which means routing is based on paths and network policies. BGP is the most important tool which interconnects everything on the Internet together.

One of the biggest risks to BGP is Path Hijacking. BGP Path Hijacking can effectively reroute internet traffic. The attackers falsely announce the ownership of a group of IP addresses (IP prefixes) to overwrite the existing valid connection. Attackers can intercept or modify the traffic. At the internet level BGP Path hijacking, attackers configure an edge router to announce a falsely prefixed network. The compromised edge router claims to offer the shorter path to the destination and the traffic may be re-routed to the IP of the attackers.

## 3 APPROACH

To facilitate the attack and ensure no systems are harmed in the demonstration we will be using the tool Mininet. Mininet is a network emulator that works by process-based virtualization, where each router, switch, or network device is located in its own process. This allows us to emulate routers running BGP and attack them in a safe way.

### 3.1 Network Layout

The setup for this network will use four different autonomous systems or AS for short. AS One, AS Two, and AS Three will represent the existing network structure. AS Four will then be brought in as an adversary to show how the path can be hijacked. Below is a diagram detailing the network layout and advertised IP address allocation chunks.

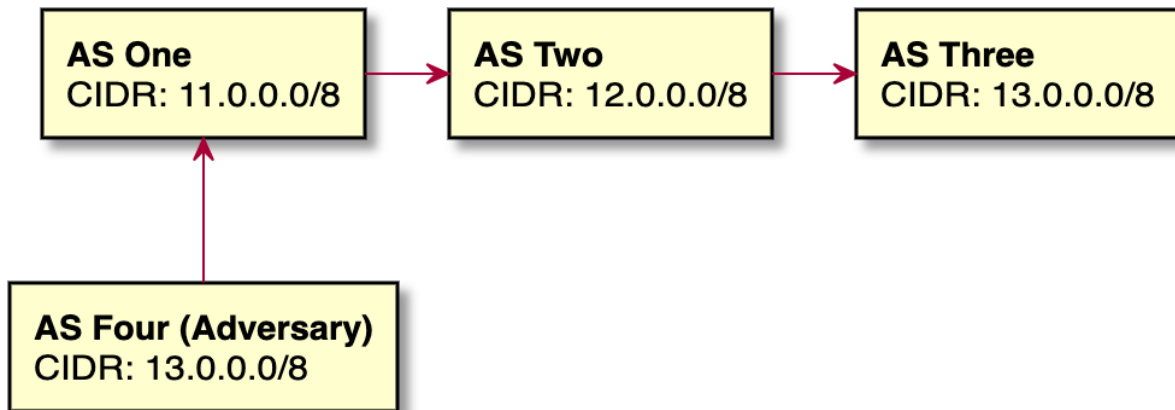


Fig. 1. Diagram detailing the network layout and advertised IP address allocation chunks

### 3.2 Setting Up Mininet Simulation

The first step is to download the Mininet VM, available at [Lantz and O'Connor [n.d.]]. This VM will also require you to have a machine capable of virtualization and an appropriate client to run the VM. In this document we use VirtualBox but other clients may work. After downloading the ZIP file above, simply open the .ovf with VirtualBox. This will import the VM settings and disk image automatically.

Next start the VM. Once the VM has been booted a login prompt will be displayed. The login details for this VM are:

```
User: mininet
Password: mininet
```

Once you have logged in, open a terminal window. The VM includes the necessary Mininet configuration and utilities for this attack. CD into the BGP directory and start the BGP Mininet service with:

```
sudo python bgp.py
```

The `bgp.py` file contains the default configuration of four different switches and 3 hosts per switch. The fourth switch is the rogue switch that we will initiate the attack from.

```
mininet@mininet-vm:~$ cd bgp
mininet@mininet-vm:~/bgp$ sudo python bgp.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1-1 h1-2 h1-3 h2-1 h2-2 h2-3 h3-1 h3-2 h3-3 h4-1 h4-2 h4-3
*** Adding switches:
R1 R2 R3 R4
*** Adding links:
(R1, R2) (R1, R4) (R1, h1-1) (R1, h1-2) (R1, h1-3) (R2, R3) (R2, h2-1) (R2, h2-2)
(R2, h2-3) (R3, h3-1) (R3, h3-2) (R3, h3-3) (R4, h4-1) (R4, h4-2) (R4, h4-3)
*** Configuring hosts
h1-1 h1-2 h1-3 h2-1 h2-2 h2-3 h3-1 h3-2 h3-3 h4-1 h4-2 h4-3
*** Starting controller
*** Starting 4 switches
R1 R2 R3 R4
Waiting 3 seconds for sysctl changes to take effect...
Starting zebra and bgpd on R1
Starting zebra and bgpd on R2
Starting zebra and bgpd on R3
Starting web servers
*** Starting CLI:
mininet> █
```

Fig. 2. Start BGP Mininet service

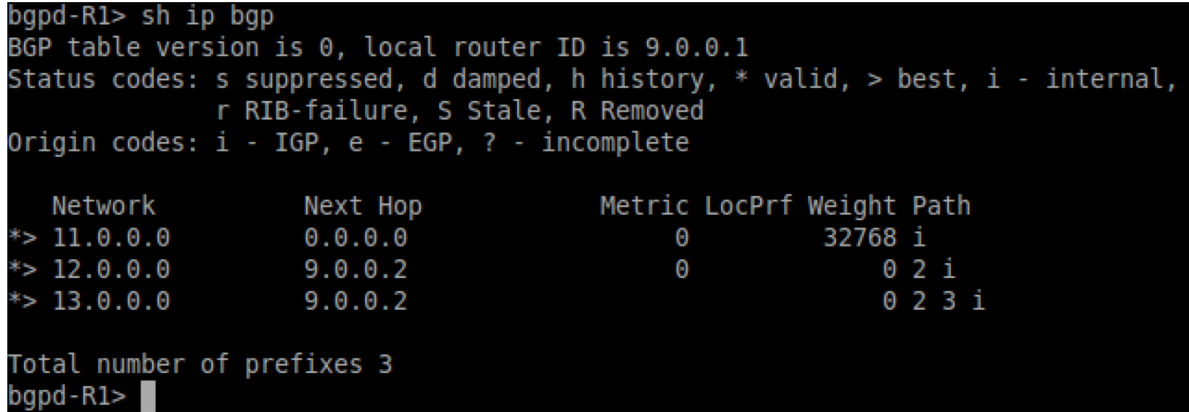
After this we can check the routing tables on the first router to see the path as defined on the network layout. This is done by executing the following commands in a new terminal.

```
sudo python run.py --node -R1 --cmd "telnet localhost bgpd"
```

The command above launches the python utility script that will start up the console of the emulated mininet network device. The node flag followed by -R1 is telling the script that we want to access the first router. The cmd flag is used to run the following command in quotes. This initiates the console to the Quagga routing suite which we are using to run the BGP protocol. A script has been provided connect.sh that facilitates this connection. When asked for a password simply enter en

After the console has been launched one can run the command below to view the routing table Fig. 3

ip bgp



```

bgpd-R1> sh ip bgp
BGP table version is 0, local router ID is 9.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop        Metric LocPrf Weight Path
*> 11.0.0.0        0.0.0.0           0         32768  i
*> 12.0.0.0        9.0.0.2           0           0 2  i
*> 13.0.0.0        9.0.0.2           0           0 2 3  i

Total number of prefixes 3
bgpd-R1>

```

Fig. 3. Routing table

The routing to the third router is down through the second router. This is seen on the third row in the path column containing the path 2 3. We have successfully set up the BGP network.

Next we will try accessing a website contained in the third network (this is setup when running the bgp.py python script) from a host in the first network. This is done with the following command:

```
sudo python run.py --node -h1-1 --cmd "curl -s 13.0.1.1"
```

A convenience script has been created in the repository that continuously makes this call every second. This script is named website.sh. Fig. 4 shows the screenshot when calling the script.

### 3.3 Initiating the Attack

Now that the network is set up we will turn on the fourth router and advertise the same address space as the third router. This will re-route traffic from going through the second router then to the third and instead just route traffic straight to the fourth router. To do this we will need to start the quagga and bgpd services. Configuration files and a convenience script to do this have been included in the repository. The first command is:

```
sudo python run.py --node R4 --cmd
"/usr/lib/quagga/zebra -f conf/zebra-R4.conf -d -i /tmp/zebra-R4.pid > logs/R4-zebra-stdout"
```

Following by the second command:

```
sudo python run.py --node R4 --cmd
"/usr/lib/quagga/bgpd -f conf/bgpd-R4.conf -d -i /tmp/bgpd-R4.pid > logs/R4-bgpd-stdout"
```

The convenience script for these commands is located in start\_rogue.sh. The output of the script shows on Fig. 5.

```

    httpd.serve_forever()
File "/usr/lib/python2.7/SocketServer.py", line 236, in serve_forever
    poll_interval)
File "/usr/lib/python2.7/SocketServer.py", line 155, in _eintr_retry
    return func(*args)
KeyboardInterrupt
mininet@mininet-vm:~/bgp$ vim webs
webserver.py website.sh
mininet@mininet-vm:~/bgp$ vim website.sh
mininet@mininet-vm:~/bgp$ bash website.sh
Fri Jun 5 15:00:29 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:00:30 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:00:31 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:00:32 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:00:33 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:00:34 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:00:35 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:00:36 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:00:37 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:00:38 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:00:39 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:00:40 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:00:41 PDT 2020 -- <h1>Default web server</h1>

```

Fig. 4. Access Default Web Server

```

mininet@mininet-vm:~/bgp$ bash start_rogue.sh
Killing any existing rogue AS
Starting rogue AS
mininet@mininet-vm:~/bgp$ █

```

Fig. 5. Reroute traffic script output

After running reroute script, one can go back to the web server script and check to see if the content is changed on the page. Fig. 6 shows the output of the check web server script during the switch from one AS to the other.

```

Fri Jun 5 15:19:59 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:20:00 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:20:01 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:20:03 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:20:04 PDT 2020 -- <h1>Default web server</h1>
Fri Jun 5 15:20:05 PDT 2020 -- <h1>Default web server</h1>
Sat Jun 6 16:19:44 PDT 2020 --
Sat Jun 6 16:19:45 PDT 2020 -- <h1>*** Attacker web server ***</h1>
Sat Jun 6 16:19:46 PDT 2020 -- <h1>*** Attacker web server ***</h1>
Sat Jun 6 16:19:47 PDT 2020 -- <h1>*** Attacker web server ***</h1>
Sat Jun 6 16:19:48 PDT 2020 -- <h1>*** Attacker web server ***</h1>

```

Fig. 6. Traffic gets rerouted to Attacker web server

## 4 PREVENTING BGP HIJACKING

BGP was not designed with security in mind, there isn't much a user or network can do to prevent BGP hijacks. Networks should only declare their IP prefixes to certain networks and only accept IP prefix declarations if necessary. IP Prefix filtering can help prevent accidental route hijacking by keeping the AS from accepting bogus IP prefix declarations. One way to detect BGP hijacking is to check for increased latency, degraded network performance, and misdirected Internet traffic. [Cloudflare 2020]

### 4.1 Mutually Agreed Norms for Routing Security

The Mutually Agreed Norms for Routing Security is an initiative to improve the resilience and security of the Internet's global routing system. Targeted towards network operators and IXPs (Internet exchange point) MANRS sets routing security expectations. There are 375 ISPs currently participating in MANRS. To be accepted as a MANRS participant there are three actions which the ISP must accomplish and two recommended actions. [A. et al. 2019]

#### 4.1.1 MANRS Required Actions.

- (1) **Prevent propagation of incorrect routing information.** [A. et al. 2019] To accomplish this the network operator must have a system where the AS numbers and IP prefixes are only announced to adjacent networks. Additionally, the network operator must verify their customers' announcements are correct.
- (2) **Facilitate global operational communication and coordination.** [A. et al. 2019] Up-to-date contact information of the network operator must be maintained in the appropriate RIR/NIR database and/or PeeringDB. This information should be publicly available, but at a minimum available to other network operators registered with PeeringDB.
- (3) **Facilitate routing information on a global scale -Internet Routing Registry.** [A. et al. 2019] Intended routing announcements must be publicly documented in the appropriate RIR routing registry, RADB, or a RADB-mirrored IRR. These announcements include ASNs and IP prefixes from their own networks along with networks that they provide transit services for.

#### 4.1.2 MANRS Recommended Actions.

- (1) **Prevent traffic with spoofed source IP addresses - Filtering.** [A. et al. 2019] Source address validation should be present for the network operators' infrastructure and end users. Anti-spoofing filtering should be included to prevent packets with an incorrect source IP address from entering or leaving the network.
- (2) **Facilitate routing information on a global scale -RPKI.** [A. et al. 2019] A network operator should create a valid Route Origination Authorization for each IP prefix or set of prefixes it's legitimately authorized and intends to originate.

## 4.2 BGPsec

Due to the lack of built-in security mechanisms in BGP, many secure BGP schemes have been proposed. Due to the complexity of computation and deployment most of them cannot be deployed in practice. BGPsec is one of the most promising secure BGP schemes proposed by the IETF. It gives ASes the ability to perform verification of legitimacy and authenticity of BGP route advertisements. Unlike other solutions like RPKI and S-BGP, BGPsec is baked into BGP which helps mitigate some of the security vulnerabilities. Due to this, there is some additional computational overhead for using BGPsec compared to just BGP for BGP routers. [Noction 2015]

### 4.2.1 How BGPsec Works.

To use BGPsec, two routers must first negotiate using BGPsec confirming compatibility. If BGPsec is used the AS\_PATH attribute from BGP is replaced with BGPsec\_path. This attribute replacement adds some additional info in the communication between routers. The router includes the AS (autonomous system) number that it's sending the update to and also generates a cryptographic signature over the information added to the AS path. This allows the receiving router to verify if the AS path is correct. The Subject Key Identifier in the signature refers to the RPKI certificate of the router that created it. To reap the full benefits of BGPsec, there must be an unbroken path of BGPsec-capable routers. If a regular BGP router is encountered BGPsec\_Path is converted back to AS\_PATH and you lose out on the security benefits. [Noction 2015]

### 4.2.2 BGPsec Vulnerabilities.

BGPsec claims to be secure and provide authenticated prefix origin and routing path announcements in routing updates but it's susceptible to vulnerabilities in the control and data plane. In the control plane, BGPsec aims to prevent blackhole attacks from route hijacking and propagation of forged routes, but even with full deployment of BGPsec it's still susceptible to wormhole attacks. Loop free routing is important for any routing protocol, however attackers can generate forwarding loops and overload network links via a mole attack in the dataplane. BGPsec wasn't made to address data plane vulnerabilities but it's necessary to fix the issue as it violates the correctness of packet forwarding. Lastly, BGP doesn't verify consistency between control and data planes leaving it vulnerable to protocol manipulation attacks. Due to fundamental design flaws with BGP, BGPsec is unable to achieve the desired security properties. [Li et al. 2018]

## 5 CONCLUSION

We were able to demonstrate the BGP Path Hijacking Attack using simulation through Mininet. We've also discussed its current vulnerabilities and different avenues to remedy these vulnerabilities. BGP is a critical backbone of the Internet, yet it has been shown to be vulnerable to hijackers on numerous occasions. Some major recent events that have featured BGP attacks include: Pakistan crashes Youtube in 2008 [McCullagh 2008], the Chinese diversion of U.S military traffic [Anderson 2010], and hijacking Bitcoin [Anderson and Newton 2014]. Even though many solutions from security experts have been proposed they still have flaws. Despite the flaws of this three decade old protocol, BGP still holds strong as the "duct tape" keeping the Internet together.

## REFERENCES

- [1] Robachevsky A., Meynell K., Belson D., Combes J., and Compton R. 2019. *MANRS for Network Operators*. Retrieved Jun 6, 2020 from <https://www.manrs.org/isps/>
- [2] Gail Anderson and Joe Newton. 2014. *Hacker Redirects Traffic From 19 Internet Providers to Steal Bitcoins*. Retrieved Jun 6, 2020 from <https://www.wired.com/2014/08/isp-bitcoin-theft/>
- [3] Nate Anderson. 2010. *How China swallowed 15 % of 'Net traffic for 18 minutes*. Retrieved Jun 6, 2020 from <https://arstechnica.com/information-technology/2010/11/how-china-swallowed-15-of-net-traffic-for-18-minutes/>
- [4] Cloudflare. 2020. *BGP Hijacking*. Retrieved Jun 4, 2020 from [cloudflare.com/learning/security/glossary/bgp-hijacking/](https://cloudflare.com/learning/security/glossary/bgp-hijacking/)
- [5] Zak Doffman. 2020. Russia And China 'Hijack' Your Internet Traffic: Here's ... (April 2020). Retrieved Jun 5, 2020 from <https://www.forbes.com/sites/zakdoffman/2020/04/18/russia-and-china-behind-internet-hijack-risk-heres-how-to-check-youre-now-secure/#76f389f35b16>
- [6] Bob Lantz and Brian O'Connor. [n.d.]. *Download/Get Started With Mininet*. Retrieved Jun 4, 2020 from <http://mininet.org/download/>
- [7] Qi Li, Jiajia Liu, Yih-Chun Hu, Mingwei Xu, and Jianping Wu. 2018. BGP with BGPsec: Attacks and Countermeasures. *IEEE Network* PP (12 2018), 1–7. <https://doi.org/10.1109/MNET.2018.1800171>
- [8] Declan McCullagh. 2008. *How Pakistan knocked YouTube offline (and how to make sure it never happens again)*. Retrieved Jun 6, 2020 from <https://www.cnet.com/news/how-pakistan-knocked-youtube-offline-and-how-to-make-sure-it-never-happens-again/>
- [9] Noction. 2015. *BGP security: the BGPsec protocol*. Retrieved Jun 5, 2020 from <https://www.noction.com/blog/bgpsec-protocol>