bgpAND - Architecting a modular BGP4 Attack & Anomalies Detection Platform

Mayank Bhatnagar

TechMahindra Limited, SDF B-1, NSEZ, Noida-201305, India
E-mail : mayank.bhatnagar2@techmahindra.com

Abstract **- Border Gateway Protocol (BGP) is an Autonomous System (AS) routing protocol. It forms the backbone of Internet core routing decisions. However, it is also equally prone to security issues and several attempts to attack & exploit the protocol have been noted. bgpAND is a platform designed and developed to analyze BGP updates, and detect anomalies and carry out attack filtrations. It is a modular platform such that new attack detection mechanisms can be implemented & integrated. bgpAND can also be linked to BGP routers providing real-time BGP updates and hence can be used as a live security solution.**

Index Terms—BGP Security, Bogon Filtering, Autonomous System Anomaly Detection.

I. INTRODUCTION

The "Internet" is as we all know a network of networks. The glue binding these networks are several network elements, communicating with each other, in a predefined, standardized mechanism, we generally refer to as "protocols".

The communicating information encapsulated in form of network "packets" flow across from one end to another, getting "routed" in the Internet, being controlled by a number of Internetworking entities, prime amongst them being "routers". The router, having major functionality to "route" these packets, carry out the process seamlessly, continuously working, cooperating, updating, forwarding these packets, packet by packet, to another route, taking decisions on these routes. In this scenario, two things are needed; first, *routing tables* match destination addresses with next hops. Second, *routing protocols* determine the contents of these tables.

Apart from the normal functioning of the routes, "security" of the routers, routing information, routing protocols plays a paramount role in managing a secure networking infrastructure. Since it's usage in the industry, there have been several attempts to attack, exploit the protocol and thereby disrupt the routing infrastructure.

This work has been carried out as part of a holistic attempt to order to analyze the BGPv4 security issue & attacks in detail and develop a platform, an environment to detect the BGP insecure updates in general and anomalies in particular. Algorithms have been developed and a tool BGP4 Anomalies Detection hereafter referred as bgpAND, has been developed which is modular & extendable in design.

This paper describes some of the features implemented in bgpAND tool an also proposes algorithms and modules for future implementations.

The paper starts with discussing general security issues surrounding BGPv4 protocol. Further on bgpAND's architecture is described in the second section, while the design and development work is discussed in the third section. The fourth section summarizes the results & observations. In the fifth section, the paper concludes in addition to providing future work that is planned and can be done in the sixth section.

BGPv4 Security Issues

Border Gateway Protocol (BGP) is an inter-Autonomous System routing protocol. It backs the core routing decisions on the Internet. Most Internet Service Providers (ISP)s use BGP to establish routing between one another. Even though most Internet users may not use it directly, it anyway is one of the most important protocols of the Internet.

BGP version 4 has been described in [1]. Since it's usage in industry, there have been several attempts to attack and exploit the weaknesses in the protocol. Studies like [2],[3] have at length analysed and studied various attacks including Peer & Session attacks, route

flapping & Route Deaggregation attacks, various route injection & denial of service attacks. Various tools [4], monitoring scripts & solutions have been developed including GrepCIDR, Pretty Good BGP (PGBGP), PHAS and others which demonstrate and detect attacks and help in useful configuration and monitoring. Recent studies in [3], which carry out large scale comparisons, have observed with facts that BGP is vulnerable to communications interruptions and failures and finding suitable improved security measures with acceptable costs is difficult. Continued threat analysis & attack detection solutions to secure and develop robust BGP are needed. The reader is encouraged to refer [2], [3] for details on BGPv4 vulnerabilities and security issues & effective security mechanisms &solutions

III. BGPAND ARCHITECTURE

The following architecture (Figure 1) is proposed for development and modeling to detect BGP attacks in real time at a BGP speaker. This architecture describes how BGPMon[5], a BGP monitoring tool, will be used to work on real time RIP data [6] and algorithms and solutions developed to add on to the BGP security capabilities.

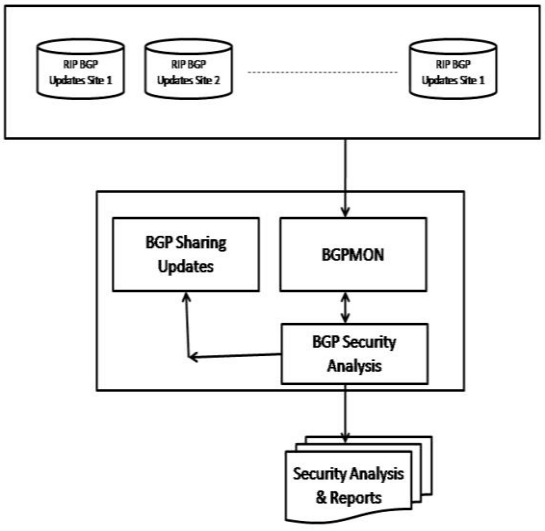

Fig 1 High Level bGPAND architecture with BGPMON[5] integration

In this architecture, the BGPMOn is to be used to analyze in real time the BGP updates available from RIP database.

For initial experiments, the available RIP data is analyzed and studied. Since BGPMOn has capability to detect RIP binary data and BGP updates in real time, it is useful for carrying out detection development work.

In the current work, however the BGPMon has not been configured to receive the updates in the form of a BGP Speaker. However, a native client has been implemented which will receive the BGP updates from the BGPMon installed at various sites. This receives the RIP updates, or BGP updates and carries out analysis.

III. DESIGN & DEVELOPMENT OF DETECTION MECHANISMS

This section discusses the design and development of bgpAND's detection mechanisms.

The high level flow of BGP updates data, it's analysis & results is depicted below in Figure 2;

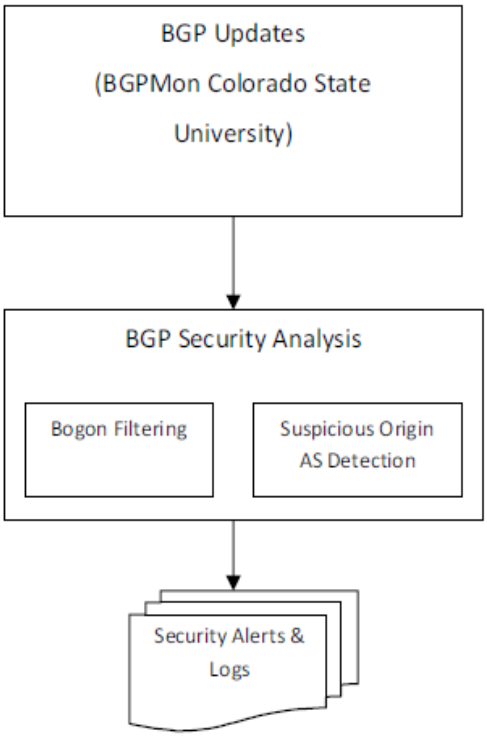

Figure 2 BGP Updates Analysis & Detection Mechanisms

The following points describe the implemented architecture.

- Live Routing Updates i.e. BGPMon live data, from Colorado State University [5] are being downloaded, by making a TCP connection to the described server, *livebgp.netsec.colostate.edu*, at port 50002. This will download a steady stream of BGP updates being received from the BGPMon installed at their end. These BGP updates are in the form of XML, in standard XFB format, XML format for BGP.

- Processing of this XML stream of data has been done and there exists a separate TCP client program which writes this data into a file.
- This TCP client will connect to BGPMon speaker installed at Colorado State University and download the BGP stream, having BGP updates in XML manner, into a file.

No BGP Updates are being sent as of now from bgpAND or shared with any of the BGP routers or speakers.

Implemented Detection Mechanisms

In the current implementation of bgpAND, the following two security mechanisms have been implemented

a. Bogon Filtering

The site [7] maintains a list of Bogon addresses. [7] describe Bogon address as follows

"A bogon prefix is a route that should never appear in the Internet routing table. A packet routed over the public Internet (not including over VPNs or other tunnels) should never have a source address in a bogon range. These are commonly found as the source addresses of DDoS attacks."

This detection mechanism identifies for any BGP message, whether the source or destination addresses that the BGP Message is being coming from or to, is not actually a BOGON or Bogus or a harmful IP address. The harmfulness or maliciousness of this Bogus IP addresses is being done by maintaining a list of Bogon IP addresses being downloaded from International Bogon address filtering sites.

Hence it is important for the system to keep bogon lists maintained and updated frequently.

For each & every message being received in the BGP stream, the extraction of Source and Destination IP address is being done and if any match occurs with an address mentioned in the BOGON Prefix list, it is alerted and logged.

For Prefix matching with IP address, a regular expression match has been performed. An open source tool, grepcidr[8] has been installed and being used.

The flow diagram of the detection is being depicted below in Figure 3

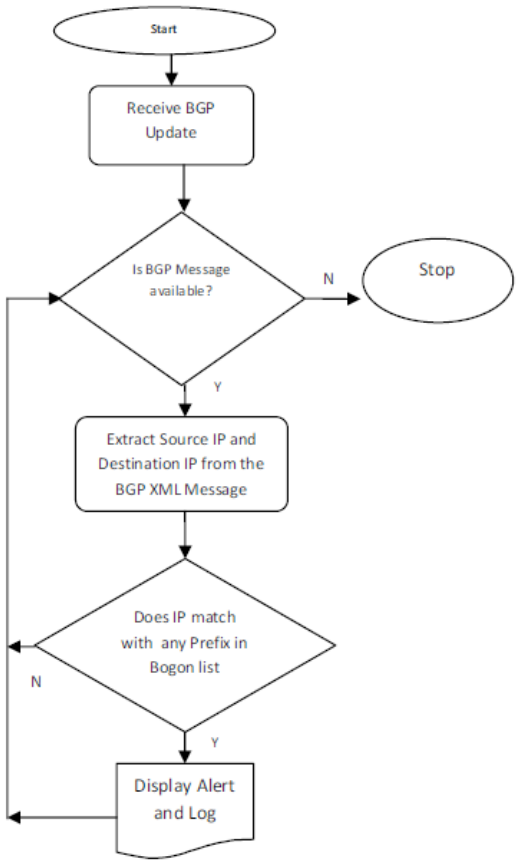

Figure 3 System Flow Bogon Filtering

b. Suspicious Origin AS Detection

This detection mechanism identifies for any BGP message, whether the Origin AS for any prefix gets changed. This is a typical suspicious anomaly behaviour that has been documented and discussed among various security experts and researchers. [9] mentions that one of the popular BGP security detection system (PHAS) or Prefix Hijack Alert system, implements the Origin AS Anomaly detection.

In this type of detection mechanism, a BGP update stream is being monitored. For each BGP Update, an alert is displayed and logging is being done, if the following behavior [9] is observed

"An origin AS or Autonomous System, in an update message that is new relative to the set of previously observed set of origin ASes for the same prefix".

The interpretation of the above detection logic is that for any AS, or any BGP device receiving BGP update, connected to other ASes, there exists a specific path or NLRI (Network Layer Reachability

Information). If there is any change in the NLRI, a AS sends a BGP update. However it is very unusual to see that a BGP update is being received for any origin AS being changed for a network prefix

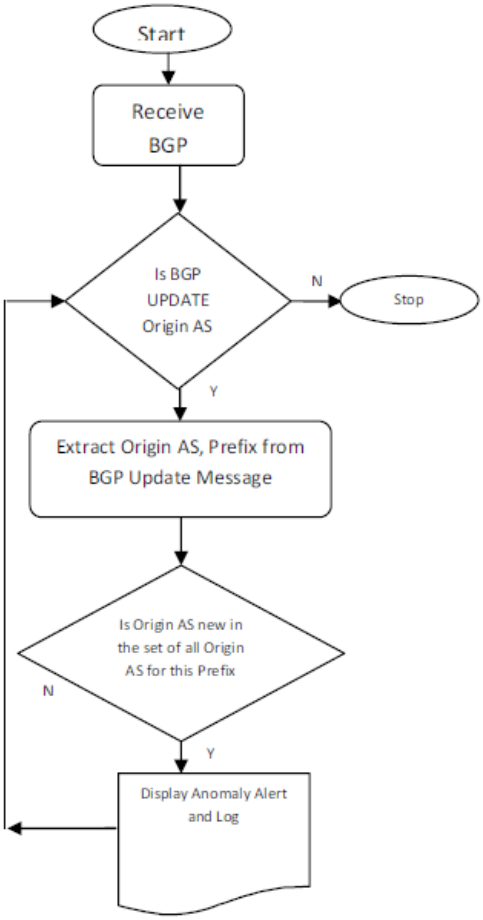

Figure 4 System Flow Suspicious Origin AS Detection

To implement this detection, a list of origin ASes are being prefixes are being maintained for each new prefix being observed. Whenever, for a prefix, which is NOT NEW, an Origin AS BGP update is being observed, it is being flagged as an Anomaly or Alert.

The flow of the detection is similar to the above. However this detection works only for BGP Update Messages and uses AS_TYPE, AS_SEQUENCE, NLRI and Prefix XML Nodes to extract the required information nodes.

The flow diagram of the detection is being depicted in Figure 4;

IV. BGPAND WORKING & OBSERVED RESULTS

This section describes the implementation as it looks like and the results observed

Initial Input Screen

The initial screen is as shown below, Figure 5. It describes how input can be given and what detection mechanisms have been implemented.

```
**************************************************
***         BGP Updates Anomaly Detection          ***
***                                                ***
***                                                ***
***                                                ***
***                                                ***
***                                                ***
*** 1. Bogon Address Filtering Detection [bogon]   ***
*** 2. Origin AS Anomaly Detection       [origas]  ***
***                                                ***
***                                                ***
***                                                ***
***                                                ***
**************************************************
*** Usage: bgpAND [-bogon][-origas] <docname.xml>
*** Input file: <docname.xml> contains BGP Updates
**************************************************
Usage: ./bgpAND [-bogon][-origas]<docname.xml>
Default Detection System: Bogon Address Filtering
Example Usage 1: bgpAND -bogon bgpupdate.xml
Example Usage 2: bgpAND -bogon -origas bgpupdate.xml
```

Figure 5 Initial Input Screen

Program Input

The BOGON detection for IP addresses can be specified as follows;

```
Usage: ./bgpAND [-bogon][-origas]<docname.xml>
Default Detection System: Bogon Address Filtering
Example Usage 1: bgpAND -bogon bgpupdate.xml
Example Usage 2: bgpAND -bogon -origas bgpupdate.xml
mayank@ganesh:~/Work/BGP-Project/bgpmain$ ./bgpAND -bogon bgpMonData
```

Figure 6 Command line Input for Bogon address filtering Detection

Here *–bogon* specifies for Bogon Prefix address Detection.

Program Result

The detection results are being stored in a file *"result_BOGON_filter.txt"*. If any match occurs, the details of the logs are being stored in this file. It is being shown here;

```
*******************************************************
Docname = bgpMonData
Anomaly Detection Options are BOGON=1 ORIGAS=0
BOGON ADDRESS NOT MATCHED
[ALERT]BOGON_DETECT::187.16.216.24::ACTION->FILTER
BOGON ADDRESS NOT MATCHED
[ALERT]BOGON_DETECT::187.16.216.24::ACTION->FILTER
BOGON ADDRESS NOT MATCHED
[ALERT]BOGON_DETECT::187.16.216.24::ACTION->FILTER
mayank@ganesh:~/Work/BGP-Project/bgpmain$
```

Figure 7- Bogon Filter detection

Suspicious Origin Autonomous System (AS) detection

As mentioned above, this detection mechanism identifies for any BGP message, whether the Origin AS for any prefix gets changed.

Program Input

The Origin AS detection for BGP Update messages can be specified as follows;

```
mayank@ganesh:~/Work/BGP-Project/bgpmain$ ./bgpAND -origas bgpMonData
```

Figure 8 Command line for Origin AS anomaly detection

Here *–origas* specifies detection for BGP Origin AS, while bgpAND is the program name, bgpMonData is the file which is storing BGP Stream.

Program Result

The detection results are being stored in a file *"resultOriginAS"*. If any detection of the above case occurs, the alert is raised and the details of the logs are being stored in this file. It is being shown in Figure 8;

Anomaly alert message

[ANOMALY]SUSPICIOUS_ORIGIN_AS_DETECT::PREFIX::65.210.166.0/24::ORIG_AS::1673::ACTION->ANALYSE

This states that an anomaly of type ORIGIN_AS has been detected for prefix 65.210.166.0/24 caused by a suspicious new Origin AS 1673. The action for which has to be analysis of the alert.

```
[ANOMALY]SUSPICIOUS_ORIGIN_AS_DETECT::PREFIX::65.210.166.0/24::ORIG_AS::1673::ACTION->ANALYSE

Printing List
####################

[Node 0]
Prefix::65.210.166.0/24
OrigAS Array::[16735][1673][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][(
*********************
[Node 1]
Prefix::64.210.166.0/24
OrigAS Array::[16735][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][(
*********************
mayank@ganesh:~/Work/BGP-Project/bgpmain$
```

Figure 9– Detection Results

Additionally Detailed Debugging macros have also been built into the program for efficient debugging

V. Conclusion

The basic system implementation in form of bgpAND, aims to ensure that the BGP updates are being analysed and any anomalies and issues related to bogon traffic being detected, alerted and logged. The two detection mechanisms implemented can be put on and used against any live BGPMon system and used for detecting any Bogon alerts or suspicious OriginAS anomalies.

bgpAND's can be used to integrate with any standardized BGP updates and reslts can be analysed. However, incorporation of these algorithms to detect any issues in real time is of utmost importance.

VI. Further Work

bgpAND can be used to integrate with any standardized BGP updates and results can be analyzed. However, incorporation of these algorithms to detect any issues in real time is of utmost importance. The following work is planned to be integrated with current version of bgpAND.

BGPMon Speaker Integration

BGPMon enabled monitors and speakers use a special format of exchanging of BGP messages. This is XML format for BGP messages (XFB). The current implementation is not integrated to directly connect to BGPMon monitors or speakers. Once integrated, they can carry out processing of updates from any of BGPMon enabled speakers. The same detection mechanisms can be then used for live detection of any of BGPMon enabled sites.

Sharing Mechanism of identified Malicious BGP Messages

Currently the system displays and logs alerts and anomalies it detects. This information is then maintained as local to the BGP updates processing machines. In an internetworking scenario, it is very effective to have a live sharing of any anomaly or any suspicious BGP update. This is helpful so that the anomaly information detected at one of the BGP nodes is shared among all other BGP neighbours and further on. Further on, standard protocols which provide rules on how to optimally share the information in a fast manner can be used.

VII. REFERENCES

[1]. Y. Rekhter, Ed., T. Li, Ed.and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", Standard Network Reference, Network Working Group, RFC, IETF RFC 4271, January 2006

[2] Rick Kuhn, Kotikalapudi Sriram, Doug Montgomery, Institute Publication Border Gateway Protocol Security; Recommendations of the National Institute of Standards and Technology, July 2007

[3] Kevin Butler, Toni R. Farley, Patrick McDaniel, Jennifer Rexford, "A Survey of BGP Security Issues and Solutions", Proceedings of the IEEE | Vol. 98, No. 1, January 2010

[4] BGP Tools, http://www.bgp4.as/tools

[5] BGPMOn - BGP Monitoring Tool, http://bgpmon. netsec.colostate.edu/

[6] RIPE NCC, RIS Raw BGP Data, http://www.ripe.net/ data-tools/stats/ris/ris-raw-data

[7] Team Cymru Bogon Service, http://www.team-cymru.org/Services/Bogons/

[8] Grep CIDR, http://www.pc-ools.net/ unix/grepcidr/

[9] K Sriram, Oliver Borchert, Okhee Kim, Patrick Gechmann, Doug Montgomery, "A comparative Analysis of BGP Anomaly Detection Robustness Algorithm", Proceedings of the Cybersecurity Applications and Technology Conference for Homeland Security (CATCH),Washington D.C., March 3-4, 2009, pp. 25-38.

◈◈◈