Firewalls and IDS

COMP 232

LAB 06

RSYSLOG

Karan Tank

Introduction

Rsyslog is a powerful, secure and high performance log processing system which accepts data from different types of source and outputs it into multiple format. The source includes system, security, applications and many more. RSYSLOG is defined as “the rocket-fast system for log processing”. rsyslog has evolved into a kind of Swiss army knife of logging. Rsyslog can deliver one million messages per second to local destinations when limited processing is applied.

In this lab, we will be configuring a syslogd central logging server and use it to collect logs from a linux and a windows client.

Step 1

Most of the Linux system has already installed rsyslog in it, but if not you can simply install it using the command **apt-get install rsyslog,** but before you install the rsyslog using apt-get, update the apt-get with new packages.

```
root@kali:~# apt-get update
Hit:1 http://kali.download/kali kali-rolling InRelease
Reading package lists... Done
root@kali:~# apt-get install rsyslog
E: Could not get lock /var/lib/dpkg/lock-frontend - open (11: Resource temporarily unavailable)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), is another process using it?
root@kali:~# ^C
root@kali:~# ps aux | grep -i apt
_apt      2535  0.4  0.3  21740  7784 ?        S    17:16   0:00 /usr/lib/apt/methods/http
_apt      2536 17.1  0.3  21736  7784 ?        S    17:16   0:15 /usr/lib/apt/methods/http
root      2673  0.0  0.0   6144   884 pts/0    S+   17:18   0:00 grep -i apt
root@kali:~# kill -9 2535
root@kali:~# ps aux | grep -i apt
root      2680  0.0  0.0   6144   884 pts/0    S+   17:18   0:00 grep -i apt
root@kali:~# apt-get install rsyslog
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  rsyslog-mysql | rsyslog-pgsql rsyslog-mongodb rsyslog-doc rsyslog-openssl | rsyslog-gnutls rsyslog-gssapi rsyslog-relp
The following packages will be upgraded:
  rsyslog
1 upgraded, 0 newly installed, 0 to remove and 1705 not upgraded.
Need to get 0 B/678 kB of archives.
After this operation, 23.6 kB of additional disk space will be used.
Reading changelogs... Done
(Reading database ... 407211 files and directories currently installed.)
Preparing to unpack .../rsyslog_8.1911.0-1_amd64.deb ...
Unpacking rsyslog (8.1911.0-1) over (8.1901.0-1) ...
Setting up rsyslog (8.1911.0-1) ...
Installing new version of config file /etc/logcheck/ignore.d.server/rsyslog ...
Removing obsolete conffile /etc/default/rsyslog ...
Processing triggers for man-db (2.8.5-2) ...
Processing triggers for systemd (241-3) ...
root@kali:~#
```

Figure 1-Updating apt and installing rsyslog

You can simply start rsyslog using the command **systemctl start rsyslog** and check the status using command **systemctl status rsyslog**

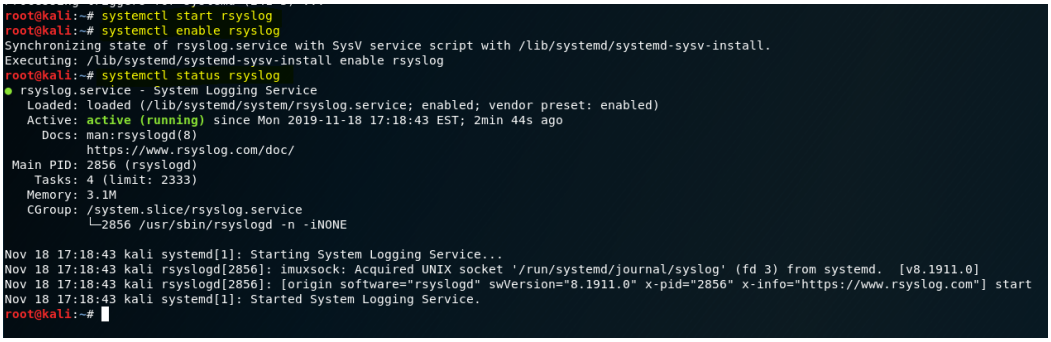


Figure 2-start and enable rsyslog

it should say active status and if there is any error starting up the rsyslog, it should pop up in the status and should give the path for the error

The config file for ryslog is **/etc/rsyslog.conf**

we will edit the conf file and make sure that the syslog server accepts connections from tcp and udp both. and collect logs from client machine and get saved in the specific path.

Figure 3- updating rsyslog.conf

i created two rules, one for linux named **kali** and one for windows named **windows10**.

You can simply create rules by adding

ruleset(name="YourName"){

action(type="omfile" file="/path/you/want/your/file/in")

}

```
###############
#### RULES ####
###############

#defining new rules for linux system
ruleset(name="Kali"){
        action(type="omfile" file="/var/log/linuxclient/linux.log")
}

#defining rules for windows system
ruleset(name="Windows10"){
        action(type="omfile" file="/var/log/windowsClient/win.log")
}
```

Figure 4-Rules for linux and windows

Once you have set the rules, You need to load the rules in the modules

For this, we will add

input(type="inudp" port="5141" ruleset="YourRuleName")

input(type="intcp" port="514" ruleset="YourRuleName")

```
#################
#### MODULES ####
#################

module(load="imuxsock") # provides support for local system logging
module(load="imklog")   # provides kernel logging support
#module(load="immark")  # provides --MARK-- message capability

# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="5141" ruleset="Windows10")

# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="514 ruleset="Kali")
```

Figure 5-Loading Modules for TCP and UDP

The SS command will dump all the static from sockets. you can use grep to pull out the static of rsyslog particularly

for this use the command **ss -tulnp l grep “rsyslog”**

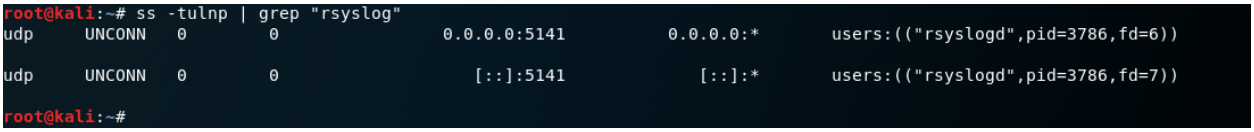


Figure 6

after this, check restart the rsyslog again and check the status ensure that everything is working perfectly fine.

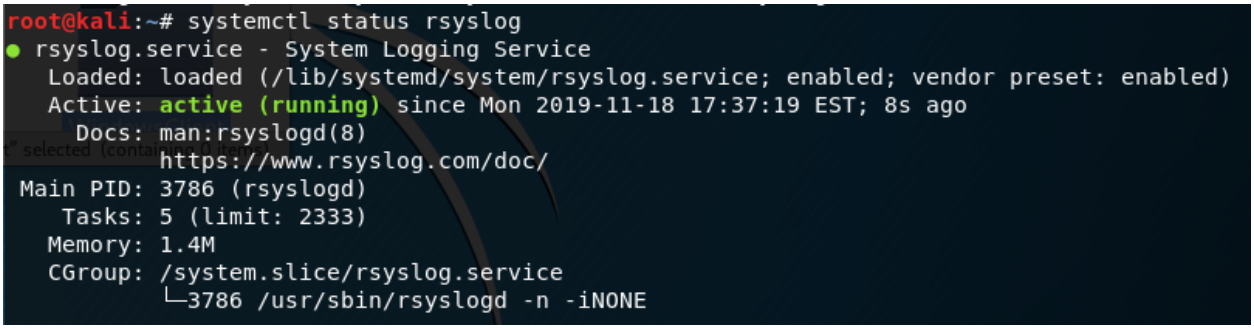


Figure 7-Checking the status again

After this, on the client side, on the rsyslog.conf file add the line
auth,authpriv.*@@ser.ver.ip.address:port

This will forward the security logs on the server to the desired port.

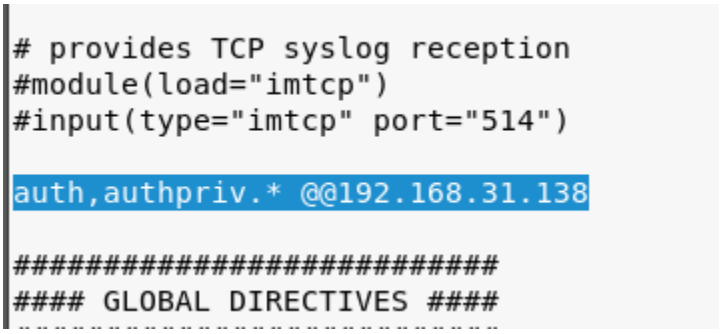


Figure 8-Client config

The above setup on the client side was for linux client, but for windows client, you will need to install rsyslog windows agent. I download version 6.0

Figure 9-installing Rsyslog windows agent

You will need to specify server's ip address and the port number.

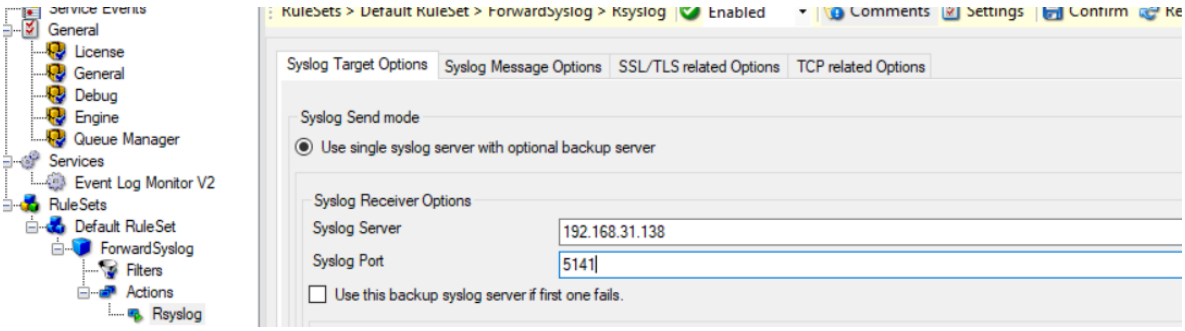


Figure 10

Running

Once you have set up both the clients, You should see the log files are automatically created on the desired location and you can tail the logs using **tail -f log_file.**

I tried to generate logs using setting up the SSH connection and trying to connect using wrong password which created security logs.

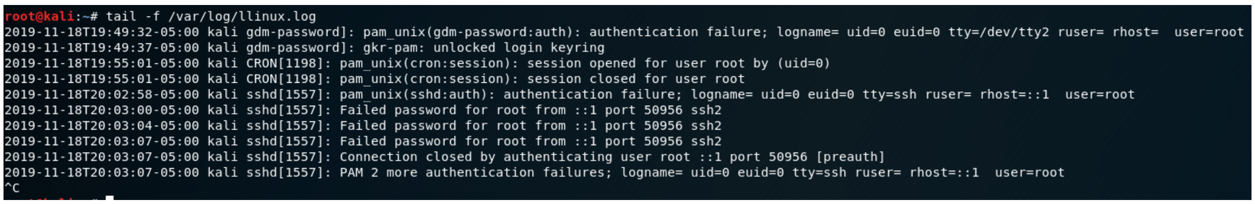


Figure 11- Linux logs

Figure 12-Windows logs

In the wireshark dump, you set the filter to RSH and it will show you all the rsyslog forwarded event logs.

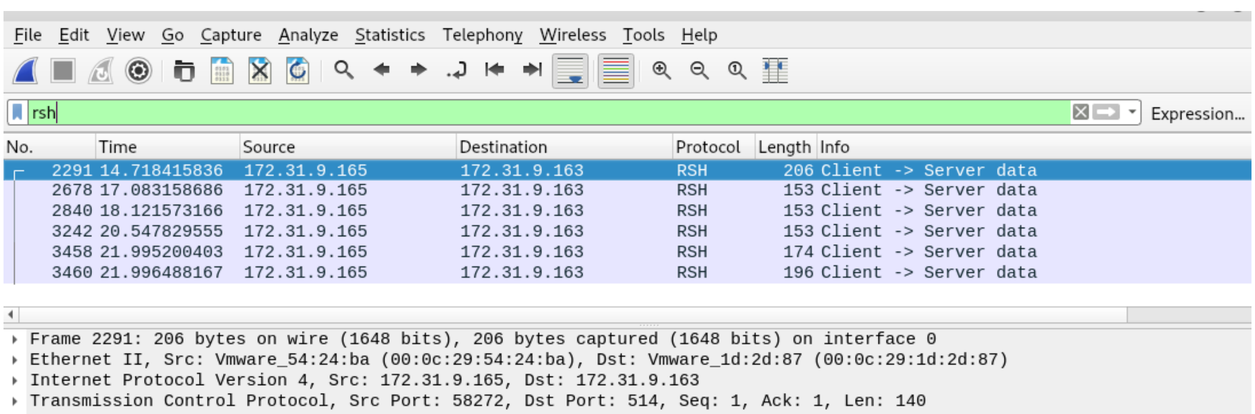
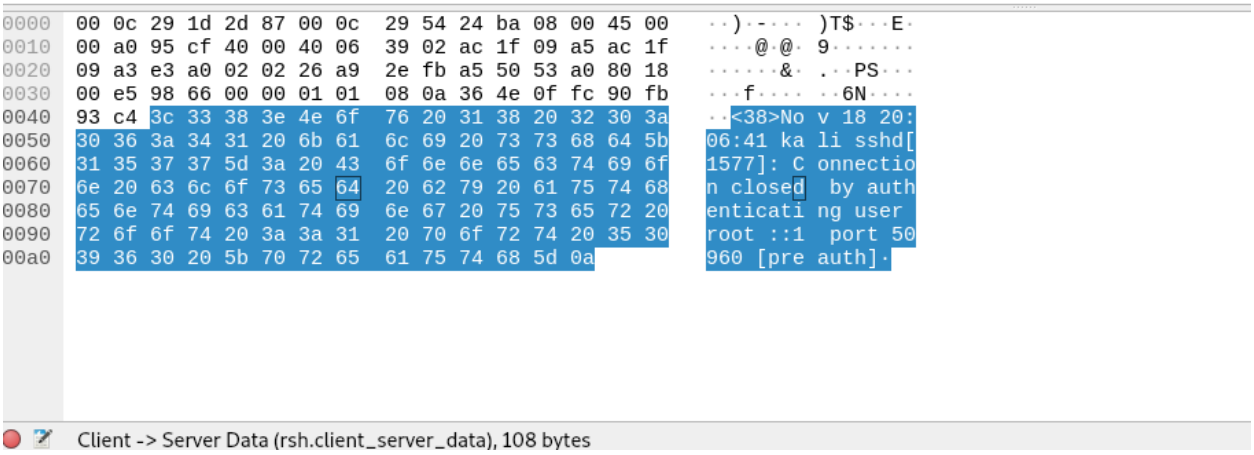


Figure 13


0000 00 0c 29 1d 2d 87 00 0c 29 54 24 ba 08 00 45 00 ··)·-···)T$···E·
0010 00 a0 95 cf 40 00 40 06 39 02 ac 1f 09 a5 ac 1f ····@·@· 9·······
0020 09 a3 e3 a0 02 02 26 a9 2e fb a5 50 53 a0 80 18 ·····&· ···PS···
0030 00 e5 98 66 00 00 01 01 08 0a 36 4e 0f fc 90 fb ···f···· ··6N····
0040 93 c4 3c 33 38 3e 4e 6f 76 20 31 38 20 32 30 3a ··<38>No v 18 20:
0050 30 36 3a 34 31 20 6b 61 6c 69 20 73 73 68 64 5b 06:41 ka li sshd[
0060 31 35 37 37 5d 3a 20 43 6f 6e 6e 65 63 74 69 6f 1577]: C onnectio
0070 6e 20 63 6c 6f 73 65 64 20 62 79 20 61 75 74 68 n closed by auth
0080 65 6e 74 69 63 61 74 69 6e 67 20 75 73 65 72 20 enticati ng user
0090 72 6f 6f 74 20 3a 3a 31 20 70 6f 72 74 20 35 30 root ::1 port 50
00a0 39 36 30 20 5b 70 72 65 61 75 74 68 5d 0a 960 [pre auth]·
Client -> Server Data (rsh.client_server_data), 108 bytes