To: William H. Flathead III

From: Matteo Franzil

Subject: BGP Hijack explaination

Date: October 24, 2021

I am writing this memo, following the latest ones on vulnerabilities, in order to discuss about some security flaws that our Autonomous System may suffer of.

To begin with, our whole corporate network is actually quite big, so big that several decades ago - as you may recall - signed a contact with RIPE in order to receive our own Autonomous System classification. We set up our external router system in order to let other AS in the whole word that we existed, and how to reach us. That's how BGP - Border Gateway Protocol - works in a nutshell, at least it's external eBGP version. Periodically, our subnets are advertised with probes to neighboring routers of other ASs. Upon learning of our routes, such routers disseminate this information by flooding to other neighbors and so on. This also works the other way: for example, our FrobozzCo router may learn about QuendorCo's ones, and tell other neighbors that to reach QuendorCo there exists a path passing through us.

This is a very simple explaination, but enough to illustrate what may be a very powerful drawback in this setup. In our testbed, we set up a seven-machine network. This is its simplified diagram. Numbers above boxes indicate the AS number assigned.

```
           65003                               65004
|--------------------------------|     |--------------------------------|
| client             asn3        |     | asn4               attacker    |
| 10.5.0.2/24 ------ 10.5.0.1/24 |     | 10.6.1.1/24 ------ 10.6.1.2/24 |
|                    10.4.0.1/24-|-----|-10.4.0.2/24                    |
|                    10.3.0.1/24 |     |--------------------------------|
|-------------------------|------|
                          |
                   65002  |                            65001
                  |-------|------|     |--------------------------------|
                  |  asn2 |      |     | asn1               server      |
                  |  10.3.0.2/24 |     | 10.1.1.1/24 ------ 10.1.1.2/24 |
                  |  10.2.0.2/24-|-----|-10.2.0.1/24                    |
                  |--------------|     |--------------------------------|
```

First, we assumed that client with IP `10.5.0.2/2` wanted to retrieve a file stored in the FTP server `10.1.1.2/24`. AS work together in order to provide a coherent path that first goes through `asn3`, `asn2`, and finally `asn1` and reaching the server safely. This is standard behaviour, and we can verify it with some tracerouting and checking of BGP tables (please consult `instructions.pdf` for a more thorough explaination and outputs of such commands).

However, assume that an attacker wanted to pretend his IP was not 10.6.1.2/24 but 10.1.1.2/24. Even worse, this rich attacker also owned his own AS - in this case, 65004. BGP is a very scalable protocol, but brings its own risks: without authentication, anyone can advertise anything and the neighbors will believe it.

So, such an attacker could set up its `asn4` router in order to advertise the route for the `10.1.0.0/16` network through itself, even though this is not true at all. This approach, however, failed in this setup as the `asn3` was smart enough to advertise smaller prefixes. Let me explain exactly what was going on.

A quick digression: in BGP, when a router needs to make a routing decision and there are conflicting routes, it will send the packet towards the route with the longest matching prefix. In our case, both `asn4` and `asn2` advertise a route for `10.1.0.0/16`, but `asn2` also advertise a route for `10.1.1.0/24` which is longer. This means that when a packet for the first subnet is received, is sent through the hijacked route - since the path is also shorter - but when it matches the second subnet, `10.1.1.0/24`, then it is still sent through the original route.

In order to “fix” this, the attacker can therefore stop advertising the shorter route and start advertising the `10.1.1.0/24` one. Following this change in setup, the client's packet will start being routed through the attacker's AS. This can be verified with another traceroute, which will now be a hop shorter (`client -> asn4 -> asn4 -> att.`). At this point, all hope is lost. With some spoofing, the attacker will pretend to be the server and start serving, in our case, malicious FTP files.