BGP Prefix Hijack Attacks - ColoState

Created by: Vamsi Kambhampati and Dr. Daniel Massey, Colorado State University. {vamsi, massey}@cs.colostate.edu

Put your due date here

Contents

Overview

The purpose of this exercise is to introduce you to Border Gateway Protocol (BGP), which is the de-facto inter-domain routing protocol of the Internet, and familiarize you with prefix hijacking attacks. Hijacking attacks pose a significant threat to the Internet and can have devastating consequences on Internet services. However, effective defenses against these attacks have not yet been deployed in the Internet. This exercise will give you first hand account of how the attacks can be perpetrated, and will walk you through a prefix hijacking attack in a small testbed environment. In order to complete the assignment, you will need some basic understanding of routing protocols, understand how IP forwarding works and have experience with Unix command line. Experience with tools such traceroute, netstat, ftp and the Quagga routing protocol suite is useful.

After successfully completing this lab, you should be able to:

1. Understand the basics of BGP routing
2. Understand simple BGP route configuration using Quagga
3. Understand the BGP prefix hijacking attacks
4. Understand a few basics of diagnosing routing problems

Required Reading

Border Gateway Protocol

BGP is the de-factor inter-domain routing protocol of the Internet. The first version of the protocol was introduced in 1989 (RFC 1105) as a replacement to Exterior Gateway Protocol (EGP), and has since gone through several revisions. The current version-4 of the protocol (last updated in 2006) is described RFC 4271.

BGP addresses the problem of exchanging reachability information between Autonomous Systems (AS). BGP speaking routers exchange routing information through a series of BGP *updates*. An originating BGP router announces an IP address *prefix* to its attached neighbors, which in turn propagate the information to other routers until some target router learns about the prefix and a route to reach the destinations in that prefix. BGP is a path vector protocol in which each path element is an Autonomous System Number (ASN). Before a router announces a prefix to its neighbors, it stamps its ASN into an *ASPATH* field of the update message. This way, a BGP router can detect and prevent routing loops (upon receiving a route update, the router will check if its ASN already exists in the ASPATH field of the route; if so, it will drop the route update).

Unlike RIP or OSPF, BGP does not use route metrics, but rather uses a prioritized list of route selection criteria to select from multiple routes to the same destination prefix. The first selection rule is based on local policy of an AS, and the second rule is based on the AS path length. In other words, if two routes to a prefix *p* are equally preferred in an AS, but one route has a shorter ASPATH length, then that route is selected as the preferred route to reach the destinations in prefix *p*. Finally, BGP uses TCP as the underlying transport protocol for carrying routing updates and various other control messages.

Further reading:

- Border Gateway Protocol 4 RFC 4271
- Border Gateway Protocol Wikipedia article

Prefix Hijacking Attacks

The BGP protocol does not specify an authentication mechanism to verify routes. What this means is that any BGP router can announce any prefix as if it owns that prefix, or it could modify the route associated with a prefix to make it more preferable to its neighbors. Network operators explicitly configure BGP routers to establish peering relationships with other ASes to exchange routing information. But, they do not have control over who is allowed inside the BGP "world" (i.e., the global Internet routing space) or what routes they can inject into the global routing space. In addition, the complexity of global routing means network operators may not always know which AS owns a particular prefix, or what path is used to reach that prefix. Even if they do know, the BGP protocol itself does have the necessary mechanisms in place to authenticate prefix ownership (or the path to a specific prefix). A malicious entity may hijack the prefixes of other ASes by either compromising a BGP speaking router or by participating in global routing themselves. In some cases, BGP hijacks happen due to misconfigurations.

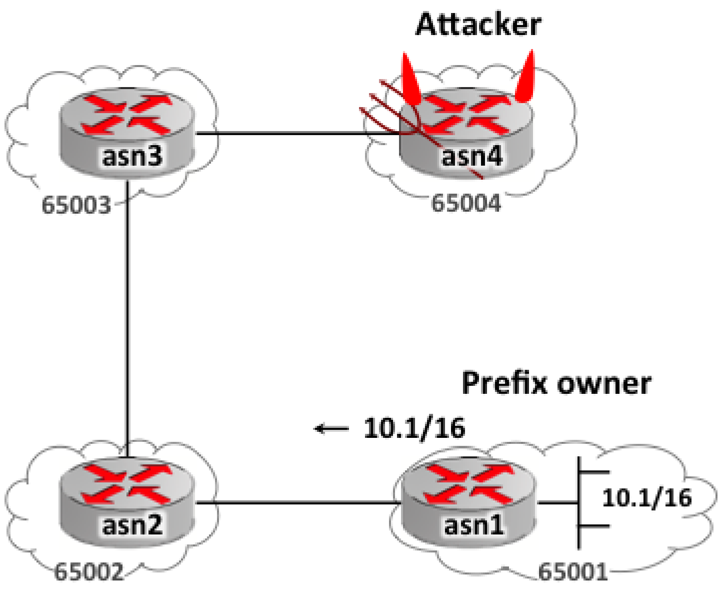

Figure 1: Example of BGP Prefix Hijacking

In a *prefix hijacking* attack, a BGP speaking router announces a direct route to prefix *p* that it does not actually own or is authorize to announce. The neighboring BGP speakers either accept this route and replace the current route they have for prefix *p* or may reject it during the route selection process. To better understand these attacks, consider the example shown in Figure-1. Here, router asn1 is announcing a prefix 10.1/16. The announcement propagates through the network and reaches router asn3, which installs a route for 10.1/16 and sets the next hop to asn2. Now suppose a malicious router asn4 wants to hijack 10.1/16; it simply announces a route for 10.1/16 to its neighbor asn3. Assuming asn3 has not set local policies for 10.1/16, asn3 will select the route for 10.1/16 based on the ASPATH attribute. In this case, the route for 10.1/16 from asn1 has ASPATH:<65001, 65002>, while the route for 10.1/16 from the adversary has ASPATH:<65004>. Since the route from the adversary is shorter, router asn3 will select this route and forward any traffic to the destinations covered by 10.1/16 to router asn4.

A special case of prefix hijacking attack is *subprefix hijack* attack. In this case, the attacker will announce a subprefix of a more general prefix to hijack part of the address space of the originating AS. Since the attacker is essentially announcing a new prefix, and since IP forwarding prefers longer prefixes, all traffic associated with the destinations covered by the subprefix will flow towards the attacker. Considering the example in Figure-1 again, let us say this time the attacker announces a route for 10.1.1/24 (i.e., subprefix of 10.1/16). Router asn3 will learn about both 10.1/16 (next hop: asn2) and 10.1.1/24 (next hop: asn4). However, when a packet arrives at asn3 for destination 10.1.1.1, it will forward the packet to asn4 due to longest prefix matching. On other hand, if a packet arrives at asn3 for destination 10.1.2.1, it will forward it to asn2 since that address matches the prefix 10.1/16.

To learn more about prefix hijacking, refer:

- IP Hijacking Wikipedia article
- PHAS: A Prefix Hijack Alert System Academic paper (optional reading)
- iSPY: Detecting IP Prefix Hijacking on My Own Academic paper (optional reading)

Software Tools

Quagga Routing Suite

Quagga is a cross-platform routing software package which supports many routing protocols including RIP, RIPng, OSPFv2, OSPFv3 and BGP. Quagga can be installed on a general purpose machine to turn it into a software based router. Moreover, it provides a simple interface to setup static routing on a host. For this lab exercise, we are particularly interested in the BGP routing part of Quagga. Throughout the exercise we assume Quagga as running on Ubuntu Linux.

Quagga stores its configuration files in `/etc/quagga/` directory, and has a separate configuration file for each of the routing protocols it supports. For instance, the BGP routing configuration is stored in `/etc/quagga/bgpd.conf`. The configuration file `/etc/quagga/daemons` specifies which of the routing protocol daemons are enabled on the host (set to "yes" to enable a protocol). Configuration information can be setup manually before starting a particular protocol daemon, or they can be entered in real-time using a special command line utility called `vtysh`. Quagga also supports a `telnet` interface to enter configuration commands in real-time. For example, the BGP instance can be configured in real-time by telnetting to `localhost` and specifying `bgpd` as the port number (`bgpd` is an alias to port 2605).

The following is the BGP configuration file on one of the testbed hosts.

```
!
hostname asn1
password test
enable password test
```

```
bgp config-type cisco
!
!
router bgp 65001
no synchronization
bgp router-id 10.1.0.1
network 10.1.0.0/16
neighbor 10.2.0.2 remote-as 65002
!
```

Note that the BGP configuration used here is overly simple. Real Internet BGP routers have many more options and their configuration can be quite complex. For more details follow the Quagga BGP documentation link below. The command `router bgp 65001` sets up the BGP instance on a (software-)router named **asn1** that belongs to ASN **65001**. The command `network 10.1.0.0/16` tells this router to announce a route for prefix **10.1/16** to its neighbor(s). The neighbor itself is specified using the command `neighbor 10.2.0.2 remote-as 65002`, i.e., the neighbor's IP address is **10.2.0.2** and its ASN is **65002**. Note that in order to exchange BGP routes, the router asn1 must have a direct connection to neighboring router 10.2.0.2, or it must be reachable from router asn1 (using perhaps some intermediate hops). The neighboring router will also have a similar configuration entry for router asn1. Finally, the login password and the enable mode password for asn1 are set to "**test**".

To change the configuration in real-time on asn1, you can use the telnet interface. The following commands show how to remove the route for prefix 10.1/16 on asn1. Note that Quagga does not allow telnet from remote machines by default, so the user must be logged in to the machine configured as the router (in this case asn1).

```
telnet localhost bgpd
enable
config terminal
router bgp 65001
no network 10.1.0.0/16
end
```

Use the password "test" to both login via telnet and to enter the "enable" mode.

Quagga documentation, including BGP configuration can be found in the following link:

- Quagga Software Routing Suite

Traceroute

The `traceroute` tool prints the route packets take to a destination host. Internally, traceroute sends a series of messages (commonly ICMP) with IP TTL value set to the number of hops it wants to discover. Since the TTL expires upon reaching the target hop, the node at that hop will respond with an error message. This way, traceroute can figure out the path to a destination. Note that some intermediate hops may not respond with an error message, in this case traceroute will print a "*". For more information about traceroute and its options, check out man page (on any Linux machine): `man traceroute`

Netstat

The `netstat` command line utility can be used to print the IP forwarding table of a host. It can also be used to print several useful information about a host's network activity, including listing active connections for each protocol, packet statistics and so forth. For more information about netstat and its options, check out man page (on any Linux machine): `man netstat`

Introduction

The exercise is divided into three parts. In the first part, you will observe the characteristics of the network when there is no hijacking attack and retrieve a file from an ftp server. In the second part, you will assume the role of an attacker and perform prefix hijacking attack. The goal here is to take over the address of the ftp server and serve a forged document. If you were a real attacker, you can do anything from stealing passwords to distributing malware. In the last part, you will perform subprefix hijacking attacks and observe the differences between prefix hijacking and subprefix hijacking.

The exercise will carefully walk you through BGP configuration to conduct the attacks and will also teach you various tools to identify routing information on hosts and routers.

Assignment Instructions

Network Information

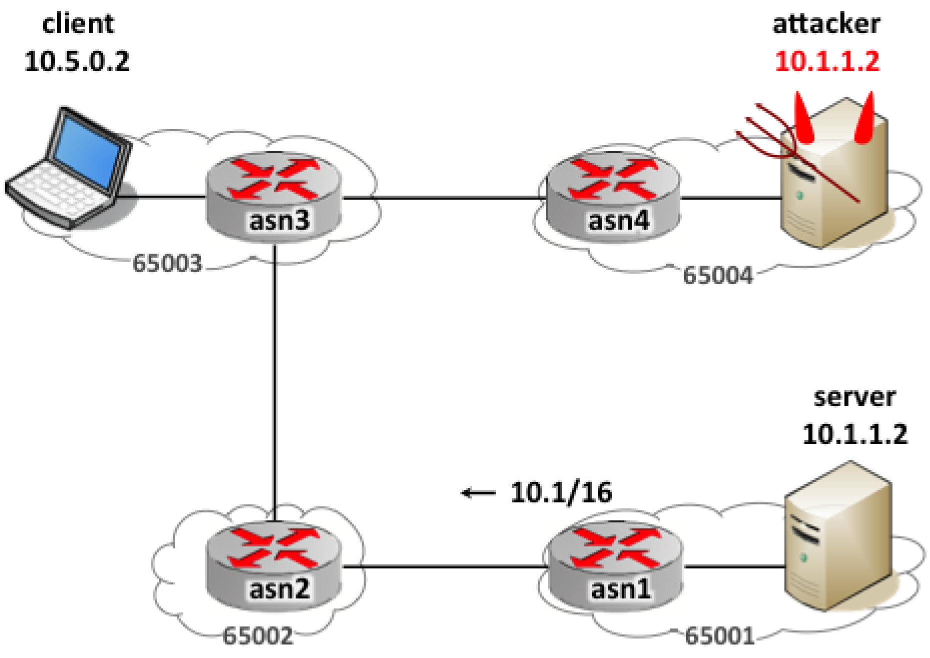

Figure 2: Testbed network on Deter

The network in this exercise is designed to simulate a small scale Internet with several Autonomous Systems and a simple client-server scenario. Figure 2 shows the network, where each AS has a single router. The client, server, and attacker hosts all directly attach to their adjacent routers inside the respective ASes. The ftp server attaches to router asn1 (which belongs to AS 65001) and has the IP address 10.1.1.2, while the client attaches to router asn3 (AS 65003) and has the IP address 10.5.0.2. The attacker ftp server attaches to router asn4 (AS 65004) and also has the IP address 10.1.1.2. Initially, the attacker ftp server is not accessible from the anywhere in the network except router asn2 i.e., only router asn4 knows about the attacker. The router asn1 announces a route for 10.1/16 network, which covers the real ftp server 10.1.1.2, therefore any packets sent to this address will reach the ftp server (until the prefix is hijacked).

Setup

1. If you don't have an account, follow the instructions in the introduction to DETER document.
2. Log into DETER.
3. Create an instance of this exercise by following the instructions here, using `/share/education/BGPHijack_ColoState/bgphijack.ns` as your NS File.

- In the "Idle-Swap" field, enter "1". This tells DETER to swap your experiment out if it is idle for more than one hour.
- In the "Max. Duration" field, enter "6". This tells DETER to swap the experiment out after six hours.

4. Swap in your new lab.
5. After the experiment has finished swapping in, you may log in to the nodes using ssh. Follow the instructions here if you are unsure how to login.

The majority of this exercise is easiest to complete simply via command line on the relevant virtual machines.

Tasks

Part 1: Understanding the Topology

Run the following commands on `asn3` and `asn2` machines to remove a specific route injected by the kernel:

```
sudo ip route del 10.1.1.0/24
```

Login to the `client` machine and perform the following tasks:

1. On the command prompt run: `traceroute -n 10.1.1.2` and record the output of the command (if you see * *s in the output, run the command again). Explain the path from client host 10.5.0.2 to the ftp server host 10.1.1.2. Specifically, note down the intermediate hops and their IP addresses. How many hops away is the ftp server from the client?
2. Run `netstat -rn` and record the output. Explain how the client is able to send packets to 10.1.1.2, i.e., what route is the client using to reach the server 10.1.1.2 (don't forget to list the gateway address and mask value).
3. Run `sudo vtysh -c "show ip route"` and record the output. Does the "information" (not the raw output) differ from the above output? If so, what additional information can you learn from this output?
4. Run `ftp 10.1.1.2` and at the prompt for username type `anonymous`, type some random text for password. Once you are connected to the ftp server, type `get README` at the ftp prompt. After the README file finishes downloading, logout (type `exit`) and read and contents of the README file. What does it say?

Now login to `asn3` machine and perform the following tasks (it helps to keep a separate terminal for each host you need to login):

5. Run `sudo vtysh -c "show ip bgp"` and record the output. What is the AS path to reach 10.1/16 ?

Login to `asn2` machine and perform the following tasks:

6. Run `sudo vtysh -c "show ip bgp"` (no need to record output). What AS path will be used to reach an IP address 10.1.1.2? What AS path will be used to reach an IP address 10.1.2.2?

Part 2: Prefix Hijacking

In this part, you will become the adversary and take over the prefix 10.1/16. Your goal is to mislead the client into accessing your false ftp server. To hijack the prefix, first login to `asn4` and run the command `telnet localhost bgpd`. Enter "test" when prompted for the password. You will then get a prompt from the BGP instance running on asn4. At this prompt, run the following series of commands:

```
enable    #type "test" when prompted for password
config terminal
router bgp 65004
network 10.1.0.0/16
end
exit
```

Then on `asn4` type:

```
sudo iptables -t nat -F
sudo iptables -t nat -A PREROUTING -d 10.1.1.2 -m ttl --ttl-gt 1 -j
NETMAP --to 10.6.1.2
sudo iptables -t nat -A POSTROUTING -s 10.6.1.2 -j NETMAP --to
10.1.1.2
```

These lines will ensure that `asn4` rewrites source and destination IPs to hide the presence of hijacking and also to make the `attacker` node properly process packets sent to 10.1.1.2.

After completing this process, wait at least 5 minutes (so that the routes are propagated throughout the network) and log back into the `client` host. Now, do the following:

1. Run `traceroute -n 10.1.1.2` and record the output of the command (if you see * *s in the output, run the command again). Explain the path from client host 10.5.0.2 to the ftp server 10.1.1.2. How many hops away is the ftp server from the client this time? Is there a difference in output from the same command in Part-1?
2. Run `ftp 10.1.1.2` and at the prompt for username type `anonymous`, type some random text for password. Once you are connected to the ftp server, type `get README` at the ftp prompt. After the README file finishes downloading, logout (type `exit`) and read and contents of the README file. What does it say? Did the contents of README file differ from the output in Part-1?

Login to `asn3` machine and perform the following tasks:

3. Run `sudo vtysh -c "show ip bgp"` and record the output. What is the AS path to reach 10.1/16? Did the AS path differ from the last time (i.e., part-1)?

Login to `asn2` machine and perform the following tasks:

4. Run `sudo vtysh -c "show ip bgp"` (no need to record output). What AS path will be used to reach an IP address 10.1.1.2? What AS path will be used to reach an IP address 10.1.2.2?

Part 3: Subprefix Hijacking

In this part, you will become the adversary again and take over a subprefix (10.1.1/24) of the prefix 10.1/16. You will achieve a similar goal as before i.e., mislead the client into accessing your server, but there are some important differences. To hijack the subprefix, login to `asn4` and run the command `telnet localhost bgpd`. Enter "test" when prompted for the password. You will get a prompt from the BGP instance running on asn4. At this prompt, run the following series of commands:

```
enable    #type "test" when prompted for password
config terminal
router bgp 65004
no network 10.1.0.0/16
network 10.1.1.0/24
```

```
end
exit
```

After completing this process, wait at least 5 few minutes (so that the routes are propagated throughout the network) and log back into the `client` host and do the following:

1. Run `traceroute -n 10.1.1.2` (no need to record output). How many hops away is the ftp server 10.1.1.2 from the client this time? Is there a difference in output from the same command in Part-2?
2. Run `ftp 10.1.1.2` and at the prompt for username type `anonymous`, type some random text for password. Once you are connected to the ftp server, type `get README` at the ftp prompt. After the README file finishes downloading, logout (type `exit`) and read and contents of the README file. Did the contents of README file differ from the output in Part-2?

Login to `asn3` machine and perform the following tasks:

3. Run `sudo vtysh -c "show ip bgp"` and record the output. What is the AS path to reach 10.1/16? Did the AS path differ from Part-2? What is the AS path to reach 10.1.1.0/24?

Login to `asn2` machine and perform the following tasks:

4. Run `sudo vtysh -c "show ip bgp"` (no need to record output). What AS path will be used to reach an IP address 10.1.1.2? What AS path will be used to reach an IP address 10.1.2.2?

What can go wrong

Specific problems:

- If you not able to connect to the ftp server, or if cannot ping any of the nodes, then most likely the bgp routing daemon is not working. First, run `ps -ef | grep bgpd` , if did not see the `bgpd` task, run the following command on each of the machines (i.e., client, server, attackers, asn1 through asn4): `sudo /etc/init.d/quagga restart`
- If traceroute is producing * * *s for intermediate hops, run the command once again.
- Do not run traceroute without the "-n" command line option. This option will prevent traceroute from resolving IP addresses to names. We dont need the names for this exercise.
- You must wait at least 5 minutes after making BGP configuration changes. Otherwise, you might run into temporary routing loops during the BGP route convergence process. During this time, your commands will produce incorrect results.

Submission Instructions

Submit a .pdf file to your instructor containing your answers to:

1. Part 1 Questions 1-6
2. Part 2 Questions 1-4
3. Part 3 Questions 1-4