Design and Implementation of Router Intrusion Detection and Protection Systems

Nareshkumar D. Harale[1], Dr. B. B. Meshram[2]

[1]Department of Computer Science & Engineering, Sant Gadge Baba Amravati University, Amravati, Maharashtra, India
nareshkumar.d.harale[at]gmail.com

[2]Department of Computer Technology, VJTI, Mumbai, Maharashtra, India
bbmeshram[at]vjti.ac.in

Abstract: ***Attackers most of the time attack the gateway router's software and make the router to behave in a malicious way. This paper focuses on various attacks and how these attacks will be detected successfully by router log analysis. This paper also illustrates the proposed software architecture, its data structure and algorithms for the detection and protection of the router intrusions. So in this way in proposed system, logs are used to detect attack and also give defense mechanism for the detected attacks. The main aim of the work is to give details about how to communicate with the router from the program to turn ON only needed debugging; Collect router logs in a separate Syslog server; Segregate log files based on protocols; Analyze the log files to detect malicious attacks or misconfiguration; Communicate with the router again to apply appropriate access lists as defense mechanism and Save the logs report. Due to space constraints, some of the screen shots are shown. This work looks at some of the techniques to improve the accuracy of automated log analysis and make it a cost-effective tool for network management and improving network reliability.***

Keywords: System Integrity, Router Configuration, Network Security, Routing Systems, Routing Algorithms, Router Intrusion Detection Systems, Router Intrusion Protection Systems

1. Introduction

Cisco routers can provide an immense quantity of real time status information to support network management simply by enabling the system logging facility. Every state transition of every line can be recorded, along with call statistics, router configuration changes, software errors, environmental warnings, IOS reloads, and more. This level of detail can provide valuable insight into network operation that goes far beyond its normal use as a tool for resolving the cause of network failures. Many classes of problems, such as weak links which fail intermittently for brief periods, will show up in the syslog long before they are noticeable to users. Given the valuable information and operational insight available from the system logs, detailed analysis of syslog data should be a routine part of every network administrator's job. Yet in real life we find that this resource is frequently ignored simply because of the difficulty of dealing with the huge quantity of raw data. Any single physical event can easily generate a dozen or more log entries. Even a simple drop on a leased line will generate a link layer down notice and a physical layer down notice from the routers at each end of the line, followed by the up notices for both layers from both routers when the line returns to normal. If the link is frame relay, there will also be log entries for the associated DLCIs going inactive or deleted (and later returning to the active state). If dial backup is in place, there will also be physical and link up (and then down) notices for the backup medium and if the backup is ISDN, there will be ISDN connect and disconnect statistics. Add them all up and a single link hit can generate twenty or more log entries. If the network incident happens to involve a shared resource, such as a frame relay link supporting dozens of PVCs, the result can be page after page of log data simply because of one event. Even if there are no failures to report, routine testing of dial backup links will generate log entries which still must be scanned to ensure that they all really are test results and not a problem which just happened to occur during test periods. In a network with hundreds of routers, the logs can easily contain thousands of lines of entries for each day and if there are unstable lines, the thousands can expand to tens of thousands. As a result, many network administrators only look at the log files when debugging an already detected problem, and do not take advantage of the proactive network management possible with routine log analysis. This prevents them from detecting the early warning signs of impending problems such as brief outages, unexplained dial backup calls, and patterns in the failures occurring.

Sawmill is a Cisco Systems Router log analyzer (it also supports the 905 other log formats listed to the left). It can process log files in Cisco Systems Router format, and generate dynamic statistics from them, analyzing and reporting events. Sawmill can parse Cisco Systems Router logs, import them into a MySQL, Microsoft SQL Server, or Oracle database (or its own built-in database), aggregate them, and generate dynamically filtered reports, all through a web interface. Sawmill can perform Cisco Systems Router log analysis on any platform, including Window, Linux, FreeBSD, OpenBSD, Mac OS, Solaris, other UNIX, and others.

Log analysis (or system and network log analysis) is an art and science seeking to make sense out of computer-generated records (also called log or audit trail records). The process of creating such records is called data logging.

Typical reasons why people perform log analysis are:

1. Compliance with security policies.
2. Compliance with audit or regulation
3. System troubleshooting.

4. Forensics (during investigations or in response to subpoena).
5. Security incident response

Logs are emitted by network devices, operating systems, applications and all manner of intelligent or programmable device. A stream of messages in time-sequence often comprises a log. Logs may be directed to files and stored on disk, or directed as a network stream to a log collector. Log messages must usually be interpreted with respect to the internal state of its source (e.g., application) and announce security-relevant or operations-relevant events (e.g., a user login, or a systems error). Logs are often created by software developers to aid in the debugging of the operation of an application. The syntax and semantics of data within log messages are usually application or vendor-specific.

Tremendous amounts of useful operations data and warnings of pending failures are available in the router logs. The challenge is that as the network gets larger, so do the number of entries in the logs, which can quickly grow to unmanageable size. Automating the analysis of router logs is essential to allow using the router logs as a proactive network management tool. Many organizations fail to take full advantage of the available information because of the high initial cost of programming around the various inconsistencies in the way various events are reported, the frequency with which individual entries are delayed, duplicated or missing, and the need to customize the software to match their network configuration.

Protecting Routing Infrastructures from Denial of Service Attacks - To protect a network from routers that incorrectly drop packets and misroute packets, which can cause denial of service. Based on our detection-diagnosis-recovery approach, we propose protocols that detect and respond to those misbehaving routers. One of our techniques is called flow analysis, which monitors the transit traffic flowing in and out of a router to ensure they are of the same amount. Periodically, the neighbors of a router exchange their counts. A failed router that incorrectly drops transit packets can thus be detected by its neighbors. Subsequently, those neighbors will cease the neighbor relationship with the failed router [17][18][19]. We prove that our protocols have the following properties:

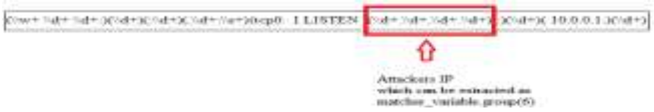
(1) A good router never incorrectly claims another router as a failed router;
(2) If a network has failed routers, one or more of them can be located;
(3) Failed routers will eventually be removed.

2. Literature Survey

Some of the methods /algorithms used for detection of router attacks are given below:

2.1 Port scan attack

An attacker can run a port scan on the router to see which ports are ON and which are not. This is mostly the first step for an attack. The regular expression pattern is as shown below to know the attackers IP:



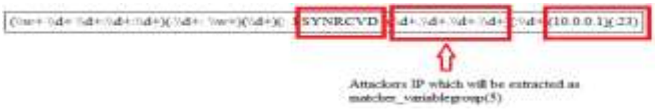
Defense mechanisms of port scan attack:

Configure the following IP ACL
Router>en
Router#conf t
Router(config)#ip access-list standard port_scan
Router(config-std-nacl)#deny attackers_ip 0.0.0.0
Router(config-std-nacl)#exit
Router(config)#interface f1/0
Router(config-if)#ip access-group port_scan in

2.2 Unknown login attack

Some attacker can somehow get router's access via telnet over the network and try to do malicious activity inside the outer. Telnet attempts on the router can be detected by log analysis.

Detection mechanism of unknown login attack: Use the following regular expression to detect unknown telnet attempt

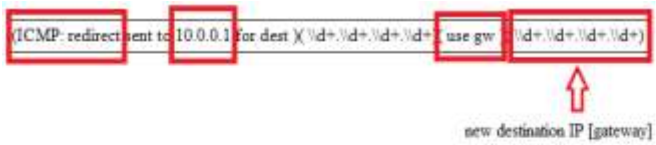


Defense mechanism of unknown login attack: Configure the following IP ACL
Router>en
Router#conf t
Router(config)#ip access-list standard unknwn_login
Router(config-std-nacl)#deny attacker's IP 0.0.0.0
Router(config-std-nacl)#exit
Router(config)#interface f1/0
Router(config-if)#ip access-group unknwn_login in

2.3 ICMP redirect attack

ICMP redirect packets are actually sent by routers to hosts if they find a better path to reach a destination. If a router receives a packet and forwards the packet to the same interface where it had received then an ICMP redirect can be sent. But an attacker can carry a man in middle attack using redirects.

Detection of ICMP redirect attack: Use the following regular expression to detect ICMP redirect attack



```
(ICMP: redirect sent to 10.0.0.1 for dest )( \\d+.\\d+.\\d+.\\d+ )( use gw )( \\d+.\\d+.\\d+.\\d+)
```

Defense mechanism of ICMP redirect attack; Remove the redirected route from routing table using following command

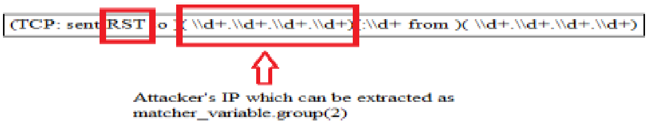
clear IP route destination IP [gateway]

The original (not redirected) route will now be learnt by the router through its running routing protocol.

2.4 BGP session termination attack

TCP reset attack is the attack in which a TCP connection is terminated by the attacker using a spoofed packet with the RST (reset) bit in the TCP packet set. This attack affects BGP protocol.

Detection of BGP session termination attack: Use the following regular expression to detect BGP session termination attack



```
(TCP: sent RST to )( \\d+.\\d+.\\d+.\\d+)(:\\d+ from )( \\d+.\\d+.\\d+.\\d+)
```

Attacker's IP which can be extracted as matcher_variable.group(2)

Defense mechanism of BGP session termination attack; Configure the following IP ACL

```
Router>en
Router#conf t
Router(config)#ip access-list standard unknwn_login
Router(config-std-nacl)#deny attacker's IP 0.0.0.0
Router(config-std-nacl)#exit
Router(config)#interface f1/0
Router(config-if)#ip access-group unknwn_login in
```

2.5 OSPF hello packet deletion attack

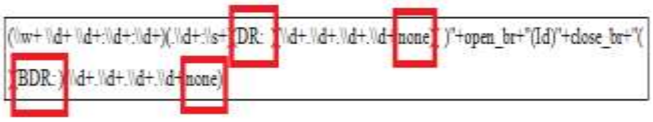
OSPF neighbors exchange hello packets every 10 seconds (Default hello timer is 10 seconds.) When 4 consecutive hello packets are missed by an OSPF process, then the OSPF process declares its neighbor as dead (Default dead timer is 40 seconds.) When a neighbor is dead, its entries will be flushed from the routing table. Attacker can purposely delete OSPF hello packets. After 4 consecutive message deletion, the neighbor ship will break. This is an attack which has caused the OSPF neighbor ship to break resulting into flushing of its OSPF entries.

Detection mechanism of OSPF hello packet deletion attack: The algorithm for detecting OSPF hello packet deletion attack is as given below:

2.6 OSPF DR BDR null attack

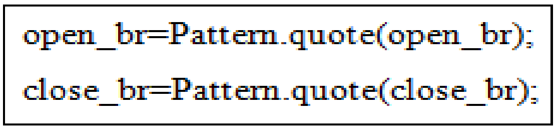
OSPF is a victim of DR, BDR null attack. OSPF elects DR, BDR on a multi access network. DR, BDR are elected based upon the priority and router ID (highest loopback address) sent in the hello message. After the election is done, the elected DR, BDR are sent in the hello message. An attacker can create a phantom router with highest priority and ID. Attacker will now set DR, BDR to null and then send that hello message. This will force reelection for DR, BDR and will elect the phantom router ad DR which will create undesirable effect.

Detection mechanism of OSPF DR BDR null attack: Use the following regular expression to detect DR BDR null attack



```
(\\w+ \\d+ \\d+:\\d+:\\d+)(.\\d+:\\s+(DR: )\\d+.\\d+.\\d+.\\d+|none) )"+open_br+"(Id)"+close_br+"(
(BDR: )\\d+.\\d+.\\d+.\\d+|none)
```

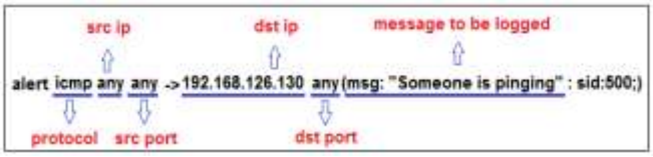
where



```
open_br=Pattern.quote(open_br);
close_br=Pattern.quote(close_br);
```

2.7 Router intrusion prevention by configuring acls on routers

Snort is an open source tool which works as an IDS (Intrusion Detection System).In this attack on router is prevented by the use of Snort to generate alerts for attacks and configure ACLs to defend. Snort can also be used to detect any intrusion in router and also took measures to take action for that intrusion using ACLs(Access Control Lists). Rules are written in Snort and they are matched against the packets. An example of a rule is as follows



```
alert icmp any any ->192.168.126.130 any(msg: "Someone is pinging" : sid:500;)
```

3. Figure 15: A Snort Rule

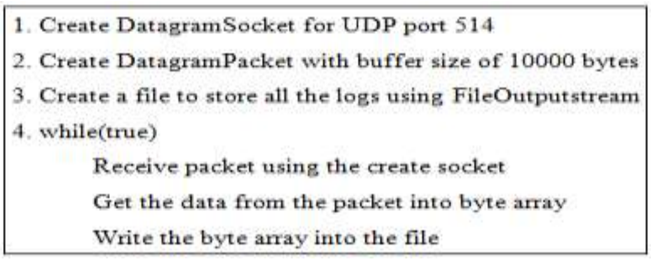
If a packet matches then messages are sent to the snort log. These logs now can be studied and appropriate access control lists can be generated for the router to curb the attack.

3.2 Algorithm Design

3.2.1. Syslog Algorithm

Input:
fos //FileOutputStream
Output:
realdata //byte array containing logs

Syslog Algorithm:

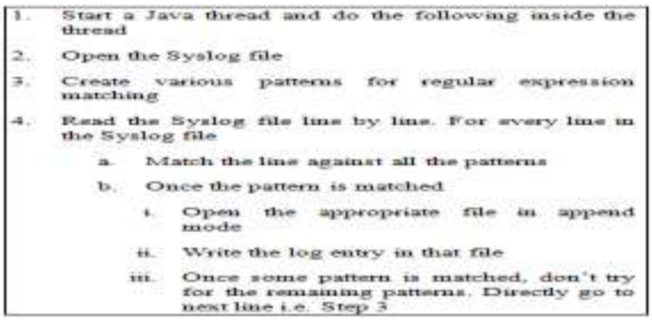


```
1. Create DatagramSocket for UDP port 514
2. Create DatagramPacket with buffer size of 10000 bytes
3. Create a file to store all the logs using FileOutputstream
4. while(true)
        Receive packet using the create socket
        Get the data from the packet into byte array
        Write the byte array into the file
```

3.2.2. File Processing Algorithm

Input:
in //RandomAccessFile pointing to syslog file
Output:
bgp //FileWriter pointing to bgp logs
tcp //FileWriter pointing to tcp logs
udp //FileWriter pointing to udp logs
icmp //FileWriter pointing to icmp logs
ospf //FileWriter pointing to ospf logs
rip //FileWriter pointing to rip logs

File Processing Algorithm:

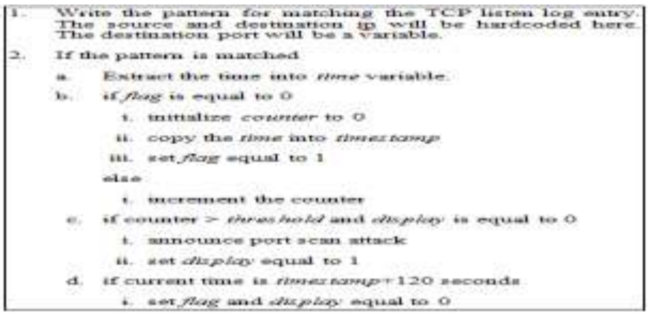


```
1.  Start a Java thread and do the following inside the thread
2.  Open the Syslog file
3.  Create various patterns for regular expression matching
4.  Read the Syslog file line by line. For every line in the Syslog file
      a.  Match the line against all the patterns
      b.  Once the pattern is matched
            i.   Open the appropriate file in append mode
            ii.  Write the log entry in that file
            iii. Once some pattern is matched, don't try for the remaining patterns. Directly go to next line i.e. Step 3
```

3.2.3. PORT_SCAN_DETECTION Algorithm

Input:
line //syslog line
Output:
alert //port scan attack alert

PORT_SCAN_DETECTION Algorithm

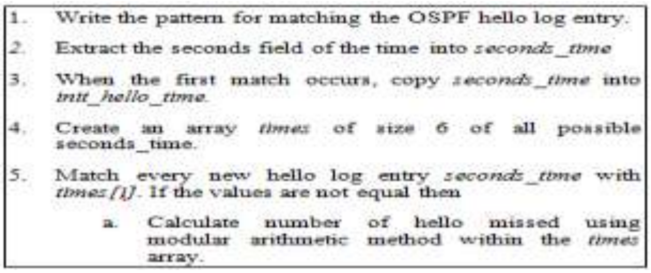


```
1.  Write the pattern for matching the TCP listen log entry. The source and destination ip will be hardcoded here. The destination port will be a variable.
2.  If the pattern is matched
      a.  Extract the time into time variable.
      b.  if flag is equal to 0
            i.   initialize counter to 0
            ii.  copy the time into timestamp
            iii. set flag equal to 1
          else
            i.   increment the counter
      c.  if counter > threshold and display is equal to 0
            i.   announce port scan attack
            ii.  set display equal to 1
      d.  if current time is timestamp+120 seconds
            i.   set flag and display equal to 0
```

3.2.4. HELLO_DELETION_DETECTION Algorithm

Input:
line //syslog line
Output:
alert //OSPF hello packet deletion attack alert

HELLO_DELETION_DETECTION Algorithm:

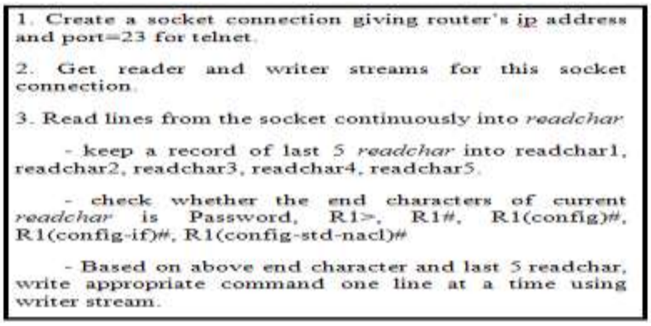


```
1.  Write the pattern for matching the OSPF hello log entry.
2.  Extract the seconds field of the time into seconds_time
3.  When the first match occurs, copy seconds_time into init_hello_time.
4.  Create an array times of size 6 of all possible seconds_time.
5.  Match every new hello log entry seconds_time with times[i]. If the values are not equal then
          a.  Calculate number of hello missed using modular arithmetic method within the times array.
```

3.2.5. COMMUNICATE_ROUTER Algorithm

Input:
host //router's IP
pass //router's password
Output:
Socket //connection to the router

COMMUNICATE_ROUTER Algorithm:



```
1. Create a socket connection giving router's ip address and port=23 for telnet.
2. Get reader and writer streams for this socket connection.
3. Read lines from the socket continuously into readchar
      - keep a record of last 5 readchar into readchar1, readchar2, readchar3, readchar4, readchar5.
      - check whether the end characters of current readchar is Password, R1>, R1#, R1(config)#, R1(config-if)#, R1(config-std-nacl)#
      - Based on above end character and last 5 readchar, write appropriate command one line at a time using writer stream.
```

4. Proposed System Implementation

4.1 Syslog Server

Routers have very less memory. But logs generated by the router are huge and will require large space to store. If the memory becomes full then logs will be overwritten. We don't want this to happen because some logs are very crucial. So now the logs are successfully stored on the Syslog server to provide a clean separation. The router console will not be interrupted with logs now. The logs can be viewed on the separate machine having Syslog server. Log analysis can now be done on this machine.

Levels of syslog server logging

There are eight levels of logging. When a particular level is set, then all logs up to and including that level are generated. The command to set log level is 'logging trap level'

The eight levels are as follows:
Emergency (severity 0)—The system is unusable
Alert (severity 1)—Immediate action is needed
Critical (severity 2)—Critical condition
Error (severity 3)—Error condition
Warning (severity 4)—Warning condition
Notification (severity 5)—Normal but significant condition
Informational (severity 6)—Informational message
Debugging (severity 7)—Debugging message [7]

Deployment of Syslog server

The machine on which the Syslog server is running is connected to the core switch. In this way, all the routers can direct their logs to the Syslog server. This gives the network administrator control over monitoring all the routers. Along with Syslog server, few other modules will be deployed on the system. These modules are shown in Fig. in purple. The black box is the system having IP address 10.0.0.100. The router interface and this system should be in the same network in order to communicate.

File Processing

Above Syslog server program will write all the log messages received on UDP port 514 into a single file on the system. This process will be happening continuously. A Java thread will be launched which will do the task of the Syslog server [11]. So the Syslog file is continuously updating. This continuously updating file will be read by a file processing module. The file processing module is launched in a separate java thread. This module is continuously running. It will parse every line from the Syslog file and sort them into different files protocol wise. As one can see, every entry has month, date, time, protocol, log message. We will extract the protocol field through regular expression matching. Based on the protocol, the log entries will be written to different file as shown in Fig. Fig shows that 7 different files were created: TCP, UDP, ICMP, RIP, OSPF, BGP, MPLS. This is needed because we want to detect attacks protocol wise. For example, OSPF attacks will be different from TCP and so on.

Algorithm for file processing

Java's regular expression matching is very strong [11]. It contains an entire package called java.util.regex dedicated to regular expression matching. The main classes used are Pattern and Matcher. Now the entries from the single Syslog file are separated and written into separate files according to protocol. This step is very important. This will be extremely useful for attacks detection where we will analyze the flow of activities inside a protocol. For example, we can monitor all OSPF activities easily from the OSPF log file. This separation will be useful. At the end of analysis, the administrator can carry these files which are sorted on protocol basis for report submission.

4.2 Router attacks Detection and Defense

This section will focus on various attacks and how these attacks will be detected successfully by the system. This is the main aim for router log analysis: Attacks detection. Some attacks can be detected by just analyzing one log entry such as BGP's session termination attack or ICMP redirect attack. On the other hand, some attacks require analyzing more than 1 line before actually declaring that an attack has happened. Some of the algorithms used for detection of router attacks are given below:

4.2.1 Port scan attack

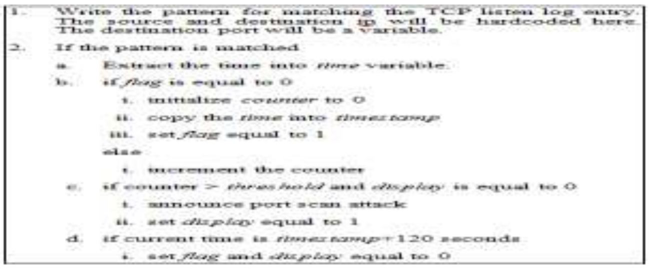
An attacker can run a port scan on the router to see which ports are ON and which are not. This is mostly the first step for an attack. Loopholes can found after a port scan.

Detection mechanisms of port scan attack

The source and destination IP address will be same everywhere. The destination ports will be different. A threshold can be maintained by our algorithm which will tell how many packets to scan before announcing a port scan attack. This threshold might be 10, 15 as stated by the network administrator. The algorithm for detecting port scan attack is as given in Fig.

Input:
line //syslog line
Output:
alert //port scan attack alert

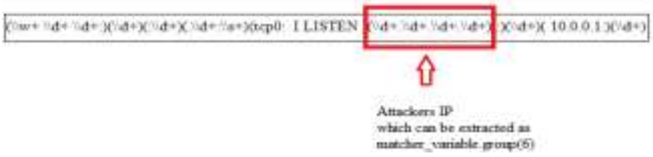
PORT_SCAN_DETECTION Algorithm:



```
1.  Write the pattern for matching the TCP listen log entry.
    The source and destination ip will be hardcoded here.
    The destination port will be a variable.
2.  If the pattern is matched
    a.  Extract the time into time variable.
    b.  if flag is equal to 0
            i.   initialize counter to 0
            ii.  copy the time into timestamp
            iii. set flag equal to 1
        else
            i.   increment the counter
    c.  if counter > threshold and display is equal to 0
            i.   announce port scan attack
            ii.  set display equal to 1
    d.  if current time is timestamp+120 seconds
            i.   set flag and display equal to 0
```

The display variable is used to prevent the announcement of attack more than once for the same attack. The flag variable is used to start a fresh new scan for the attack. Distributed Denial of Service (DDoS) attack on the router can also be detected in the similar fashion.

The regular expression pattern is as shown below:



```
(\\w+ \\d+ \\d+ )(\\d+)(\\d+)(\\d+\\s+)(tcp0: I LISTEN (\\d+ \\d+ \\d+ \\d+) )(\\d+)( 10.0.0.1 )(\\d+)
```

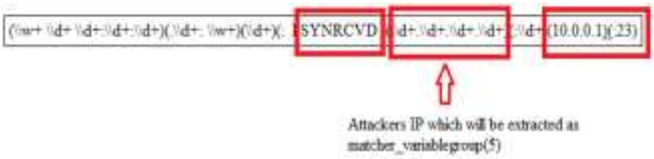
Defense mechanism of port scans attack
Configure the following IP ACL
Router>en
Router#conf t
Router(config)#ip access-list standard port_scan
Router(config-std-nacl)#deny attackers_ip 0.0.0.0
Router(config-std-nacl)#exit
Router(config)#interface f1/0
Router(config-if)#ip access-group port_scan in

4.2.2 Unknown login attack

Some attacker can somehow get router's access via telnet over the network and try to do malicious activity inside the outer. Telnet attempts on the router can be detected by log analysis.

Detection mechanism of unknown login attack

Use the following regular expression to detect unknown telnet attempt



```
(\\w+ \\d+ \\d+:\\d+:\\d+)(.\\d+: \\w+)(\\d+)(. SYNRCVD (\\d+.\\d+.\\d+.\\d+):\\d+(10.0.0.1)(:23)
```

Defense mechanism of unknown login attack

Configure the following IP ACL

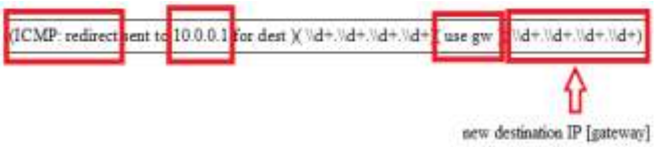
```
Router>en
Router#conf t
Router(config)#ip access-list standard unknwn_login
Router(config-std-nacl)#deny attacker’s IP 0.0.0.0
Router(config-std-nacl)#exit
Router(config)#interface f1/0
Router(config-if)#ip access-group unknwn_login in
```

4.2.3 ICMP redirect attack

ICMP redirect packets are actually sent by routers to hosts if they find a better path to reach a destination. If a router receives a packet and forwards the packet to the same interface where it had received then an ICMP redirect can be sent. But an attacker can carry a man in middle attack using redirects.

Detection of ICMP redirect attack

Use the following regular expression to detect ICMP redirect attack



```
(ICMP: redirect sent to 10.0.0.1 for dest )( \\d+.\\d+.\\d+.\\d+)( use gw )( \\d+.\\d+.\\d+.\\d+)
```

Defense mechanisms of ICMP redirect attack

Remove the redirected route from routing table using following command

clear IP route destination IP [gateway]

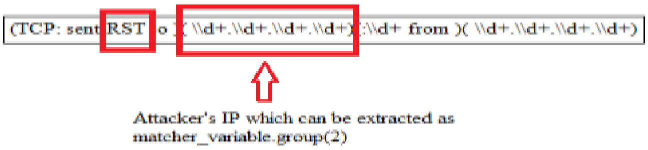
The original (not redirected) route will now be learnt by the router through its running routing protocol.

4.2.4 BGP session termination attack

Tcp reset attack is the attack in which a TCP connection is terminated by the attacker using a spoofed packet with the RST (reset) bit in the TCP packet set. This attack affects BGP protocol.

Detection of BGP session termination attack

Use the following regular expression to detect BGP session termination attack



```
(TCP: sent RST to )( \\d+.\\d+.\\d+.\\d+)(:\\d+ from )( \\d+.\\d+.\\d+.\\d+)
```

Attacker's IP which can be extracted as
matcher_variable.group(2)

Defense mechanism of BGP session termination attack

Configure the following IP ACL

```
Router>en
Router#conf t
Router(config)#ip access-list standard unknwn_login
Router(config-std-nacl)#deny attacker’s IP 0.0.0.0
Router(config-std-nacl)#exit
Router(config)#interface f1/0
Router(config-if)#ip access-group unknwn_login in
```

4.2.5 OSPF hello packet deletion attack

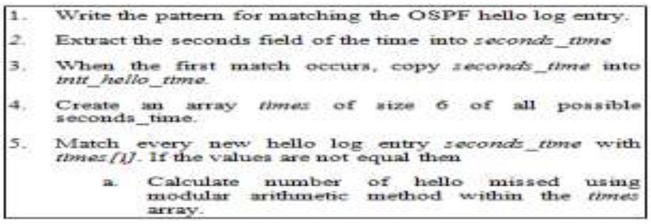
OSPF neighbors exchange hello packets every 10 seconds (Default hello timer is 10 seconds.) When 4 consecutive hello packets are missed by an OSPF process, then the OSPF process declares its neighbor as dead (Default dead timer is 40 seconds.) When a neighbor is dead, its entries will be flushed from the routing table. Attacker can purposely delete OSPF hello packets. After 4 consecutive message deletions, the neighborship will break. This is an attack which has caused the OSPF neighborship to break resulting into flushing of its OSPF entries.

Detection mechanism of OSPF hello packet deletion attack

The algorithm for detecting OSPF hello packet deletion attack is as given in Fig.

Input:
line //syslog line
Output:
alert //OSPF hello packet deletion attack alert

HELLO_DELETION_DETECTION Algorithm:



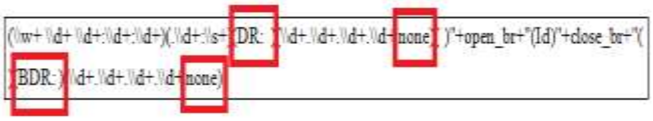
```
1. Write the pattern for matching the OSPF hello log entry.
2. Extract the seconds field of the time into seconds_time
3. When the first match occurs, copy seconds_time into init_hello_time.
4. Create an array times of size 6 of all possible seconds_time.
5. Match every new hello log entry seconds_time with times[i]. If the values are not equal then
      a. Calculate number of hello missed using modular arithmetic method within the times array.
```

4.2.6 OSPF DR BDR null attack

OSPF is a victim of DR, BDR null attack. OSPF elects DR, BDR on a multi access network. DR, BDR are elected based upon the priority and router ID (highest loopback address) sent in the hello message. After the election is done, the elected DR, BDR are sent in the hello message. An attacker can create a phantom router with highest priority and ID. Attacker will now set DR, BDR to null and then send that hello message. This will force reelection for DR, BDR and will elect the phantom router ad DR which will create undesirable effect.

Detection mechanism of OSPF DR BDR null attack

Use the following regular expression to detect DR BDR null attack



```
(\\w+ \\d+ \\d+:\\d+:\\d+)(.\\d+:\\s+)(DR: )(\\d+.\\d+.\\d+.\\d+|none)( )"+open_br+"(Id)"+close_br+"(
)(BDR: )(\\d+.\\d+.\\d+.\\d+|none)
```

Where

```
open_br=Pattern.quote(open_br);
close_br=Pattern.quote(close_br);
```

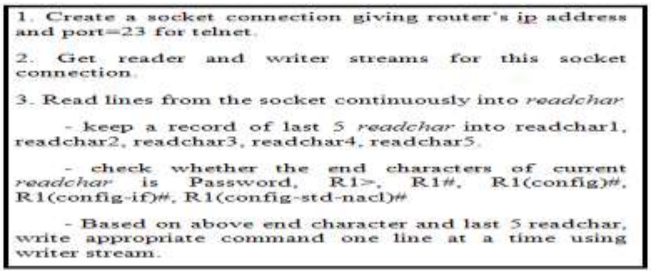
4.5 Router Protection Mechanism

4.5.2 Algorithm for communication with router

Communication with the router will be done using protocol Telnet. The algorithm for this module is given in Fig.

Input:
host //router's IP
pass //router's password
Output:
Socket //connection to the router

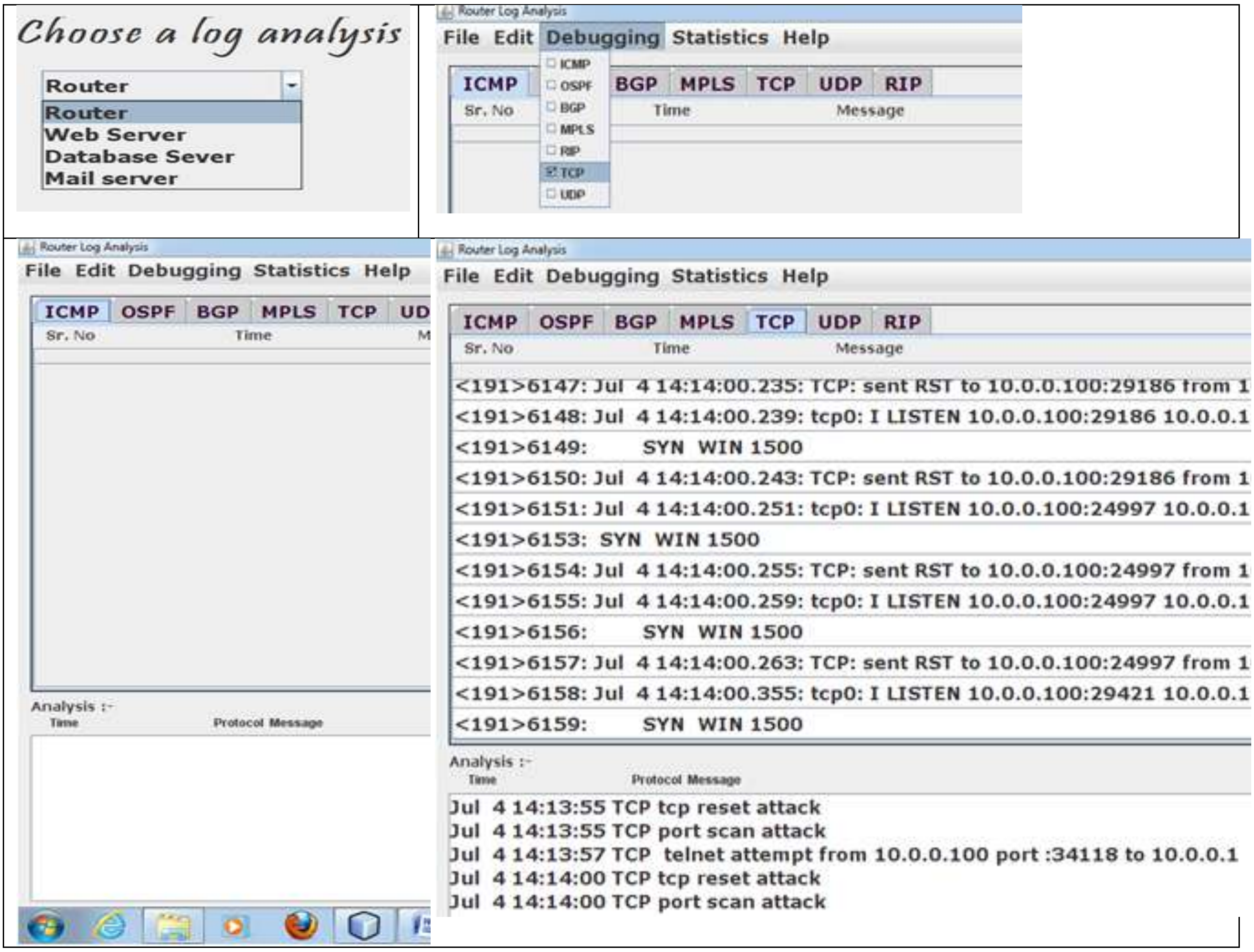
COMMUNICATE_ROUTER Algorithm:



```
1. Create a socket connection giving router's ip address and port=23 for telnet.
2. Get reader and writer streams for this socket connection.
3. Read lines from the socket continuously into readchar
    - keep a record of last 5 readchar into readchar1, readchar2, readchar3, readchar4, readchar5.
    - check whether the end characters of current readchar is Password, R1>, R1#, R1(config)#, R1(config-if)#, R1(config-std-nacl)#
    - Based on above end character and last 5 readchar, write appropriate command one line at a time using writer stream.
```

Screenshots of output

Following are the detailed screenshots:



```
<191>6147: Jul  4 14:14:00.235: TCP: sent RST to 10.0.0.100:29186 from 1
<191>6148: Jul  4 14:14:00.239: tcp0: I LISTEN 10.0.0.100:29186 10.0.0.1
<191>6149:        SYN  WIN 1500
<191>6150: Jul  4 14:14:00.243: TCP: sent RST to 10.0.0.100:29186 from 1
<191>6151: Jul  4 14:14:00.251: tcp0: I LISTEN 10.0.0.100:24997 10.0.0.1
<191>6153:  SYN  WIN 1500
<191>6154: Jul  4 14:14:00.255: TCP: sent RST to 10.0.0.100:24997 from 1
<191>6155: Jul  4 14:14:00.259: tcp0: I LISTEN 10.0.0.100:24997 10.0.0.1
<191>6156:        SYN  WIN 1500
<191>6157: Jul  4 14:14:00.263: TCP: sent RST to 10.0.0.100:24997 from 1
<191>6158: Jul  4 14:14:00.355: tcp0: I LISTEN 10.0.0.100:29421 10.0.0.1
<191>6159:        SYN  WIN 1500
```

```
Jul  4 14:13:55 TCP tcp reset attack
Jul  4 14:13:55 TCP port scan attack
Jul  4 14:13:57 TCP  telnet attempt from 10.0.0.100 port :34118 to 10.0.0.1
Jul  4 14:14:00 TCP tcp reset attack
Jul  4 14:14:00 TCP port scan attack
```

Observation from above screen shot is as below:

- All logs are not mixed. ICMP logs shown in separate tab
- After a threshold, an attack alert is shown below
- An ACL is automatically configured on the router after the attack as defense mechanism.
- An attack can happen in absence of the network admin. So the ACL is automatically configured to defend the router.

Due to space constraints the screen shots of all the attacks on ICMP, OSPF, BGP, UDP etc. is not simulated. Only attack on TCP is shown herewith.

5. Conclusion and Future Research Directions

Attacks detection without log analysis has following problems

- Several logs were generated having the same time
- All logs are intermixed
- It's impossible for the network admin who is seeing these logs to detect a DDOS attack. And maybe, by the time he has detected the attack; a huge loss had already occurred.
- Moreover there can be situations when the admin is not present there to see these logs and DDOS attack can happen in his absence.

A complete network monitoring tool which will tell about all the working protocols on the routers, the connectivity between the routers and the malicious activities happening on the routers is not developed till date. This work has created a base and has provided the first step to do so. It has found a way to dump log to separate machine and also sort them. It can fire commands on the router which means it can easily obtain the output of 'sh run' from routers. It knows how to extract information from the log to detect an attack. It has also defended routers by configuring ACLs.

References

[1] Charalampos Patrikakis, Michalis Masikos, and Olga Zourarak, "Distributed Denial of Service Attacks", The Internet Protocol Journal - Volume 7, Number 4, 2004.
[2] "ICMP Attacks Illustrated, SANS Institute InfoSec Reading Room". [Online]. Available: http://www.sans.org/reading_room/whitepapers/threats/icmp-attacks-illustrated_477
[3] "Routing protocol". [Online]. Available: http://en.wikipedia.org/wiki/Routing_protocol.
[4] Kotikalapudi Sriram, Doug Montgomery, Oliver Borchert, Okhee Kim and D. Richard Kuhn, "Study of BGP Peering Session Attacks and Their Impacts on Routing Performance", IEEE Journal On Selected Areas In Communications: Special Issue On High-Speed Network Security, Vol. 24, No. 10, October 2006.
[5] "Michael Sudkovitch, David I. Roitman, , OSPF Security project book". [Online]. Available: http://webcourse.cs.technion.ac.il/236349/Spring2013/ho/WCFiles/2009-2-ospf-report.pdf
[6] "Communicate with router".: http:// www. omnisecu. com/ cisco- certified –network –associate - ccna/ how- to- communicate- with- a- router.htm
[7] "Karsten Iwen, Logging in Cisco IOS".: http://security-planet.de/wp-content/uploads/2008/12/logging-ios.pdf
[8] "Anand Deveriya, An overview of the Syslog protocol, Cisco Press". http://www.ciscopress.com/articles/article.asp?p=426638
[9] "Cisco IOS Debug Command Reference". [Online]. Available: http://www.cisco.com/en/US/docs/ios-xml/ios/debug/command/s1/db-s1-cr-book.pdf
[10] "Sean Wilkins, Basic access lists configuration for cisco devices, Cisco Press".: http://www.ciscopress.com/articles/article.asp?p=1697887
[11] http://www.iss.net/security_center/advice/Intrusions/2000012/default.htm
[12] Danai Chasaki and Tilman Wolf, "Attacks and Defenses In The Data Plane Of Networks", IEEE Transactions On Dependable And Secure Computing (Tdsc), 2012.
[13] Kirk A.Radley, Steven Cheung, Nicholas Puketza, Biswanath Mukherjee, and Ronald A. Olsson, "Detecting Disruptive Routers: A Distributed Network Monitoring Approach."
[14] Vrizlynn L. L. Thing, Morris Sloman, Naranker Dulay, "Locating Network Domain Entry And Exit Point/Path For Ddos Attack Traffic."
[15] Muhammad Naveed, Shams un Nihar, Mohammad Inayatullah Babar, "Network Intrusion Prevention By Configuring Acls On The Routers, Based On Snort Ids Alerts", Emerging Technologies (ICET), 2010.
[16] Eric Y. K. Chan, H. W. Chan, K. M. Chan, P. S. Chan, Samuel T. Chanson, M. H. Cheung, C. F. Chong, K. P. Chow, Albert K. T. Hui, Lucas C. K. Hui, S. K. Ip, C. K. Lam, W. C. Lau, K. H. Pun, Y. F. Tsang, W. W. Tsang, C. W. Tso, D. Y. Yeung, S. M. Yiu, K. Y. Yu, and WeihuaJu, "Intrusion Detection Routers: Design, Implementation and Evaluation Using an Experimental Testbed", " IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS", VOL. 24, NO. 10, OCTOBER 2006.
[17] Steven Cheung, "An Efficient Message Authentication Scheme for Link State Routing". Proceedings of the 13th Annual Computer Security Applications Conference.
[18] Steven Cheung and Karl Levitt, "Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection". Proc. New Security Paradigms Workshop, 1997.
[19] Kirk A. Bradley, Steven Cheung, Nick Puketza, Biswanath Mukherjee, and Ronald A. Olsson, "Detecting Disruptive Routers: A Distributed Network Monitoring Approach". Proceedings of the 1998 IEEE Symposium on Security and Privacy