IRACST – International Journal of Computer Networks and Wireless Communications (IJCNWC), ISSN: 2250-3501
Vol.3, No3, June 2013

Router attacks detection through log analysis and defense mechanism

Saili R. Waichal
Dept. of Computer Engineering and IT
Veermata Jijabai Technological Institute
Mumbai, India

Gopal J. Sonune
Dept. of Computer Engineering and IT
Veermata Jijabai Technological Institute
Mumbai, India

B.B.Meshram
Head & Asst Professor, Dept. of Computer Engineering and IT
Veermata Jijabai Technological Institute
Mumbai, India

***Abstract*— Router is one of the most important devices in networking. Its function is to route packets between different networks. Thus all the interfaces of the router belong to different networks. Any campus-wide network architecture may contain many routers or multilayer switches for inter-department communication. It plays a very important role in communication in the campus. So security of router is inevitable. A network administrator has to monitor router for its security. Routers have an excellent inbuilt facility of logging which can be enabled. Using logs is the best way to monitor any system: Database server, Mail server, Web server or router etc. Proper analysis of such logs continuously can help us detect many serious attacks. This paper focuses on using router logs, directed to a separate Syslog server, for attacks detection. It also gives a method to configure appropriate access control lists on the router as defense mechanism.**

Keywords- Access Control lists; Router Attacks; Router Debugging; Router Logs; Syslog.

I. INTRODUCTION

A router can be attacked in many ways by a hacker. Most commonly, Distributed Denial of Service Attack can hit the router. ARP poisoning, Smurf attack, Ping of Death are the examples of DDoS attack on router that can collapse the entire network. Routing packets between networks is the main aim of a router. Many Man in the Middle (MiM) attacks can be caused which will direct the traffic to an attacker instead of sending it to the legitimate router. ICMP redirect attack is one such attack. Routers contain routing entries which decide the path selection. Routing entry poisoning is probably the most obvious attack which will make the router to behave in an anomalous way. [1][2]

Routers communicate with other routers using a routing protocol. Some of the major routing protocols are as given below

- OSPF and IS-IS are link state routing protocols which do Interior gateway routing.
- IGRP and EIGRP are distance vector routing protocols which also do Interior gateway routing.
- BGP is the protocol which does Exterior gateway routing.

Routing protocols also can be targeted for an attack. Session termination is a serious problem with hugely spanned and sensitive protocols like BGP. OSPF is the victim of many attacks like Hello message deletion, Max sequence attack, External routes attacks etc. [3]

Any organization's major concern is to defend their routers from such attacks. The network administrator of the organization can implement a Firewall which can filter incoming and outgoing traffic. Encryption techniques can be used to prevent certain attacks. Software monitoring through Snort can be done. Apart from all above techniques, proper log analysis of logs can give a lot of insight in attacks detection. Following section will propose a system which will enable and use logs for router attacks detection. After attacks are detected, a mechanism is also stated to configure appropriate access list on the router as defense mechanism.

II. PROPOSED SYSTEM

A. Router Configuration

One can communicate with a router using any of the following ways [4]

- Using Console port: Router has a console port. A console cable can connect your machine to the router. A terminal emulator program like hyperterminal or putty is required on your machine. This will give you access to the router prompt.
- Using Aux port: By using a remote computer through a modem that calls another modem connected to the router with a cable using the Auxiliary Port on the router.
- Using protocols: Protocols like Telnet, ssh, http, https can be used to connect to the router over a network.

In our system, we are going to use telnet connection to communicate with the router. Line vty on router needs to

configured to achieve telnet connection. Following commands should be configured on the router for using vty line.

Router>en

Router#conf t

Router(config)#line vty 0

Router(config-line)#password cisco

Router(config-line)#exit

Router(config)#exit

A router can log information to console, host (syslog), snmp, buffer, monitor (ssh, telnet). Logging to console can be turned OFF if it is ON [6]. In our system, we want to direct logs to another host (Syslog server). We set the logging to the highest i.e. 7. Following commands will achieve this.

Router>en

Router#conf t

Router(config)#no logging console

Router(config)#logging 10.0.0.100

Router(config)#logging trap 7

Router(config)#exit

B. 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. The algorithm for implementing Syslog server is given in Fig. 1

1. Open UDP port 514 on the system with ip address 10.0.0.100.
2. Receive packets from this port into a buffer.
3. Extract the data portion from this packet
4. Write the data portion in a file on the system

Figure 1. Algorithm for Syslog Server

So now the logs are successfully stored on the Syslog server. This also provides 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: There are eight levels of logging. When a particular level is set, then all logs upto and including that level are generated. The command to set log level is 'logging trap *level*' [5]

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

Syslog server will be continuously running. So thread programming of java is used. The most appropriate place for deploying the Syslog server and the log analyzing program would be as shown in Fig. 2

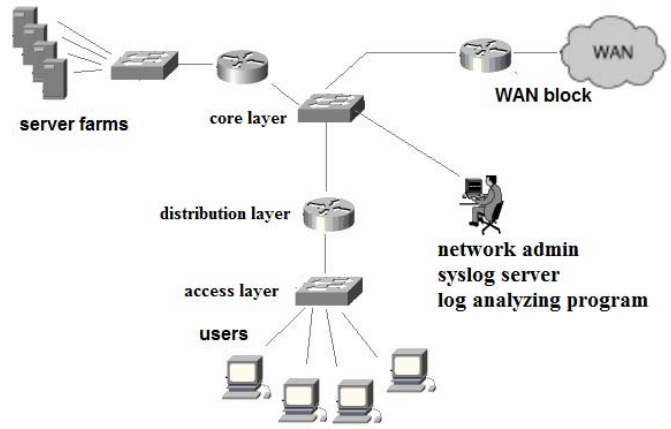

Figure 2. Deployment of Syslog Server in Campus Network

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.

Linux has an inbuilt Syslog daemon. This inbuilt Syslog server needs to be enabled to use it as a network based Syslog server through the /etc/Syslog.conf file. It also needs to be enabled to receive logs from the network. But this Syslog daemon has not been updated to add more useful features. Windows doesn't have any built in Syslog facility. You can download Syslog application for windows from the internet. Syslog-ng is the latest with some extra features also [5]. But for our model, the Syslog server is written in java using the algorithm given in Fig.1. It satisfies our needs properly. So we are content with it.

Along with Syslog server, few other modules will be deployed on the system. These modules are shown in Fig.3 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.

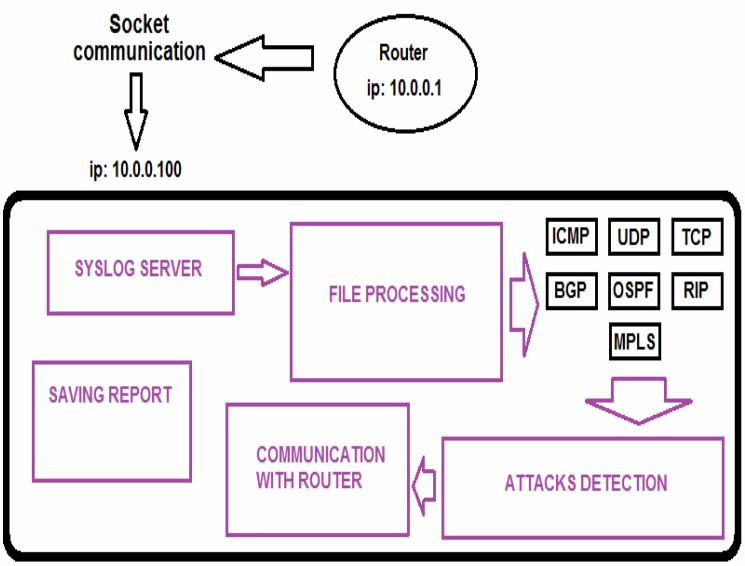

Figure 3. Components of the log analyzing system

C. *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. 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. Some examples of Cisco router log entries in the Syslog file are shown in Fig. 4

May 14 15:32:18.835: OSPF: Interface FastEthernet1/0 going Up

May 13 22:05:54.839: ICMP: echo reply sent, src 10.0.0.1, dst 10.0.0.100

May 14 11:11:48.923: RIP: sending v1 flash update to 255.255.255.255 via FastEthernet1/0 (10.0.0.1)

May 13 22:48:40.043: UDP: rcvd src=10.0.0.100(137), dst=10.255.255.255(137), length=58

Figure 4. Sample Cisco log entries

As one can see, every entry has month, date, time, protocol, log message. We will extract the protocol field through regular expression matching in java. Based on the protocol, the log entries will be written to different file as shown in Fig 3. Fig 3. shows that 7 different files were created: TCP, UDP, ICMP, RIP, OSPF, BGP, MPLS. The default syslogd daemon of Linux operating system doesn't do this. So developing our own Syslog program was the better idea because we want to detect attacks protocol wise. For example, OSPF attacks will be different from TCP and so on.

Java's regular expression matching is very strong. It contains an entire package called java.util.regex dedicated to regular expression matching. The main classes used are Pattern and Matcher. The algorithm that we will be using for regular expression is given in Fig. 5

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

Figure 5. Algorithm for separation of files using regular expression matching

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.

D. *Attacks Detection*

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:

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 neighborship will break. This is an attack which has caused the OSPF neighborship to break resulting into flushing of its OSPF entries [3]. To generate log entry for each OSPF hello packet sent or received, the 'debug ip ospf events' debugging command can be used. This will enable logging for OSPF events. The log entry for the hello packet is as shown in Fig. 6

```
May 13 22:40:09.455: OSPF: Send hello to 224.0.0.5 area
0 on FastEthernet1/0 from 10.0.0.1
```

Figure 6: Log entry for OSPF hello packets

The algorithm for detecting OSPF hello packet deletion attack is as given in Fig. 7

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.

Figure 7: Algorithm for detecting OSPF hello packets deletion attack

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. When a port scan happens on a router, the log entries which are generated are shown in Fig. 8

```
May 14 10:58:25.627: tcp0: I LISTEN 10.0.0.100:1495
10.0.0.1:1 seq 1473192529

May 14 10:58:25.791: tcp0: I LISTEN 10.0.0.100:1496
10.0.0.1:2 seq 4232257361

May 14 10:58:25.883: tcp0: I LISTEN 10.0.0.100:1497
10.0.0.1:3 seq 452016969

May 14 10:58:26.007: tcp0: I LISTEN 10.0.0.100:1498
10.0.0.1:4 seq 3164921931

May 14 10:58:26.087: tcp0: I LISTEN 10.0.0.100:1499
10.0.0.1:5 seq 2912730653
```

Figure 8: Log entries during 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 givenn in Fig. 9

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

Figure 9: Algorithm for detecting port scan attack

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.

Similarly, algorithms are written to detect many other attacks as well. The log entry after an ICMP redirect attack, BGP session termination attack is given in Fig. 10 respectively:

```
May 13 23:25:07.938: ICMP: redirect sent to 10.0.0.1 for
dest 172.16.1.111 use gw 172.21.80.23

May 13 22:48:42.515: TCP: sent RST to 10.0.0.100:100
from 10.0.0.1:
```

Figure 10: Log entries during ICMP redirect attack and BGP session termination attack respectively

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. The log entry after telnet attempt is shown in Fig. 11

```
May 13 22:15:15.915: tcp2: I ESTAB 10.0.0.100:2466
10.0.0.1:23 seq 3097509464 ACK 2052496558 WIN 17440

May 13 22:15:15.931: tcp2: O ESTAB 10.0.0.100:2466
10.0.0.1:23 seq 2052496577 DATA 31 ACK 3097509464
PSH WIN 4089

May 13 22:15:15.943: tcp2: I ESTAB 10.0.0.100:2466
10.0.0.1:23 seq 3097509464 ACK 2052496560 WIN
17438

May 13 22:15:15.951: tcp2: I ESTAB 10.0.0.100:2466
10.0.0.1:23 seq 3097509464 ACK 2052496562 WIN
17436
```

Figure 11: Log entries after a telnet attempt into the router

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 [3]. The raw log after a DR, BDR null attack is shown in Fig. 12.

```
May 13 20:02:36.447: OSPF: Interface FastEthernet1/0 going Up
May 13 20:02:36.447: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet1/0 from 10.0.0.1
May 13 20:02:46.451: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet1/0 from 10.0.0.1
May 13 20:02:54.699: %SYS-5-CONFIG_I: Configured from console by console
May 13 20:02:56.455: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet1/0 from 10.0.0.1
May 13 20:03:00.095: ICMP: echo reply rcvd, src 10.0.0.100, dst 10.0.0.1
May 13 20:03:00.163: ICMP: echo reply rcvd, src 10.0.0.100, dst 10.0.0.1
May 13 20:03:00.163: TCP: sent RST to 10.0.0.100:10230 from 10.0.0.1:23
May 13 22:49:44.134:      DR: none   BDR: none
May 13 22:49:44.122: OSPF: DR/BDR election on FastEthernet1/0
May 13 22:49:44.126: OSPF: Elect BDR 10.0.0.1
May 13 22:49:44.126: OSPF: Elect DR 200.0.0.1
May 13 22:49:44.134:       DR: 200.0.0.1 (Id)   BDR: 10.0.0.1
```

Figure 12. Raw log after a DR, BDR null attack

The screenshots in Fig. 13 and Fig. 14 will give an idea about the purpose of this paper.

ICMP redirect attack detection without log analysis

```
May 21 20:02:36.447: OSPF: Interface FastEthernet1/0 going Up
May 21 20:02:36.447: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet1/0 from 10.0.0.1
May 21 20:02:46.451: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet1/0 from 10.0.0.1
May 21 20:02:54.699: %SYS-5-CONFIG_I: Configured from console by console
May 21 20:02:56.455: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet1/0 from 10.0.0.1
May 21 20:03:00.095: ICMP: echo reply rcvd, src 10.0.0.100, dst 10.0.0.1
May 21 20:03:00.163: ICMP: echo reply rcvd, src 10.0.0.100, dst 10.0.0.1
May 21 20:03:00.235: ICMP: echo reply rcvd, src 10.0.0.100, dst 10.0.0.1
May 21 20:03:00.423: ICMP: echo reply rcvd, src 10.0.0.100, dst 10.0.0.1
May 21 20:03:00.487: ICMP: echo reply rcvd, src 10.0.0.100, dst 10.0.0.1
May 21 20:03:00.490: ICMP: redirect sent to 10.0.0.1 for dest 172.16.1.111 use gw 172.21.80.23
May 21 20:03:06.455: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet1/0 from 10.0.0.1
May 21 20:03:16.447: OSPF: end of Wait on interface FastEthernet1/0
May 21 20:03:16.447: OSPF: DR/BDR election on FastEthernet1/0
May 21 20:03:16.451: OSPF: Elect BDR 10.0.0.1
May 21 20:03:16.451: OSPF: Elect DR 10.0.0.1
May 21 20:03:16.455: OSPF: Elect BDR 0.0.0.0
May 21 20:03:16.455: OSPF: Elect DR 10.0.0.1
May 21 20:03:16.455:       DR: 10.0.0.1 (Id)   BDR: none
May 21 20:03:16.459: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet1/0 from 10.0.0.1
May 21 20:03:16.955: OSPF: No full nbrs to build Net Lsa for interface FastEthernet1/0
May 21 20:03:26.463: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet1/0 from 10.0.0.1
May 21 20:03:36.467: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet1/0 from 10.0.0.1
May 21 20:03:46.471: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet1/0 from 10.0.0.1
```

Figure 13. Attacks detection without log analysis

ICMP redirect attack detection with log analysis

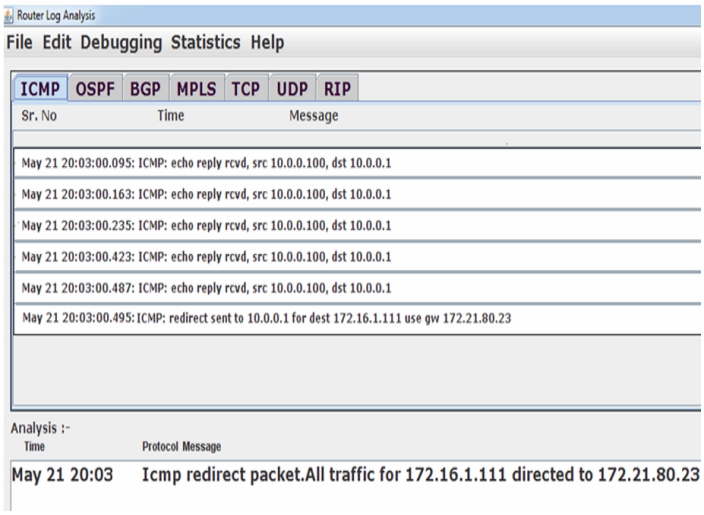

Figure 14. Attacks detection with log analysis program

E. Communication with the router

There are 2 main reasons for having this module.

1. Turning ON all possible logging is not recommended on the routers because it will slow down its operation. So if in case, the network administrator wants to monitor only routing protocols for some time, then the software will give a menu item to turn ON debugging only for OSPF, BGP etc. When you select this option, appropriate debugging command will be fired on the router. So for this, communication with router will be required. Examples of debugging command are ‘debug ip ospf events’ for major OSPF debugging; ‘debug ip tcp packet’ for tcp protocol; ‘debug ip icmp’ for ICMP packets etc [7].

2. After an alert for some attack is raised, its defense mechanism must be started. The network administrator will usually take all efforts to prevent any attack happening on router in the first place. But unfortunately if any attack occurs then its defense mechanism should be initiated. Now the network administrator cant be available all the time to monitor the router activities. Ideally the software itself should take steps to defend the router from the attack as soon as possible. So again here ‘communication with router’ module will be required to configure ACLs on the router to defend the attack. ACLs manage IP traffic as network access grows; control flow of data entering/ exiting routers port; filter data packet on the basis of L3 and above information. An example of ACL being configured on a router is as shown below:

```
Router>en
Router#conf t
Router(config)#ip access-list standard abc
Router(config-std-nacl)#deny 10.0.0.100 0.0.0.0
Router(config-std-nacl)#exit
Router(config)#interface f1/0
Router(config-if)#ip access-group abc in
```

Communication with the router will be done using protocol Telnet through java programming. The algorithm for this module is given in Fig. 15

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.

Figure 15. Communication with router algorithm

F. System Configuration

As we have already seen that the file processing module starts thread to read and sort and detect attacks from a continually updating file; the communication with router module has the burden of holding socket communication with routers when commands need to be fired; the Syslog server program is continually receiving UDP packets at port 514. All above points are reasons enough to say that a system with more RAM and processing power will be recommended. Monitoring softwares do require a high configuration system. Same applies to our software. A system with minimum of 8 GB RAM and 2 Hz processing power is recommended. The software when run on a machine with lesser configuration will take about 1 second to configure a command on router through socket connection or to display an alert after the attack.

III. CONCLUSION

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 paper 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. SO IT KNOWS A LOT. We used UDP protocol to receive log which was an unreliable protocol. A TCP based Syslog can also be created where utmost reliability is required. One more issue is that, if the log format of the routers change, then changes will be required in the regex module accordingly.

REFERENCES

[1] Charalampos Patrikakis, Michalis Masikos, and Olga Zouraraki, Distributed Denial of Service Attacks, The Internet Protocol Journal - Volume 7, Number 4, 2004.

[2] ICMP Attacks Illustrated, SANS Institute InfoSec Reading Room

[3] Michael Sudkovitch and David I. Roitman, OSPF Security project book, 2010.

[4] http:// www. omnisecu. com/ cisco- certified –network –associate - ccna/ how- to- communicate- with- a- router.htm

[5] Anand Deveriya, An overview of the Syslog protocol, Cisco Press, 2005.

[6] Karsten Iwen, Logging in Cisco IOS.

[7] Cisco IOS Debug Command Reference, Release 12.3.

AUTHORS PROFILE

Saili Waichal is a post graduate student in computer engineering and IT department, VJTI, Matunga, Mumbai. Her area of interest includes Computer Network Design and Security, Data Structures and Algorithms.

B.B.Meshram is a Professor and Head of Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Matunga, Mumbai. He is Ph.D. in Computer Engineering. He has been in the academics & research since 20 years. His current research includes database technologies, data mining, Information securities, forensic analysis, video processing and distributed computing. He has authored over 203 research publications, out of which over 38 publications at National, 91 publications at international conferences, and more than 71 in international journals, also he has filed two patents. He has given numerous invited talks at various conferences, workshops, training programs and also served as chair/co-chair for many conferences/workshops in the area of computer science and engineering. The industry demanded M.Tech program on Network Infrastructure Management System, and the International conference "Interface" are his brain childs to interface the industry, academia & researchers.