

Report 11

For this week's lab focused on part 2 of Linux forensics using the Linux audit system. The first task involved using the Linux Auditing System by utilizing the (auditd) package. The next task involved Security Auditing using the package called Lynis.

For the first task, we needed to install the Linux audit system **auditd**. This is a component that is added to the audit component Linux Audit System This utility is responsible for writing audit records to the disk. Viewing the logs is done with the **ausearch** or **aureport** utilities. Configuring the audit system or loading rules is done with the **auditctl** utility. To begin install the auditd utility, use **sudo apt-get install auditd** command. Here below is the output.

```
(kali㉿kali)-[~]  
$ sudo apt-get install auditd  
[sudo] password for kali:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done
```

Once installed, we need to display the status **auditd**. To display the status use **systemctl status auditd | grep -i active**. Note the that the status shows as inactive.

```
(kali㉿kali)-[~]  
$ systemctl status auditd | grep -i active  
Active: inactive (dead)
```

Since it is inactive, we need to start the service. To do that, use **systemctl start auditd** command. Here below is the output. Note that no output was shown.

```
(kali㉿kali)-[~]  
$ systemctl start auditd
```

Now, if we want to verify the status of **auditd**, we need to use the previous command **systemctl status auditd | grep -i active** that we used to check the status of **auditd**. Here below is the output. Note that the status now is active and its running. You also see the time and date the status has been active.

```
(kali㉿kali)-[~]  
$ systemctl status auditd | grep -i active  
Active: active (running) since Wed 2022-11-09 20:13:35 CST; 37s ago
```

The Linux Auditing System is a native auditing system to Linux kernel. Some of these components include the **auditctl** which controls the behavior of the daemon, adds rules etc. The **/etc/audit/audit.rules** file contains the rules and various parameters of the auditd daemon.

The component **aureport** generate report of the activity on a system. The component **ausearch** searches for various events. **auditspd** which can be used to relay event notifications to other applications instead of writing them to disk in the audit log. The **autrace** command can be used to trace a process, much like **strace**. These audit daemon configurations are options that system admins can choose to customize. These are controlled in the **/etc/audit/auditd.conf** file. You can list the contents of the audit folder using **sudo ls -l /etc/audit/** command. Here below is the output.

```
(kali㉿kali)-[~]
└─$ sudo ls -l /etc/audit/
total 20
-rw-r----- 1 root root 881 Mar 13 2022 auditd.conf
-rw-r----- 1 root root 107 Nov  3 19:15 audit.rules
-rw-r----- 1 root root 127 Mar 13 2022 audit-stop.rules
drwxr-x--- 2 root root 4096 Nov  9 20:08 plugins.d
drwxr-x--- 2 root root 4096 Nov  9 20:08 rules.d
```

To view the details of the file configurations of the audit daemon, use the command **sudo cat /etc/audit/auditd.conf | head**. Here below is the output. You see some of option that are enabled such as the local events and writing logs.

```
(kali㉿kali)-[~]
└─$ sudo cat /etc/audit/auditd.conf | head
#
# This file controls the configuration of the audit daemon
#
local_events = yes
write_logs = yes
log_file = /var/log/audit/audit.log
log_group = adm
log_format = ENRICHED
flush = INCREMENTAL_ASYNC
```

The **auditctl** can be used to control the audit system. You can view the query status of the audit daemon by using the command **sudo auditctl -s**. Here below is the output.

```
(kali㉿kali)-[~]
└─$ sudo auditctl -s
enabled 1
failure 1
pid 4171
rate_limit 0
backlog_limit 8192
lost 0
backlog 0
backlog_wait_time 60000
backlog_wait_time_actual 0
loginuid_immutable 0 unlocked
```

Note that depending on the flag determines the meaning and possible values. For example, for **enabled** there are 3 options. These are 0 – disabled, 1 - enable, and 2 – enable and lock down the configuration. The **pid** is the process Id of **auditd**. The failure flag specifies how the kernel will handle critical errors. These options include 0 – Silent, 1 – printk, and 2 – panic. The lost flag refers to the amount of lost audit messages.

Now to enable the audit system, use **sudo auditctl -e 1** command. Note that there is no output. If you want to view audit events use the command **sudo cat /var/log/audit/audit.log | grep -I**

syscall | head -n 1. You can choose the number of events you want to display by changing the number. Here below is the output only displaying one. Note that the type of event shown is a system call (SYSCALL). You can use this to determine the type of events that occur and what caused them to trigger them.

```
(kali@kali)-[~]
$ sudo cat /var/log/audit/audit.log | grep -i syscall | head -n 1
type=SYSCALL msg=audit(1667521852.069:4): arch=c000003e syscall=44 success=yes exit=60 a0=3 a1=7ffdd3c6f450 a2=3c a3=0 items=0 ppid=6225 pid=6235 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="auditctl" exe="/usr/sbin/auditctl" subj=unconfined key=(null)ARCH=x86_64 SYSCALL=sendto AUID="unset" UID="root" GID="root" EUID="root" SUID="root" FSUID="root" EGID="root" SGID="root" FSGID="root"
```

The **aureport** utility is responsible for producing summary reports of the audit system logs. These reports are stored in the **/var/log/audit/audit.log** file. You can view the reports by using the command **sudo aureport -if /var/log/audit/audit.log**. You can see below the details of the report. These include the number of users, logins, and changes in configuration.

```
(kali@kali)-[~]
$ sudo aureport -if /var/log/audit/audit.log

Summary Report
=====
Range of time in logs: 11/03/2022 19:30:52.008 - 11/09/2022 20:27:04.762
Selected time for report: 11/03/2022 19:30:52 - 11/09/2022 20:27:04.762
Number of changes in configuration: 14
Number of changes to accounts, groups, or roles: 0
Number of logins: 0
Number of failed logins: 0
Number of authentications: 1
Number of failed authentications: 1
Number of users: 4
Number of terminals: 8
Number of host names: 1
Number of executables: 9
Number of commands: 7
Number of files: 22
Number of AVC's: 0
Number of MAC events: 0
Number of failed syscalls: 10
Number of anomaly events: 1
Number of responses to anomaly events: 0
Number of crypto events: 0
Number of integrity events: 0
Number of virt events: 0
Number of keys: 0
Number of process IDs: 81
Number of events: 586
```

The **aureport** utility can also display authentication attempts. You can view this by using the **aureport -au** command. Note the details of the attempts and as well as how many were made.

```
(kali@kali)-[~]
$ aureport -au
Error opening config file (Permission denied)
NOTE - using built-in logs: /var/log/audit/audit.log

Authentication Report
=====
# date time acct host term exe success event
=====
1. 11/03/2022 19:58:29 root ? /dev/pts/0 /usr/bin/su no 99
2. 11/03/2022 20:26:46 root ? /dev/pts/1 /usr/bin/su yes 432
```

Viewing the logins attempts is also great to know. To view this, use the **sudo aureport -l** command. Note that there were no logins attempts on my system.


```
(kali㉿kali)-[~]
$ sudo aureport -l

Login Report
=====
# date time auid host term exe success event
=====
<no events of interest were found>
```

Now to view attempts that were not successful, use the command **sudo aureport --failed**. You can see below the amount failed authentications attempts on my system.

```
(kali㉿kali)-[~]
$ sudo aureport --failed

Failed Summary Report
=====
Range of time in logs: 11/03/2022 19:30:52.008 - 11/09/2022 20:31:52.689
Selected time for report: 11/03/2022 19:30:52 - 11/09/2022 20:31:52.689
Number of changes in configuration: 0
Number of changes to accounts, groups, or roles: 0
Number of logins: 0
Number of failed logins: 0
Number of authentications: 0
Number of failed authentications: 1
Number of users: 2
Number of terminals: 3
Number of host names: 1
Number of executables: 3
Number of commands: 2
Number of files: 8
Number of AVC's: 0
Number of MAC events: 0
Number of failed syscalls: 10
Number of anomaly events: 0
Number of responses to anomaly events: 0
Number of crypto events: 0
Number of integrity events: 0
Number of virt events: 0
Number of keys: 0
Number of process IDs: 3
Number of events: 12
```

To view the attempts that were successful, use **sudo aureport --success** command. Here below is the output.

```
(kali㉿kali)-[~]
$ sudo aureport --success

Success Summary Report
=====
Range of time in logs: 11/03/2022 19:30:52.008 - 11/09/2022 20:32:38.272
Selected time for report: 11/03/2022 19:30:52 - 11/09/2022 20:32:38.272
Number of changes in configuration: 14
Number of changes to accounts, groups, or roles: 0
Number of logins: 0
Number of failed logins: 0
Number of authentications: 1
Number of failed authentications: 0
Number of users: 4
Number of terminals: 8
Number of host names: 1
Number of executables: 9
Number of commands: 7
Number of files: 14
Number of AVC's: 0
Number of MAC events: 0
Number of failed syscalls: 0
Number of anomaly events: 1
Number of responses to anomaly events: 0
Number of crypto events: 0
Number of integrity events: 0
Number of virt events: 0
Number of keys: 0
Number of process IDs: 83
Number of events: 561
```

You can also modify the start and end date for your audit report. To do this, **use sudo aureport -ts yesterday -te now --success**. You can specify the **yesterday** and **now** parameters to your choosing. Here below is a summary report of 11-08-2022 to 11-09-22. This is important to know to so that you can view certain days that you want.

```
(kali@kali)-[~]
$ sudo aureport -ts yesterday -te now --success
Success Summary Report
Range of time in logs: 11/03/2022 19:30:52.008 - 11/09/2022 20:33:51.897
Selected time for report: 11/08/2022 00:00:00 - 11/09/2022 20:33:51
Number of changes in configuration: 3
Number of changes to accounts, groups, or roles: 0
Number of logins: 0
Number of failed logins: 0
Number of authentications: 0
Number of failed authentications: 0
Number of users: 3
Number of terminals: 5
Number of host names: 1
Number of executables: 4
Number of commands: 3
Number of files: 0
Number of AVC's: 0
Number of MAC events: 0
Number of failed syscalls: 0
Number of anomaly events: 0
Number of responses to anomaly events: 0
Number of crypto events: 0
Number of integrity events: 0
Number of virt events: 0
Number of keys: 0
Number of process IDs: 19
Number of events: 104
```

If you want to view the user report, use **sudo aureport -u --summary**.

```
(kali@kali)-[~]
$ sudo aureport -u --summary
User Summary Report
=====
total  auid
=====
430  1000
90   0
70  -1
3   130
```

The **auid** mentions the user. For example, 1000 is typically reserved for the user. 0 is reserved for the root. Total refers to the amount of audit reports that each user has.

You can view a user summary in detail by using **sudo aureport -e -i --summary**. Here you see some events that were performed. Such a system call and a login.

```
(kali@kali)-[~]
$ sudo aureport -e -i --summary
Event Summary Report
=====
total  type
=====
125  SYSCALL
74   USER_START
73   USER_ACCT
71   CRED_DISP
71   USER_END
53   CRED_REFR
53   USER_CMD
30   BPF
21   CRED_ACQ
19   LOGIN
14   CONFIG_CHANGE
11   SERVICE_START
9    SERVICE_STOP
2    USER_AUTH
2    DAEMON_START
1    ANOM_ABEND
```

There are times when you to display the events such as the system call and the daemon started in a formatted way. To do this, you can use **sudo aureport -e -ts yesterday -te now | head**. Below, we specify the event that happened yesterday. Note some of the parameters that are shown such as the date, time and if they were successful.

```
(kali㉿kali)-[~]
$ sudo aureport -e -ts yesterday -te now | head

Event Report
=====
# date time event type auid success
=====
1. 11/09/2022 20:13:35 6714 DAEMON_START -1 yes
2. 11/09/2022 20:13:35 4 CONFIG_CHANGE -1 yes
3. 11/09/2022 20:13:35 5 CONFIG_CHANGE -1 yes
4. 11/09/2022 20:13:35 6 CONFIG_CHANGE -1 yes
5. 11/09/2022 20:13:35 7 SERVICE_START -1 yes
```

Now, if you are looking for a process report, use **sudo aureport -p | head**. The report below shows the pid number as well as the exe file that was used. You also see the events and systems call there as well.

```
(kali㉿kali)-[~]
$ sudo aureport -p | head

Process ID Report
=====
# date time pid exe syscall auid event
=====
1. 11/03/2022 19:30:52 6222 ? 0 -1 8146
2. 11/03/2022 19:30:52 6235 /usr/sbin/auditctl 44 -1 4
3. 11/03/2022 19:30:52 6235 /usr/sbin/auditctl 44 -1 5
4. 11/03/2022 19:30:52 6235 /usr/sbin/auditctl 44 -1 6
5. 11/03/2022 19:30:52 1 /usr/lib/systemd/systemd 0 -1 7
```

You can view the system call events in detail by using **sudo aureport -s | head** command. Here below are the system events, Syscall.

```
(kali㉿kali)-[~]
$ sudo aureport -s | head

Syscall Report
=====
# date time syscall pid comm auid event
=====
1. 11/03/2022 19:30:52 44 6235 auditctl -1 4
2. 11/03/2022 19:30:52 44 6235 auditctl -1 5
3. 11/03/2022 19:30:52 44 6235 auditctl -1 6
4. 11/03/2022 19:35:01 1 7292 cron 0 16
5. 11/03/2022 19:39:01 1 8349 cron 0 34
```


The **id** command will the identification number of the current user. Here you can see the uid that we saw earlier to view the user report. The number 1000 refers to the user. Which in this case is the user named kali. Here you also see other ids and their associated number as well.

```
(kali@kali)-[~]
$ id
uid=1000(kali) gid=1000(kali) groups=1000(kali),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),109(netdev),119(wireshark),122(bluetooth),134(scanner),143(kaboxer)
```

Now that we know which **id** belongs to user, you can expand your search by searching for events by a particular user. Here below are some of the events that occurred for this user. If you want to search for events initiated by a given user, you need to specify the user id and then use **ausearch** to do this. To do this use **ausearch -ui 1000 | head** command.

```
(kali@kali)-[~]
$ sudo ausearch -ui 1000 | head
time-->Thu Nov  3 19:33:17 2022
type=USER_CMD msg=audit(1667521997.970:8): pid=6872 uid=1000 auid=1000 ses=2 subj=unconfined msg='op=PAM:accounting grantors=pam_permit acct="kali" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/0 res=success'

time-->Thu Nov  3 19:33:17 2022
type=USER_CMD msg=audit(1667521997.970:9): pid=6872 uid=1000 auid=1000 ses=2 subj=unconfined msg='cwd="/home/kali" cmd=6C73202D0C202F6574632F61756469742F exe="/usr/bin/sudo" terminal=pts/0 res=success'

time-->Thu Nov  3 19:33:17 2022
type=CRED_REFR msg=audit(1667521997.970:10): pid=6872 uid=1000 auid=1000 ses=2 subj=unconfined msg='op=PAM:setcred grantors=pam_permit acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/0 res=success'
```

When you use **autrace** to trace audit processes, it is important to delete any audit rules that are in queue to avoid conflict with **autrace** adds. To delete the audit rules so that this problem does not occur, use **auditctl -D** command. Note that by default there are no rules.

```
(kali@kali)-[~]
$ sudo auditctl -D
No rules
```

Here below is an example tracing the **/usr/bin/less** file. The command that to do this is **sudo autrace /usr/bin/less**. Note now you can locate the record now with the command below.

```
(kali@kali)-[~]
$ sudo autrace /usr/bin/less
Waiting to execute: /usr/bin/less
Missing filename ("less --help" for help)
Cleaning up ...
Trace complete. You can locate the records with 'ausearch -i -p 15045'
```

Now when you enter **sudo ausearch -i -p 15045** command into the terminal, you can see below the output. Note the events displayed in relation to the **/less** file.

```
(kali@kali)-[~]
$ sudo ausearch -i -p 15045
type=PROCTITLE msg=audit(11/09/2022 20:54:50.066:179) : proctitle=autrace /usr/bin/less
type=SYSCALL msg=audit(11/09/2022 20:54:50.066:179) : arch=x86_64 syscall=close success=yes exit=0 a0=0x4 a1=0x1 a2=0x7f49e8b9bad8 a3=0x7f49e8c1b028 items=0 ppid=15043 pid=15045 auid=kali uid=root gid=root euid=root suid=root fsuid=root egid=root sgid=root fsgid=root tty=pts1 ses=2 comm=autrace exe=/usr/sbin/autrace subj=unconfined key=(null)

type=PROCTITLE msg=audit(11/09/2022 20:54:50.066:180) : proctitle=autrace /usr/bin/less
type=PATH msg=audit(11/09/2022 20:54:50.066:180) : item=0 name= inode=4 dev=00:15 mode=character,620 ouid=root ogid=ttty rdev=88:01 nametype=NORMAL cap_fp=none cap_fm=none cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(11/09/2022 20:54:50.066:180) : cwd=/home/kali
type=SYSCALL msg=audit(11/09/2022 20:54:50.066:180) : arch=x86_64 syscall=newfstatat success=yes exit=0 a0=0x1 a1=0x7f49e8b52f15 a2=0x7ffcdca618990 a3=0x1000 items=1 ppid=15043 pid=15045 auid=kali uid=root gid=root euid=root suid=root fsuid=root egid=root sgid=root fsgid=root tty=pts1 ses=2 comm=autrace exe=/usr/sbin/autrace subj=unconfined key=(null)

type=PROCTITLE msg=audit(11/09/2022 20:54:50.066:181) : proctitle=autrace /usr/bin/less
type=SYSCALL msg=audit(11/09/2022 20:54:50.066:181) : arch=x86_64 syscall=write success=yes exit=34 a0=0x1 a1=0x55e056c5c4d0 a2=0x22 a3=0x7f49e8b4c3e0 items=0 ppid=15043 pid=15045 auid=kali uid=root gid=root euid=root suid=root fsuid=root egid=root sgid=root fsgid=root tty=pts1 ses=2 comm=autrace exe=/usr/sbin/autrace subj=unconfined key=(null)

type=PROCTITLE msg=audit(11/09/2022 20:54:50.066:183) : proctitle=autrace /usr/bin/less
type=SYSCALL msg=audit(11/09/2022 20:54:50.066:183) : arch=x86_64 syscall=read success=yes exit=1 a0=0x3 a1=0x7ffcdca619110 a2=0x1 a3=0x7f49e8b4c3e0 items=0 ppid=15043 pid=15045 auid=kali uid=root gid=root euid=root suid=root fsuid=root egid=root sgid=root fsgid=root tty=pts1 ses=2 comm=autrace exe=/usr/sbin/autrace subj=unconfined key=(null)

type=PROCTITLE msg=audit(11/09/2022 20:54:51.070:184) : proctitle=autrace /usr/bin/less
type=SYSCALL msg=audit(11/09/2022 20:54:51.070:184) : arch=x86_64 syscall=close success=yes exit=0 a0=0x3 a1=0x7ffcdca619110 a2=0x1 a3=0x7f49e8b4c3e0 items=0 ppid=15043 pid=15045 auid=kali uid=root gid=root euid=root suid=root fsuid=root egid=root sgid=root fsgid=root tty=pts1 ses=2 comm=autrace exe=/usr/sbin/autrace subj=unconfined key=(null)
```

The second part of the lab involved using Lynis for Security Auditing. This is an auditing tool that used for testing the vulnerability of a system. These include security auditing, compliance testing, penetration testing, vulnerability detection, and system hardening. The first task to do is to install this utility by using **sudo apt-get install lynis**. Here below is the output below.

```
(kali@kali)-[~]
$ sudo apt-get install lynis
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
lynis is already the newest version (3.0.8-1.1).
The following packages were automatically installed and are no longer required:
 fonts-roboto-slab libexporter-tiny-perl libgdk-pixbuf-xlib-2.0-0 libgdk-pixbuf2.0-0 libhttp-server-simple-perl
 liblist-moreutils-perl liblist-moreutils-xs-perl liblttng-ust-ctl4 liblttng-ust0 python3-ipaddr
 python3-singledispatch python3-twisted-bin ruby-atomic ruby-thread-safe sphinx-rtd-theme-common
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 949 not upgraded.
```

Now that we have installed Lynis into your system, we need to view the commands that are available to use. To show that command, use **lynis show commands**. You can see below the various commands and configuration options.

```
(kali@kali)-[~]
$ lynis show commands

Commands:
lynis audit
lynis configure
lynis generate
lynis show
lynis update
lynis upload-only
```

Lynis has various settings to customize its behavior from. To show the settings, use **lynis show settings** commands. This configuration file is generally located at the /etc/audit directory. Here below is the output.

```
(kali@kali)-[~]
$ lynis show settings
# Colored screen output
colors=1

# Compressed uploads
compressed-uploads=0

# Use non-zero exit code if one or more warnings were found
error-on-warnings=0

# Language
language=en

# License key
license-key=[not configured]

# Logging of tests that have a different OS
log-tests-incorrect-os=1

# Machine role (personal, workstation or server)
machine-role=server
```


To perform a security auditing on your system, use **sudo lynis audit system**. Note that this will take a while audit your system. If you want a quick audit, use the **--quick** option. Here below is an output of the first section of audit results.

```
(kali@kali)-[~]
$ sudo lynis audit system
[sudo] password for kali:

[ Lynis 3.0.8 ]

#####
Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
welcome to redistribute it under the terms of the GNU General Public License.
See the LICENSE file for details about using this software.

2007-2021, CISofy - https://cisofy.com/lynis/
Enterprise support available (compliance, plugins, interface and tools)
#####

[+] Initializing program
-----
- Detecting OS ... [ DONE ]
- Checking profiles ... [ DONE ]

-----
Program version: 3.0.8
Operating system: Linux
Operating system name: Kali Linux
Operating system version: Rolling release
Kernel version: 5.14.0
Hardware platform: x86_64
Hostname: kali

-----
Profiles: /etc/lynis/default.prf
Log file: /var/log/lynis.log
Report file: /var/log/lynis-report.dat
Report version: 1.0
Plugin directory: /etc/lynis/plugins

-----
Auditor: [Not Specified]
Language: en
Test category: all
Test group: all

-----
- Program update status ... [ NO UPDATE ]
```

Lynis displays a report when it done running a security audit. These reports include warning and suggestions. There is also a unique identifier. The warning below has the identifier NETW-2704

```
Warnings (1):
-----
! Nameserver 192.168.1.254 does not respond [NETW-2704]
https://cisofy.com/lynis/controls/NETW-2704/

Suggestions (49):
-----
* This release is more than 4 months old. Check the website or GitHub to see if there is an update available. [LYN
IS]
https://cisofy.com/lynis/controls/LYNIS/

* Install libpam-tmpdir to set $TMP and $TMPDIR for PAM sessions [DEB-0280]
https://cisofy.com/lynis/controls/DEB-0280/

* Install apt-listbugs to display a list of critical bugs prior to each APT installation. [DEB-0810]
https://cisofy.com/lynis/controls/DEB-0810/

* Install apt-listchanges to display any significant changes prior to any upgrade via APT. [DEB-0811]
https://cisofy.com/lynis/controls/DEB-0811/

* Install fail2ban to automatically ban hosts that commit multiple authentication errors. [DEB-0880]
https://cisofy.com/lynis/controls/DEB-0880/
```

Now that we know the identifier, we can use **sudo lynis show details NETW-2705** command for more information on this warning. Note that identifier will change depending on your error. This error signifies to check your **resolv.conf** file.

```
(kali@kali)-[~]
$ sudo lynis show details NETW-2705
2022-11-09 21:20:44 Performing test ID NETW-2705 (Check availability two nameservers)
2022-11-09 21:20:44 Result: found at least 2 responsive nameservers
2022-11-09 21:20:44 Hardening: assigned maximum number of hardening points for this item (3). Currently having 109 p
oints (out of 167)
2022-11-09 21:20:44 ==
```

To view the content of this file, use **cat /etc/resolv.conf**. Below are some of nameservers for system. The warning from the Lynis audit report specifies that we need to backup nameservers to the system.

```
(kali@kali)-[~]
$ cat /etc/resolv.conf
# Generated by NetworkManager
search attlocal.net
nameserver 103.86.96.100
nameserver 103.86.99.100
nameserver 192.168.1.254
```

To resolve it, we will add public DNS servers in the system. First thing is login as a super user. This is done by using the command **sudo su**. This is required because you aren't able add the public DNS using a user account. You need root privilege access. Not doing so will result in this error below. Remember that changes done to the system while as a super user can damage your system Always make sure the changes don't negatively affect your system

```
zsh: permission denied: /etc/resolv.conf
```

Now we can add the name server using **echo nameserver 8.8.8.8>> /etc/resolv.conf**, **echo nameserver 75.75.75.75>> /etc/resolv.conf**, and **echo nameserver 75.75.76.76>> /etc/resolv.conf** commands. Here below is the output. Note that the skull represents a super user.

```
(root👤kali)-[/home/kali]
# echo nameserver 8.8.8.8>> /etc/resolv.conf

(root👤kali)-[/home/kali]
# echo nameserver 75.75.75.75>> /etc/resolv.conf

(root👤kali)-[/home/kali]
# echo nameserver 75.75.76.76>> /etc/resolv.conf
```

We can view these changes by using **cat /etc/resolv.conf** command. You can see below the domain servers that were added.

```
(kali@kali)-[~]
$ cat /etc/resolv.conf
# Generated by NetworkManager
search attlocal.net
nameserver 103.86.96.100
nameserver 103.86.99.100
nameserver 192.168.1.254
nameserver 8.8.8.8
nameserver 75.75.75.75
nameserver 75.75.76.76
```

Now we can run the security audit again using **sudo lynis audit system**. Now we can see below that the issue has been resolved.

```
kali@kali [~] sudo lynis audit system
-[ Lynis 3.0.7 Results ]-
```

Great, no warnings

Now you can review the score that you get when the security audit is done auditing. Below you can see the score. You can use this information and see what you can do to increase the security of your system. Note that there are different options when scanning.

Lynis security scan details:

Hardening index : 65 [#####]
Tests performed : 273
Plugins enabled : 1

Components:

- Firewall [V]
- Malware scanner [X]

Scan mode:

Normal [V] Forensics [] Integration [] Pentest []

Lynis modules:

- Compliance status [?]
- Security audit [V]
- Vulnerability scan [V]

Files:

- Test and debug information : /var/log/lynis.log
- Report data : /var/log/lynis-report.dat