

Lab 2.2 - Syslog Organization on log01

💡 We will spend considerable time both implementing security controls and the means to monitor these controls. An understanding of logging and logging architecture is critical for continuous monitoring. We will start with traditional syslog servers and later we will leverage host based agents to report events of interest.

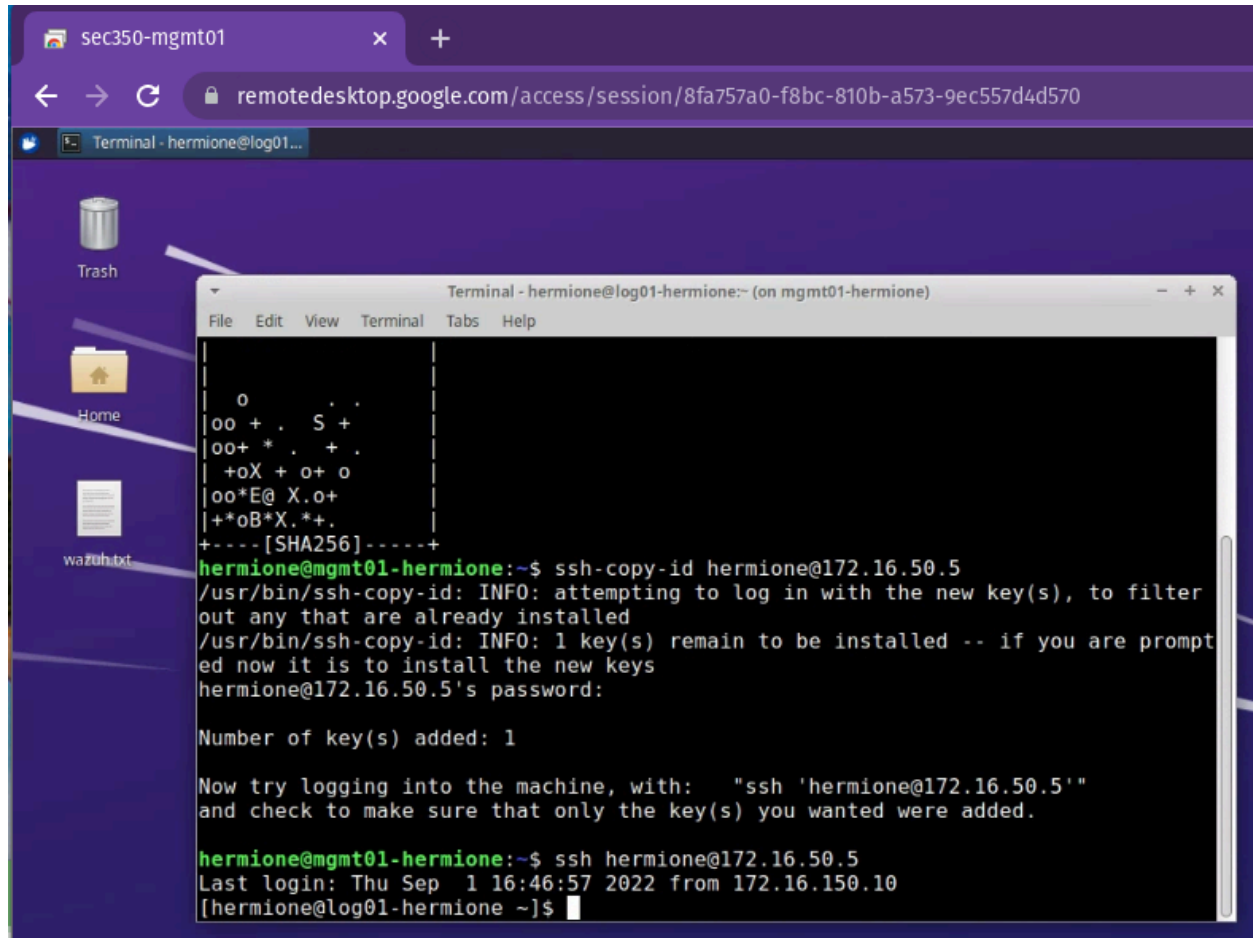
I. Set up mgmt01

mgmt01 is an xubuntu system that will be used to simplify remote management, giving you the ability to copy paste to your internal systems to include vyos.

1. mgmt01 should be placed on your LAN (update VMware adapter)
2. give it the appropriate [IP address](#)
3. change the default password and create a named administrative user.
4. **Note:** Remember that NAT source rule for DMZ?
 - You need to do something similar for the LAN. Also, make sure to add additional DNS forwarding entries to take in account the new listening address as well as the allowed ip addresses for the LAN.
5. Install Chrome Remote Desktop on mgmt01. Here's a [tutorial](#).

Deliverable 1. Using a chrome remote desktop session on mgmt01, ssh into your log01's named user account similar to the screenshot below. (Note, the session below uses ssh key authentication which you are welcome to configure). Provide a screenshot that shows your CRD session as well as your SSH login.

Updated Sep 2, 2023



II. log01: Log Organization

💡 Having all of our remote logs stuffed into log01's **/var/log/messages** or **/var/log/secure** is not helpful. Remote logs should be segregated and ideally stored on reliable and redundant storage in a manner that supports dealing with discrete event types. We are going to store logs in a directory hierarchy in order to provide this organization.

We are going to back out our changes from Lab 1 to the main **/etc/rsyslog.conf** file on log01 and make a custom "drop-in" configuration file for sec350.

The screenshot below shows the **comments (#)** being reapplied.

```
# Provides UDP syslog reception
# for parameters see http://www.rsyslog.com/doc/imudp.html
#module(load="imudp") # needs to be done just once
#input(type="imudp" port="514")

# Provides TCP syslog reception
# for parameters see http://www.rsyslog.com/doc/imtcp.html
#module(load="imtcp") # needs to be done just once
#input(type="imtcp" port="514")

#### GLOBAL DIRECTIVES ####
```

Custom rsyslog drop in file

Examine the following linked [03-sec350.conf](http://10.0.17.3/03-sec350.conf) syslog configuration file

From your VMs (SEC-350-WAN), it can also be accessed from the SEC 350 web server:

<http://10.0.17.3/03-sec350.conf>

```
raw.githubusercontent.com/gmcyber/sec350-share/main/03-sec350.conf

module(load="imudp")
input(type="imudp" port="514" ruleset="RemoteDevice")
template(name="DynFile" type="string"
          string="/var/log/remote-syslog/%HOSTNAME%/%$YEAR%.$MONTH%.$DAY%.$PROGRAMNAME%.log"
)
ruleset(name="RemoteDevice"){
    action(type="omfile" dynaFile="DynFile")
}
```

Manually typing this file is no fun, If you have figured out how to achieve full on copy paste that is the way to go. Alternatively a wget similar to below will save you some time. However, you can use wget <http://10.0.17.3/03-sec350.conf>

```
[hermione@log01-hermione ~]$ sudo -i
[sudo] password for hermione:
[root@log01-hermione ~]# nano /etc/rsyslog.conf
[root@log01-hermione ~]# cd /etc/rsyslog.d/
[root@log01-hermione rsyslog.d]# wget https://raw.githubusercontent.com/gmcyber/sec350-share/main/03-sec350.conf
--2022-01-22 10:45:05-- https://raw.githubusercontent.com/gmcyber/sec350-share/main/03-sec350.conf
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133,
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 275 [text/plain]
Saving to: '03-sec350.conf'

100%[=====>] 275          --.-K/s   in 0s

2022-01-22 10:45:05 (5.89 MB/s) - '03-sec350.conf' saved [275/275]

[root@log01-hermione rsyslog.d]# cat 03-sec350.conf
module(load="imudp")
input(type="imudp" port="514" ruleset="RemoteDevice")
template(name="DynFile" type="string"
          string="/var/log/remote-syslog/%HOSTNAME%/%$YEAR%.$MONTH%.$DAY%.$PROGRAMNAME%.log"
)
ruleset(name="RemoteDevice"){
    action(type="omfile" dynaFile="DynFile")
}
```

Updated Sep 2, 2023

The config file should be saved to /etc/rsyslog.d

This configuration file (03-sec350.conf) will dynamically create and name files based upon hostname, date and process name. Input over udp 514 is associated with the RemoteDevice ruleset which in turn uses the dynamic template configuration called "DynFile".

Restart rsyslog and test

Restart the rsyslog service on log01(1), and repeat your tests from the first logging lab by using logger on web01 (2)

```
[root@log01-hermione rsyslog.d]# systemctl restart rsyslog 1
[root@log01-hermione rsyslog.d]# ls -lR --color /var/log/remote-syslog/ 3
/var/log/remote-syslog/:
total 0
drwx----- 2 root root 35 Jan 22 11:01 web01-hermione

/var/log/remote-syslog/web01-hermione:
total 4
-rw----- 1 root root 83 Jan 22 11:01 2022.01.22.SEC350.log
[root@log01-hermione rsyslog.d]# cat /var/log/remote-syslog/web01-hermione/2022.01.22.SEC350.log 4
Jan 22 11:01:11 web01-hermione SEC350: Testing web01- custom rsyslog configuration
[root@log01-hermione rsyslog.d]# █

[hermione@web01-hermione 2]# logger -t SEC350 Testing web01->log01 custom rsyslog configuration
[hermione@web01-hermione ~]$ █
```

Deliverable 2. Provide a screenshot that shows steps 3 and 4 from above.

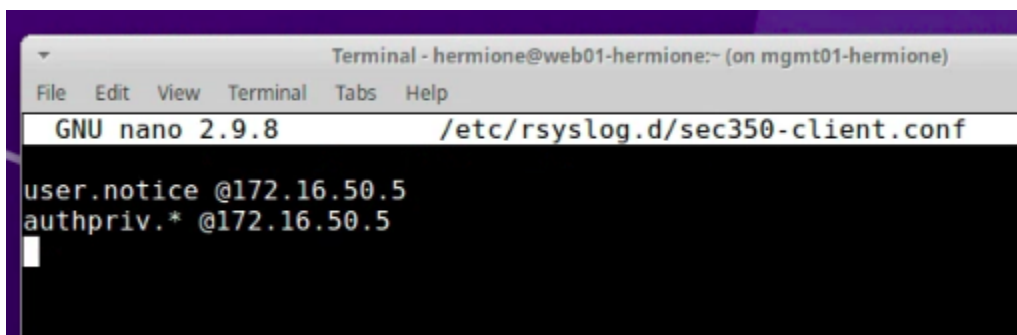
web01: Logging Authorization Events

Modify the rsyslog client configuration on **web01** so that authentication events are forwarded to our log server. **(NOTE: check your Tech Journal for web01's client config file)**

Make sure to restart the rsyslog service on web01.

Reminder. This config file entry goes on web01 not log01.

(If you botch this you will set up a logging loop that will rather quickly fill up log01's hard drive.)



```
Terminal - hermione@web01-hermione:~ (on mgmt01-hermione)
File Edit View Terminal Tabs Help
GNU nano 2.9.8 /etc/rsyslog.d/sec350-client.conf
user.notice @172.16.50.5
authpriv.* @172.16.50.5
█
```

rw01->ssh->web01

SSH into web01 from rw01, make sure you type the wrong password at least once, if you've enabled keybased authentication, passwords aren't really an issue so use an invalid user instead.

```
Last login: Sat Jan 21 19:08:33 2023 from 10.0.17.110
hermione@rw01-hermione:~$ ssh hermione@172.16.50.3
The authenticity of host '172.16.50.3 (172.16.50.3)' can't be established.
ED25519 key fingerprint is SHA256:Kfd+T+CT0uZZ+UBxcMs61aVTSQySMpUURP9atuoEggo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.16.50.3' (ED25519) to the list of known hosts.
hermione@172.16.50.3's password:
Permission denied, please try again.
hermione@172.16.50.3's password:
Permission denied, please try again.
hermione@172.16.50.3's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last failed login: Sat Jan 21 19:28:30 EST 2023 from 10.0.17.100 on ssh:notty
There were 2 failed login attempts since the last successful login.
Last login: Sat Jan 21 19:24:48 2023 from 172.16.150.10
[hermione@web01-hermione ~]$
```

Deliverable 3. Login to log01 via mgmt01, Take a screenshot showing the failed login from your mgmt01 linux system.

```
Terminal - root@log01-hermione:/var/log/remote-syslog/web01-hermione (on mgmt01-hermione)
File Edit View Terminal Tabs Help
hermione@mgmt01-hermione:~$ ssh hermione@172.16.50.5
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Sat Jan 21 19:16:49 2023 from 172.16.150.10
[hermione@log01-hermione ~]$ sudo -i
[sudo] password for hermione:
[root@log01-hermione ~]# cd /var/log/remote-syslog/web01-hermione/
[root@log01-hermione web01-hermione]# ls
2023.01.19.hermione.log 2023.01.21.sshd.log 2023.01.21.unix_chkpwd.log
2023.01.21.polkitd.log 2023.01.21.test.log
[root@log01-hermione web01-hermione]# cat 2023.01.21.sshd.log
2023-01-21T19:27:32-05:00 web01-hermione sshd[28210]: Received disconnect from 172.16.150.10 port 43
2023-01-21T19:27:32-05:00 web01-hermione sshd[28210]: Disconnected from user hermione 172.16.150.10
2023-01-21T19:27:32-05:00 web01-hermione sshd[28207]: pam_unix(sshd:session): session closed for use
2023-01-21T19:28:23-05:00 web01-hermione sshd[28274]: pam_unix(sshd:auth): authentication failure; u
user=hermione
2023-01-21T19:28:25-05:00 web01-hermione sshd[28274]: Failed password for hermione from 10.0.17.100
2023-01-21T19:28:30-05:00 web01-hermione sshd[28274]: Failed password for hermione from 10.0.17.100
2023-01-21T19:28:35-05:00 web01-hermione sshd[28274]: Accepted password for hermione from 10.0.17.10
2023-01-21T19:28:35-05:00 web01-hermione sshd[28274]: pam_unix(sshd:session): session opened for use
2023-01-21T19:29:56-05:00 web01-hermione sshd[28280]: Received disconnect from 10.0.17.100 port 3310
2023-01-21T19:29:56-05:00 web01-hermione sshd[28280]: Disconnected from user hermione 10.0.17.100 po
2023-01-21T19:29:56-05:00 web01-hermione sshd[28274]: pam_unix(sshd:session): session closed for use
[root@log01-hermione web01-hermione]#
```


fw01: Logging Authorization Events

We are going to adjust the vyos configuration to send authentication messages from fw01 to log01. Note, vyos does produce a ton of useless authentication messages which we are going to have to deal with at some point.

💣 Get in the habit of changing the default passwords, particularly on systems that are exposed to others. Figure out how to do this on vyos. You can also use ssh key-based authentication.

```
[edit]
vyos@fw01-hermione# set system syslog host 172.16.50.5 facility authpriv level info
[edit]
vyos@fw01-hermione# commit
[edit]
vyos@fw01-hermione# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@fw01-hermione#
```

Exit out of vyos repeatedly until you are forced to login again you can do this over ssh or from the console. Login again. This should send fw01 authentication events to your log01 server. Make sure you make a mistake on one of the logins.

```
Connection to 172.16.150.2 closed.
hermione@mgmt01-hermione:~$ ssh invaliduser@172.16.150.2
invaliduser@172.16.150.2's password:
Permission denied, please try again.
invaliduser@172.16.150.2's password:
Permission denied, please try again.
invaliduser@172.16.150.2's password:
invaliduser@172.16.150.2: Permission denied (publickey,password).
hermione@mgmt01-hermione:~$
```

Updated Sep 2, 2023

Deliverable 4. Submit a screenshot showing the tree structure of log01 /var/log/remote-syslog directory **as well as** the contents of a failed login message from fw01. It should look like the following screenshot. If tree is missing, install it.

```
[root@log01-hermione remote-syslog]# tree .
.
├── fw01-hermione
│   ├── 2023.01.21.agetty.log
│   └── 2023.01.21.sshd.log
└── web01-hermione
    ├── 2023.01.19.hermione.log
    ├── 2023.01.21.polkitd.log
    ├── 2023.01.21.sshd.log
    ├── 2023.01.21.test.log
    └── 2023.01.21.unix_chkpwd.log

2 directories, 7 files
[root@log01-hermione remote-syslog]# cat /var/log/remote-syslog/fw01-hermione/2023.01.21.sshd.log | grep invalid
[root@log01-hermione remote-syslog]# cat /var/log/remote-syslog/fw01-hermione/2023.01.21.sshd.log
2023-01-22T00:34:21-05:00 fw01-hermione sshd[147119]: pam_unix(sshd:session): session closed for user vyos
2023-01-22T00:34:34-05:00 fw01-hermione sshd[147782]: pam_unix(sshd:auth): check pass; user unknown
2023-01-22T00:34:34-05:00 fw01-hermione sshd[147782]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh
0
2023-01-22T00:34:39-05:00 fw01-hermione sshd[147782]: pam_unix(sshd:auth): check pass; user unknown
2023-01-22T00:34:44-05:00 fw01-hermione sshd[147782]: pam_unix(sshd:auth): check pass; user unknown
2023-01-22T00:34:47-05:00 fw01-hermione sshd[147782]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= r
[root@log01-hermione remote-syslog]# cat /var/log/remote-syslog/fw01-hermione/
```

Deliverable 5. Tech Journal Entry

- Augment your documentation to include how to change a vyos password
- If you've not done so in your SYS classes, make sure to document how to use ssh keybased authentication. Make this happen from mgmt01 to at least web01 or log01.
- Address how to log authpriv messages on linux systems
- Augment your rsyslog documentation to cover the new drop in file configuration on the server as well as the changes to the web01 client to forward authentication events.
- Describe how to forward authentication events from vyos to a remote syslog server
- Make sure to capture any difficulties or observations as reflections