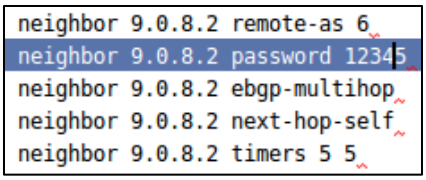
This project demonstrates effectiveness of BGP Hijack Attacks. A malicious Autonomous System is set up to mount an attack through false BGM announcements from a rogue AS. This causes victim ASes to route traffic bound for another AS through the malicious AS. This works because the malicious AS advertises a shorter path to reach a prefix [1].

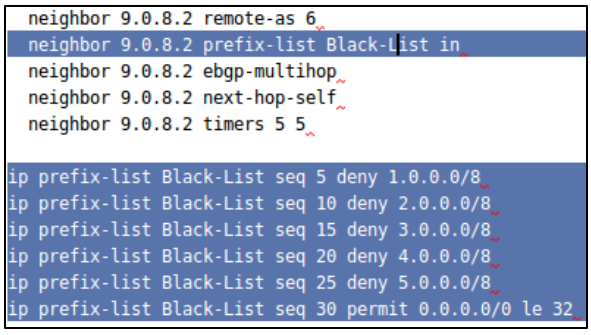
The paper by Nordstrom and Davrolis references many different possible countermeasures for BGM Hijacking. Some of the countermeasures discussed include BGP TTL Security Hack, TCP MD5 Signature, Unicast Reverse Path Filtering, MaxPrefixLimit Feature, Route Filtering and S-BGP [2].

For Part 4 of this project, the TCP MD5 Signature and Route Filtering were attempted to be implemented side by side. MD5 neighbor authentication ensures only authorized connections can be established. MD5 authentication signatures must match on both sides of the connection, but for the purpose of simplifying the demonstration of this feature, all routers 1-5 are set with the same authentication, while the attacking router has an unknown password, since it has no visibility into possible authentication setup of the rest of the network. If the two peers do not have the same MD5 authentication, the packets being sent are discarded. Similarly, Route Filtering with Prefix Lists prevents a BGP from installing unwanted prefixes in its routing table. Prefix Lists can be configured for both inbound and outbound directions. Outbound prefix lists are generally configured with prefixes a BGP is allowed to broadcast to, whereas Inbound prefix lists are configured with allowed inbound connections.

To demonstrate MD5 Authentication, all “pgbd-R*.conf” files 1-5 are modified to include a line such as “neighbor 9.0.x.x password 12345” for each neighbor in the BGP. For the file “bgpd-R6.conf” instead, the line “neighbor 9.0.x.x password pwd” is added. The reason for the password in R6 neighbors being different form al others is to simulate the fact that the attacking AS does not know the authentication keys inside the network. As such, the password used internally can be anything, as long as the healthy connections match, whereas the passwords included in R6 neighbors must not match any, in order to properly demonstrate packet dropping from R6 to the rest of the network.



To demonstrate Prefix List Route Filtering, a prefix list is added to “bgpd-R5.conf” file. Additionally, the line “neighbor 9.0.x.x prefix-list <list-name> in” is added only to the neighbor connection to AS6. The inbound prefix filtering is chosen to block all AS addresses which currently already have connections on the network from making a connection through this neighbor. In other words, anyone making a connection through AS6 should not be trying to look like any of the other ASes on the network. This works to filter the attacking AS as it attempts to look behave like AS1. Only bgpd-R5 configuration file is modified since R6 connects directly to it and the effects of blocking the attacking AS can be easily seen directly from R5.



In order to test the aforementioned countermeasures, perform the following steps:

For MD5 Authentication,

1. Follow all instructions from Project Part 3 to observe “*** Attack web server ***” messages being sent from attack AS.
2. Copy the entire folder/file set from Part 3 of the project to a new folder.

3. Make all the edits described for MD5 Authentication, but DO NOT make the edits required for Prefix List Route Filtering.
4. In terminal 1, type “sudo python bgp.py”
5. In terminal 2, type “./connect.py”, following by password “en” and command “sh ip bgp”.
6. In terminal 3, type “./website.sh”.
7. In terminal 4, type “./start_rpgue.sh”.
8. In terminal 3, Observe “*** Attack web server ***” message is NOT displayed in message run.
9. In terminal 2, type “sh ip bgp” and observe that attack AS is not connected.
10. In terminal 4, type “./stop_rogue.sh”.

For Prefix List Route Filtering,

1. Follow all instructions from Project Part 3 to observe “*** Attack web server ***” messages being sent from attack AS.
2. Copy the entire folder/file set from Part 3 of the project to a new folder.
3. Make all the edits described for Prefix List filtering, but DO NOT make the edits required for MD5 Authentication. (Alternatively, leave the MD5 authentication code in, and simply set the password in AS6 the same as the rest of the network, giving it access from MD5 authentication perspective and only blocking access using Prefix Filtering).
4. In terminal 1, type “sudo python bgp.py”
5. In terminal 2, type “./connect.py”, following by password “en” and command “sh ip bgp”.
6. In terminal 3, type “./website.sh”.
7. In terminal 4, type “./start_rpgue.sh”.
8. In terminal 3, Observe “*** Attack web server ***” message is NOT displayed in message run.
9. In terminal 2, type “sh ip bgp” and observe that attack AS is not connected.
10. In terminal 4, type “./stop_rogue.sh”.

References

[1] Lad, M., Oliveira, R., Zhang, B., & Zhang, L. (n.d.). Understanding Resiliency of Internet Topology Against Prefix Hijack Attacks . Retrieved April 18, 2020, from https://www2.cs.arizona.edu/~bzhang/paper/07-dsn-hijack.pdf

[2] Nordstrom, O., & Davrolis, C. (n.d.). Beware of BGP Attacks. Retrieved April 18, 2020, from https://www.cc.gatech.edu/~dovrolis/Papers/ccr-bgp.pdf

[3] Cisco Security Threat and Vulnerability Intelligence. (2014, November 10). Retrieved from https://tools.cisco.com/security/center/resources/protecting_border_gateway_protocol#6