Defensive Security 1 (INFO70243)

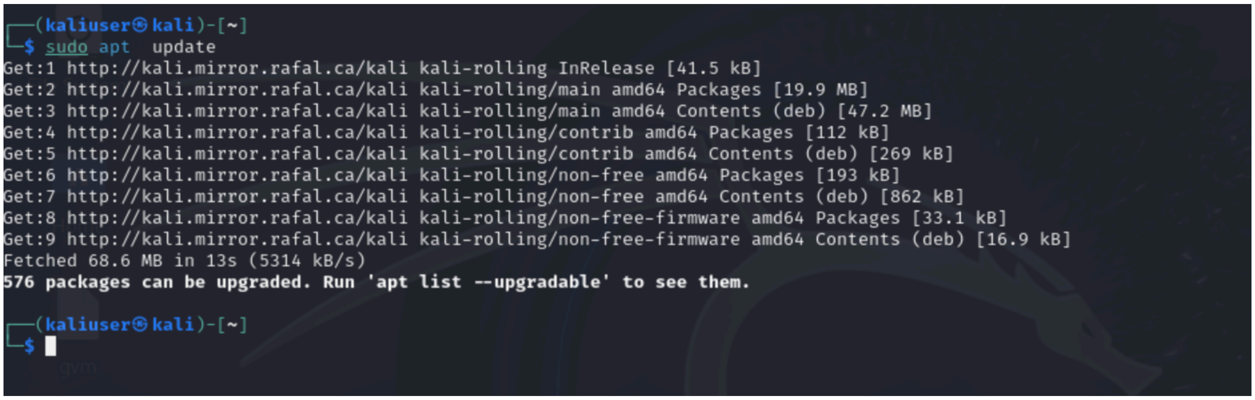
Assignment 1

by

Uthaya Krishnan
Sheridan College

Task 1 – Using chkrootkit to check Linux box

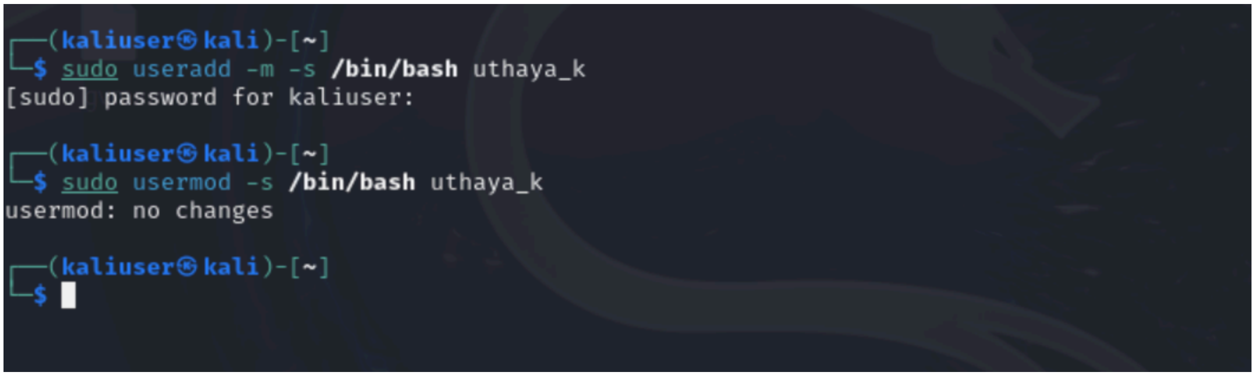
Pre-lab configuration



```
┌──(kaliuser㉿kali)-[~]
└─$ sudo apt  update
Get:1 http://kali.mirror.rafal.ca/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.mirror.rafal.ca/kali kali-rolling/main amd64 Packages [19.9 MB]
Get:3 http://kali.mirror.rafal.ca/kali kali-rolling/main amd64 Contents (deb) [47.2 MB]
Get:4 http://kali.mirror.rafal.ca/kali kali-rolling/contrib amd64 Packages [112 kB]
Get:5 http://kali.mirror.rafal.ca/kali kali-rolling/contrib amd64 Contents (deb) [269 kB]
Get:6 http://kali.mirror.rafal.ca/kali kali-rolling/non-free amd64 Packages [193 kB]
Get:7 http://kali.mirror.rafal.ca/kali kali-rolling/non-free amd64 Contents (deb) [862 kB]
Get:8 http://kali.mirror.rafal.ca/kali kali-rolling/non-free-firmware amd64 Packages [33.1 kB]
Get:9 http://kali.mirror.rafal.ca/kali kali-rolling/non-free-firmware amd64 Contents (deb) [16.9 kB]
Fetched 68.6 MB in 13s (5314 kB/s)
576 packages can be upgraded. Run 'apt list --upgradable' to see them.

┌──(kaliuser㉿kali)-[~]
└─$ ▌
```

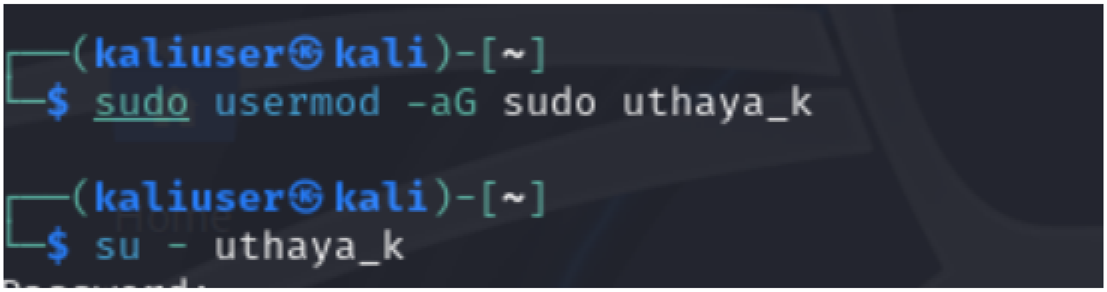
Create a user and his home directory with the same name as you,

Change default shell of the user to /bin/bash



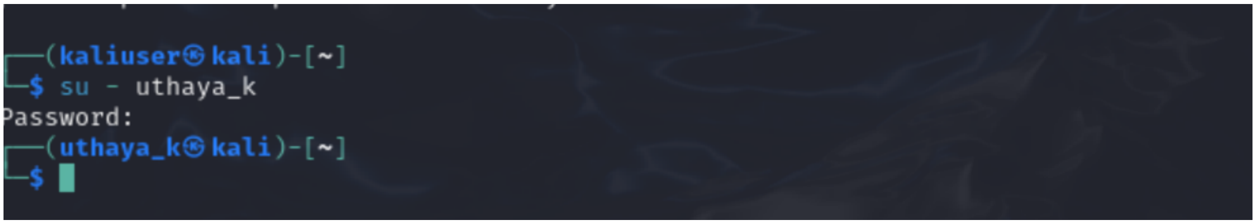
```
┌──(kaliuser㉿kali)-[~]
└─$ sudo useradd -m -s /bin/bash uthaya_k
[sudo] password for kaliuser:

┌──(kaliuser㉿kali)-[~]
└─$ sudo usermod -s /bin/bash uthaya_k
usermod: no changes

┌──(kaliuser㉿kali)-[~]
└─$ ▌
```

Add the user to sudo group



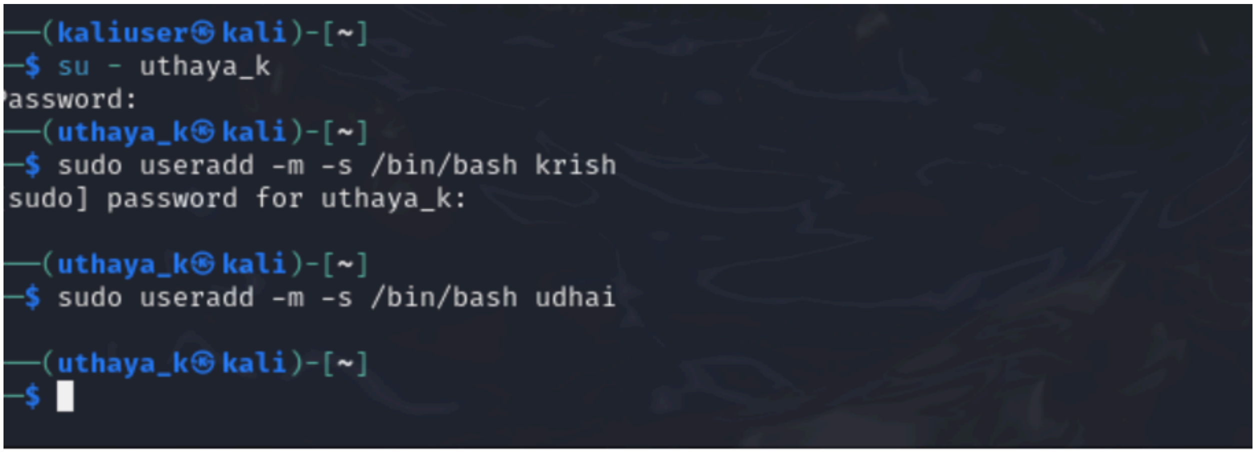
```
┌──(kaliuser㉿kali)-[~]
└─$ sudo usermod -aG sudo uthaya_k

┌──(kaliuser㉿kali)-[~]
└─$ su - uthaya_k
Password:
```

Login as that user



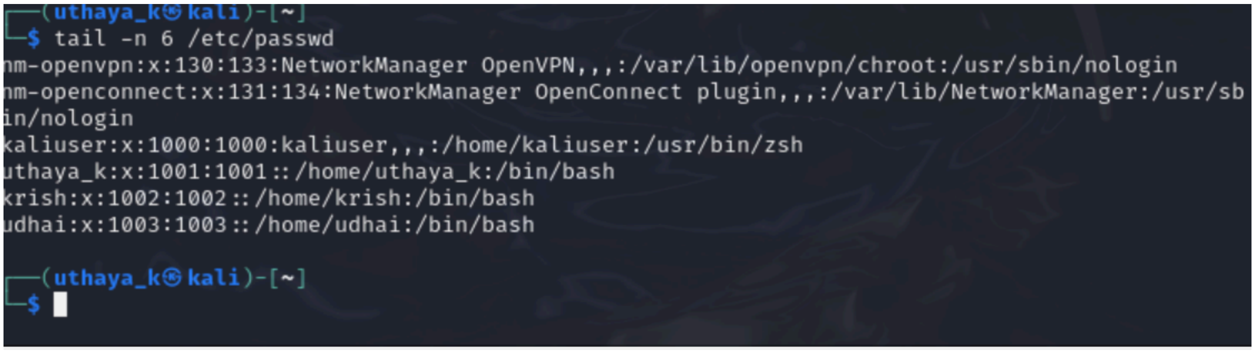
```
┌──(kaliuser㉿kali)-[~]
└─$ su - uthaya_k
Password:
┌──(uthaya_k㉿kali)-[~]
└─$ 
```

Create additional users, all members of your group, change default shell to /bin/bash



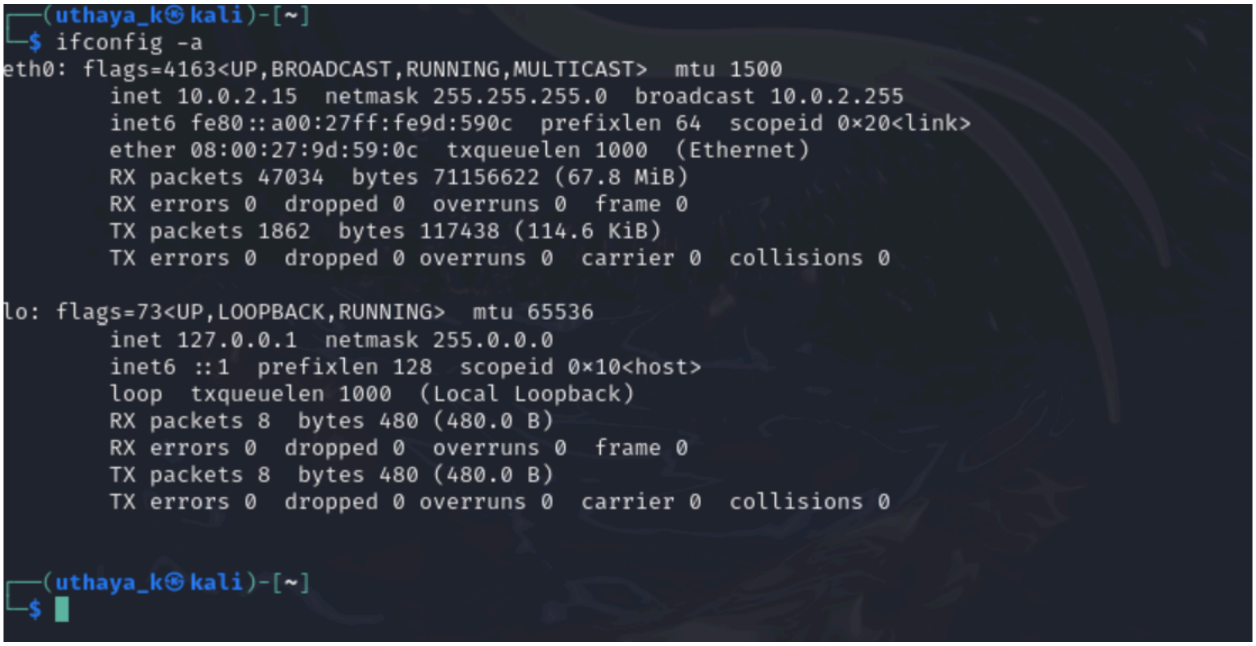
```
──(kaliuser㉿kali)-[~]
─$ su - uthaya_k
assword:
──(uthaya_k㉿kali)-[~]
─$ sudo useradd -m -s /bin/bash krish
sudo] password for uthaya_k:

──(uthaya_k㉿kali)-[~]
─$ sudo useradd -m -s /bin/bash udhai

──(uthaya_k㉿kali)-[~]
─$ 
```

Using tail command, give me the output of last 6 lines of /etc/passwd file



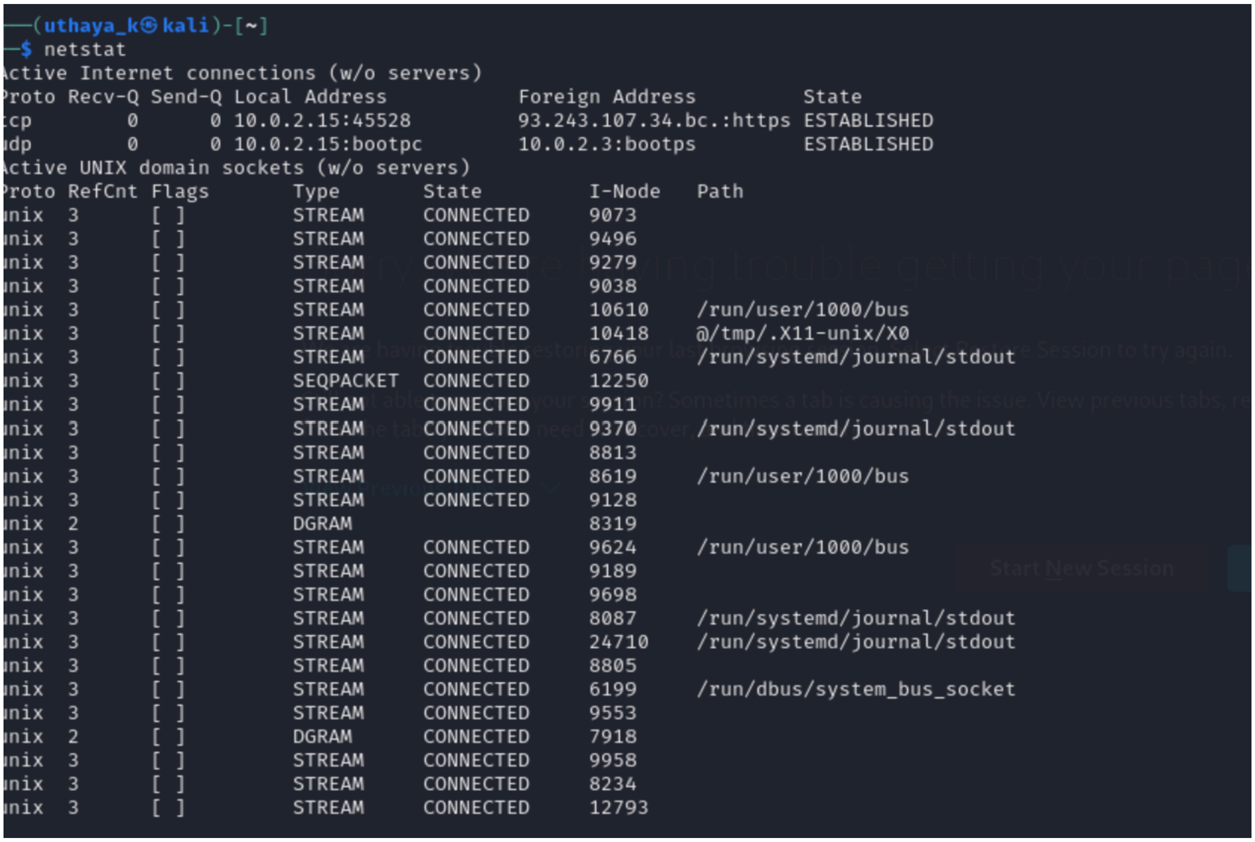
```
┌──(uthaya_k㉿kali)-[~]
└─$ tail -n 6 /etc/passwd
nm-openvpn:x:130:133:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
nm-openconnect:x:131:134:NetworkManager OpenConnect plugin,,,:/var/lib/NetworkManager:/usr/sb
in/nologin
kaliuser:x:1000:1000:kaliuser,,,:/home/kaliuser:/usr/bin/zsh
uthaya_k:x:1001:1001::/home/uthaya_k:/bin/bash
krish:x:1002:1002::/home/krish:/bin/bash
udhai:x:1003:1003::/home/udhai:/bin/bash

┌──(uthaya_k㉿kali)-[~]
└─$ 
```

Get the information of network interfaces of that VM



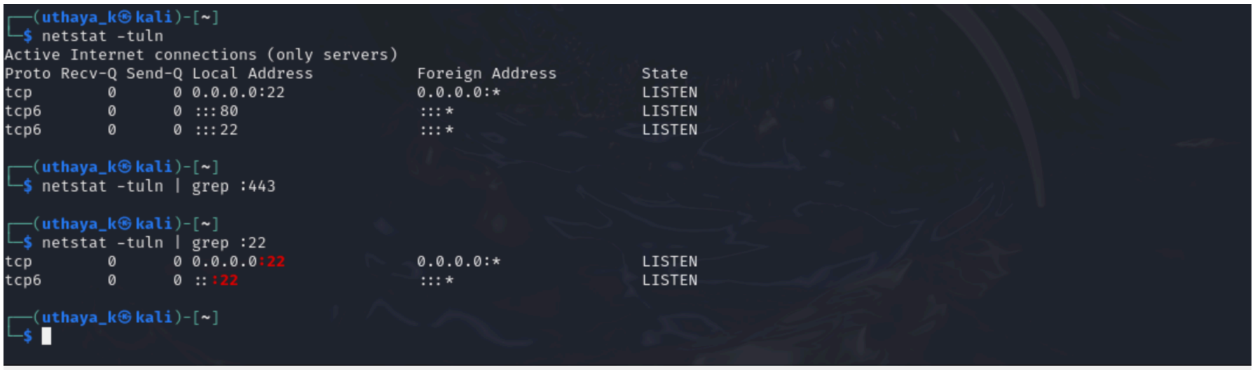
```
┌──(uthaya_k㉿kali)-[~]
└─$ ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe9d:590c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:9d:59:0c  txqueuelen 1000  (Ethernet)
        RX packets 47034  bytes 71156622 (67.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1862  bytes 117438 (114.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(uthaya_k㉿kali)-[~]
└─$ 
```

Get the information of all the listening port in that VM



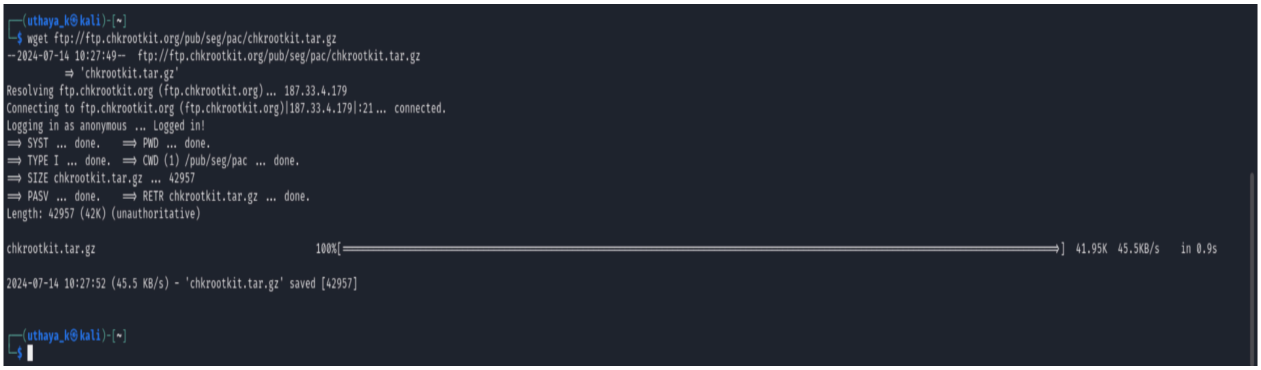
```
─(uthaya_k㉿kali)-[~]
─$ netstat
ctive Internet connections (w/o servers)
roto Recv-Q Send-Q Local Address           Foreign Address         State
cp         0      0 10.0.2.15:45528         93.243.107.34.bc.:https ESTABLISHED
dp         0      0 10.0.2.15:bootpc        10.0.2.3:bootps         ESTABLISHED
ctive UNIX domain sockets (w/o servers)
roto RefCnt Flags       Type       State         I-Node   Path
nix  3      [ ]         STREAM     CONNECTED     9073
nix  3      [ ]         STREAM     CONNECTED     9496
nix  3      [ ]         STREAM     CONNECTED     9279
nix  3      [ ]         STREAM     CONNECTED     9038
nix  3      [ ]         STREAM     CONNECTED     10610    /run/user/1000/bus
nix  3      [ ]         STREAM     CONNECTED     10418    @/tmp/.X11-unix/X0
nix  3      [ ]         STREAM     CONNECTED     6766     /run/systemd/journal/stdout
nix  3      [ ]         SEQPACKET  CONNECTED     12250
nix  3      [ ]         STREAM     CONNECTED     9911
nix  3      [ ]         STREAM     CONNECTED     9370     /run/systemd/journal/stdout
nix  3      [ ]         STREAM     CONNECTED     8813
nix  3      [ ]         STREAM     CONNECTED     8619     /run/user/1000/bus
nix  3      [ ]         STREAM     CONNECTED     9128
nix  2      [ ]         DGRAM                    8319
nix  3      [ ]         STREAM     CONNECTED     9624     /run/user/1000/bus
nix  3      [ ]         STREAM     CONNECTED     9189
nix  3      [ ]         STREAM     CONNECTED     9698
nix  3      [ ]         STREAM     CONNECTED     8087     /run/systemd/journal/stdout
nix  3      [ ]         STREAM     CONNECTED     24710    /run/systemd/journal/stdout
nix  3      [ ]         STREAM     CONNECTED     8805
nix  3      [ ]         STREAM     CONNECTED     6199     /run/dbus/system_bus_socket
nix  3      [ ]         STREAM     CONNECTED     9553
nix  2      [ ]         DGRAM      CONNECTED     7918
nix  3      [ ]         STREAM     CONNECTED     9958
nix  3      [ ]         STREAM     CONNECTED     8234
nix  3      [ ]         STREAM     CONNECTED     12793
```

Pass all the output of ‘netstat’ command to pipe and then grep the port 443 from the results



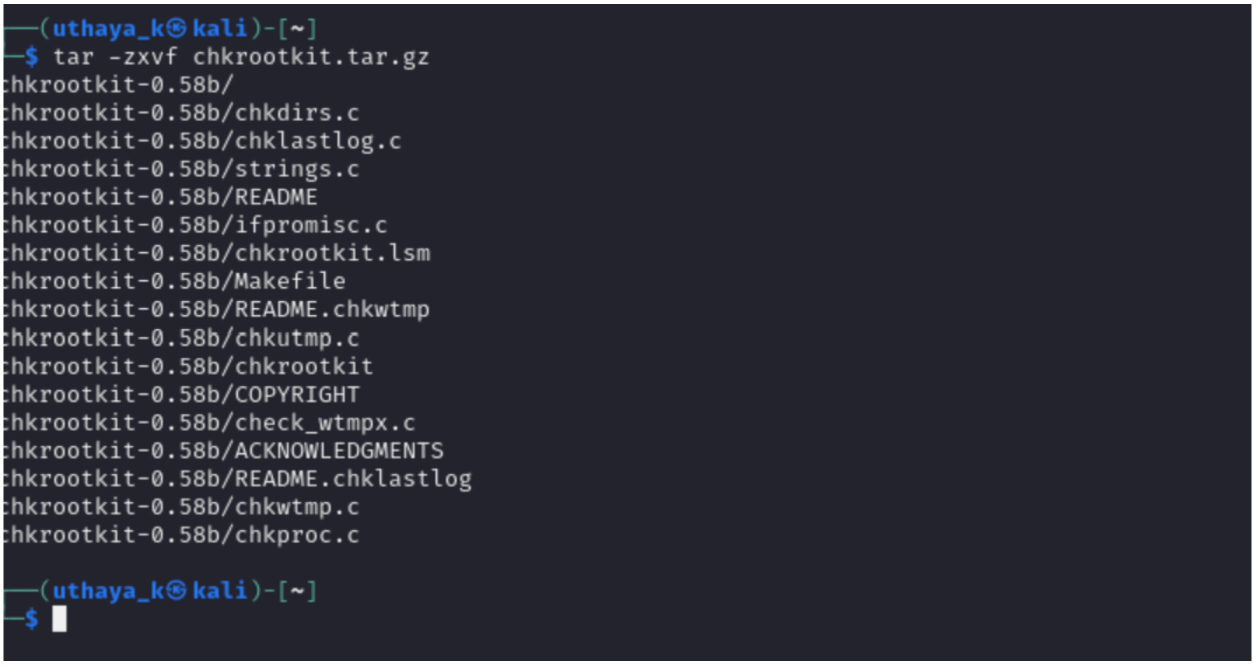
```
┌──(uthaya_k㉿kali)-[~]
└─$ netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN

┌──(uthaya_k㉿kali)-[~]
└─$ netstat -tuln | grep :443

┌──(uthaya_k㉿kali)-[~]
└─$ netstat -tuln | grep :22
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN

┌──(uthaya_k㉿kali)-[~]
└─$ █
```

Download Chkrootkit

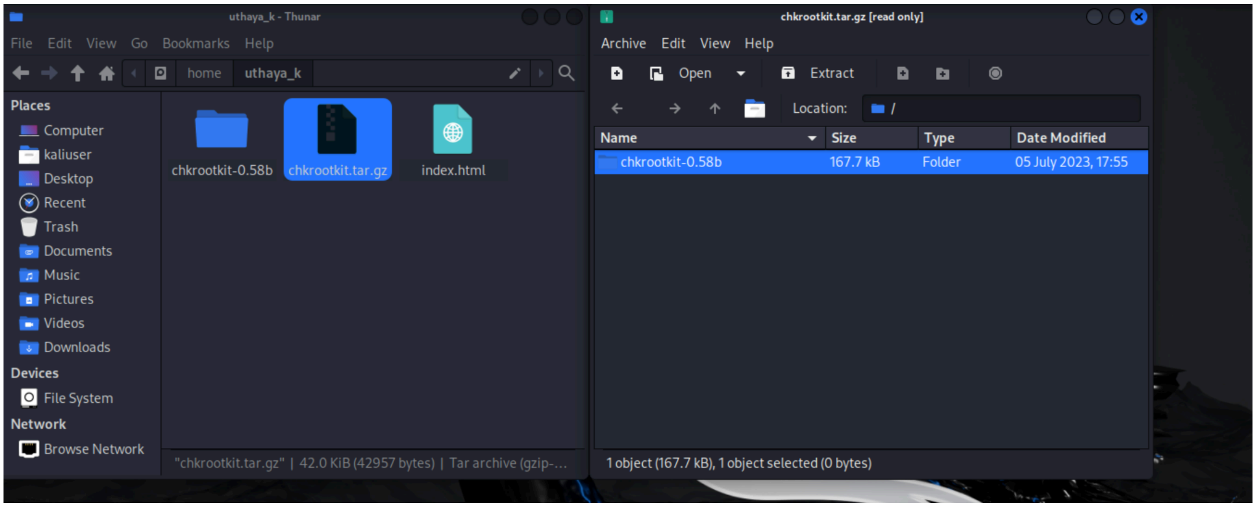


```
┌──(uthaya_k㉿kali)-[~]
└─$ wget ftp://ftp.chkrootkit.org/pub/seg/pac/chkrootkit.tar.gz
--2024-07-14 10:27:49--  ftp://ftp.chkrootkit.org/pub/seg/pac/chkrootkit.tar.gz
           ⇒ 'chkrootkit.tar.gz'
Resolving ftp.chkrootkit.org (ftp.chkrootkit.org)... 187.33.4.179
Connecting to ftp.chkrootkit.org (ftp.chkrootkit.org)|187.33.4.179|:21... connected.
Logging in as anonymous ... Logged in!
⟹ SYST ... done.    ⟹ PWD ... done.
⟹ TYPE I ... done.  ⟹ CWD (1) /pub/seg/pac ... done.
⟹ SIZE chkrootkit.tar.gz ... 42957
⟹ PASV ... done.    ⟹ RETR chkrootkit.tar.gz ... done.
Length: 42957 (42K) (unauthoritative)

chkrootkit.tar.gz                    100%[===================================================>]  41.95K  45.5KB/s    in 0.9s

2024-07-14 10:27:52 (45.5 KB/s) - 'chkrootkit.tar.gz' saved [42957]


┌──(uthaya_k㉿kali)-[~]
└─$ 
```

untar the file



```
┌──(uthaya_k㉿kali)-[~]
└─$ tar -zxvf chkrootkit.tar.gz
chkrootkit-0.58b/
chkrootkit-0.58b/chkdirs.c
chkrootkit-0.58b/chklastlog.c
chkrootkit-0.58b/strings.c
chkrootkit-0.58b/README
chkrootkit-0.58b/ifpromisc.c
chkrootkit-0.58b/chkrootkit.lsm
chkrootkit-0.58b/Makefile
chkrootkit-0.58b/README.chkwtmp
chkrootkit-0.58b/chkutmp.c
chkrootkit-0.58b/chkrootkit
chkrootkit-0.58b/COPYRIGHT
chkrootkit-0.58b/check_wtmpx.c
chkrootkit-0.58b/ACKNOWLEDGMENTS
chkrootkit-0.58b/README.chklastlog
chkrootkit-0.58b/chkwtmp.c
chkrootkit-0.58b/chkproc.c

┌──(uthaya_k㉿kali)-[~]
└─$ 
```

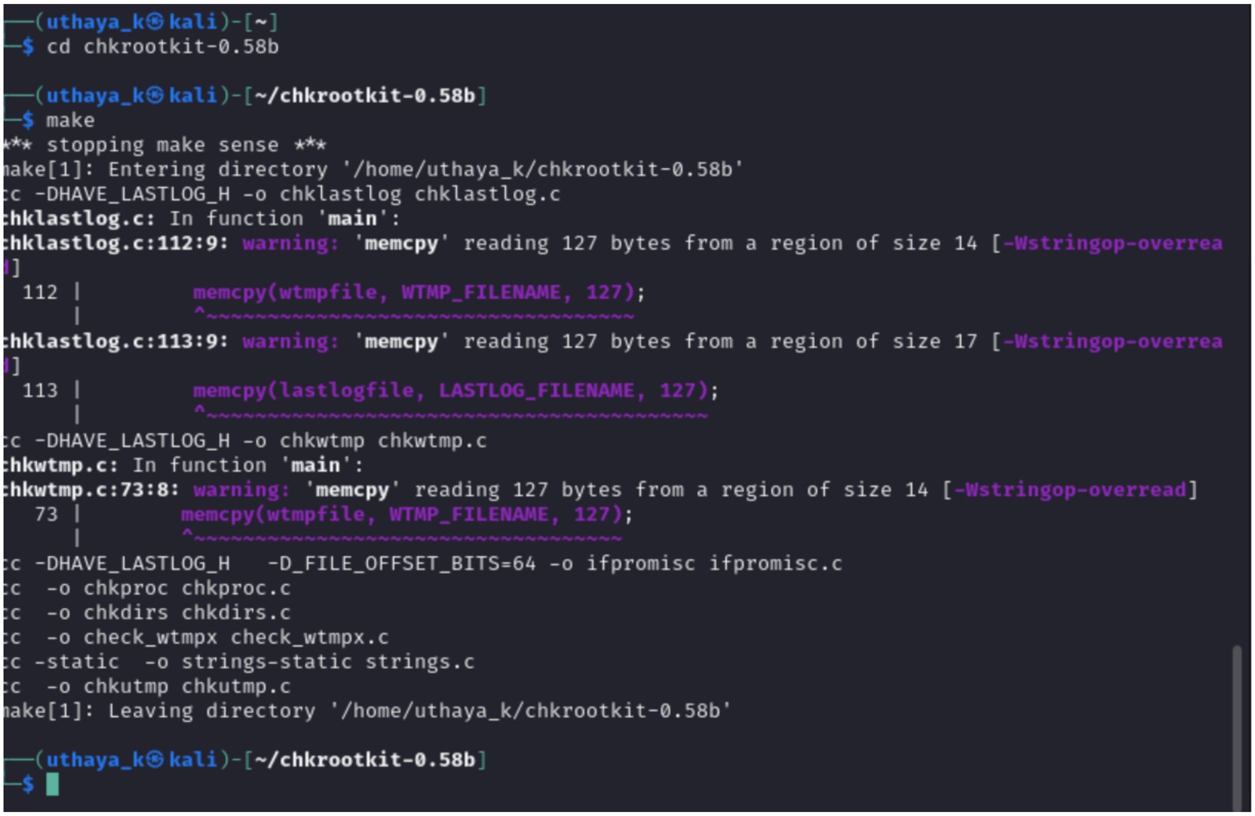


"chkrootkit.tar.gz" | 42.0 KiB (42957 bytes) | Tar archive (gzip-…

Name	Size	Type	Date Modified
chkrootkit-0.58b	167.7 kB	Folder	05 July 2023, 17:55

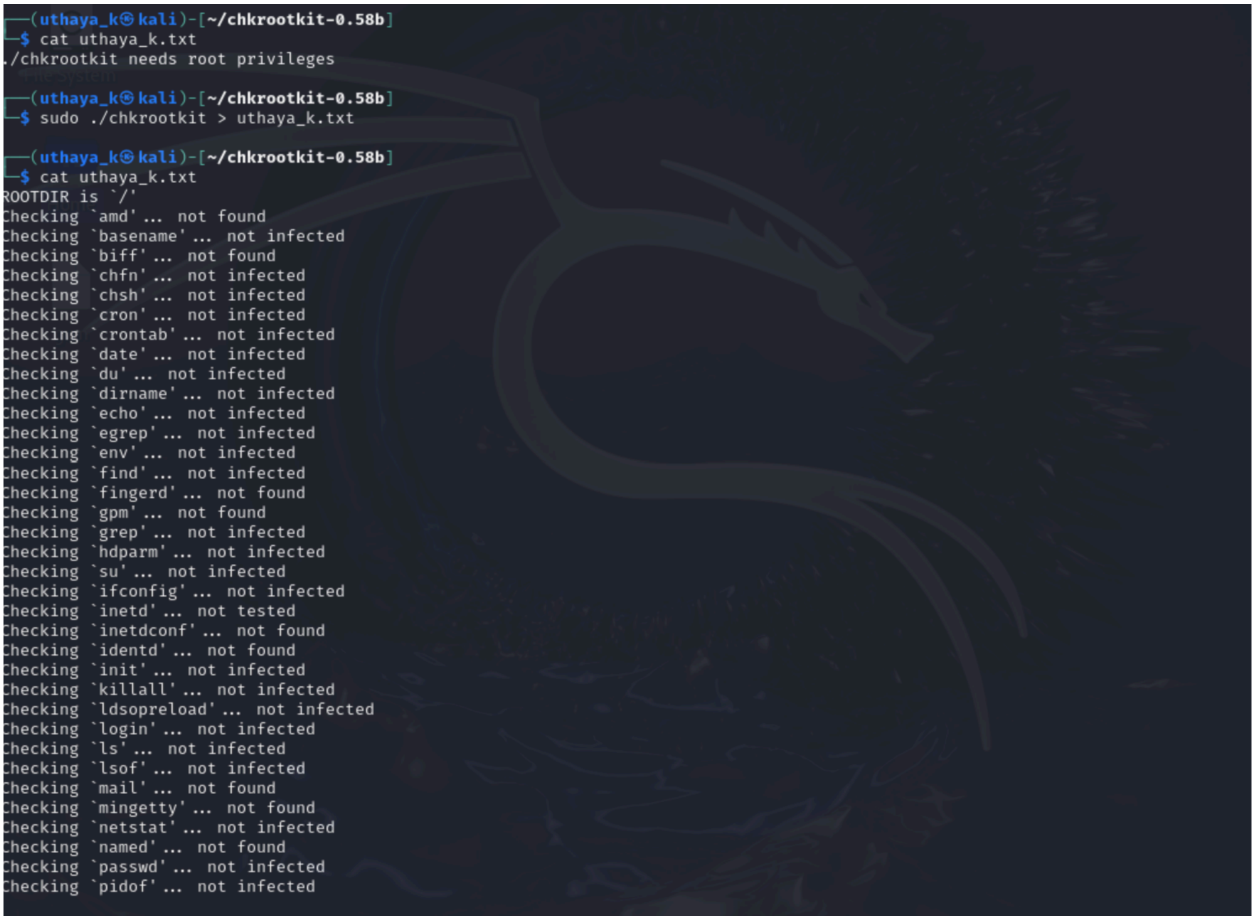
1 object (167.7 kB), 1 object selected (0 bytes)

'Make' the file and run it – just run the basic 'chkrootkit' script

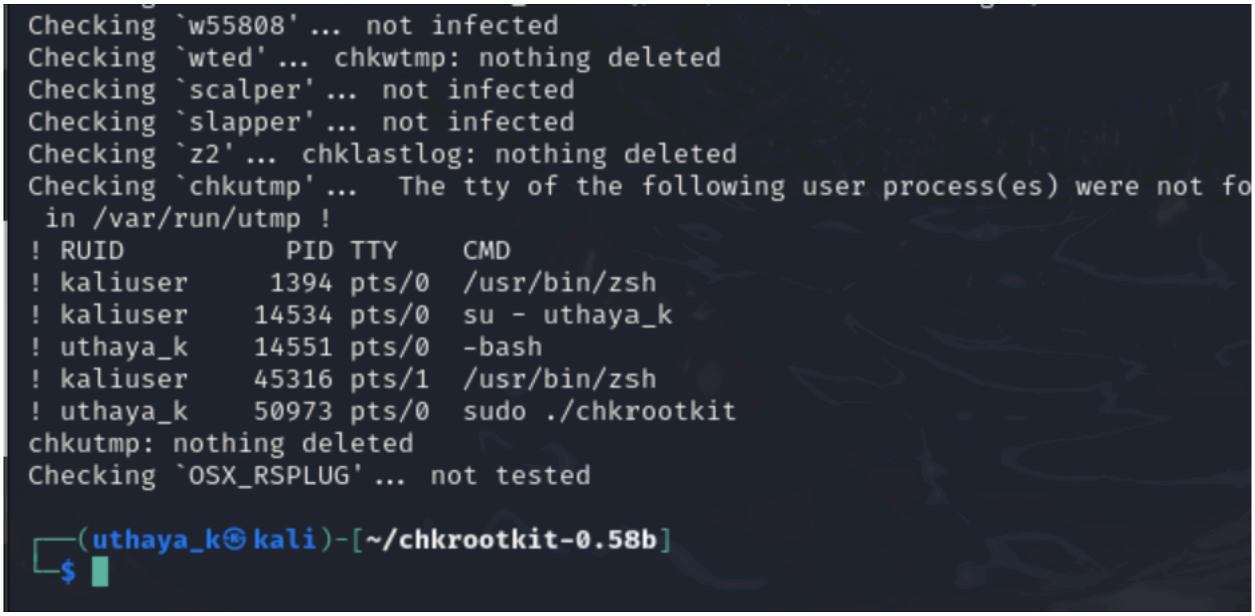


```
──(uthaya_k㉿kali)-[~]
─$ cd chkrootkit-0.58b

──(uthaya_k㉿kali)-[~/chkrootkit-0.58b]
─$ make
** stopping make sense ***
ake[1]: Entering directory '/home/uthaya_k/chkrootkit-0.58b'
c -DHAVE_LASTLOG_H -o chklastlog chklastlog.c
hklastlog.c: In function 'main':
hklastlog.c:112:9: warning: 'memcpy' reading 127 bytes from a region of size 14 [-Wstringop-overrea
]
 112 |         memcpy(wtmpfile, WTMP_FILENAME, 127);
     |         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
hklastlog.c:113:9: warning: 'memcpy' reading 127 bytes from a region of size 17 [-Wstringop-overrea
]
 113 |         memcpy(lastlogfile, LASTLOG_FILENAME, 127);
     |         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
c -DHAVE_LASTLOG_H -o chkwtmp chkwtmp.c
hkwtmp.c: In function 'main':
hkwtmp.c:73:8: warning: 'memcpy' reading 127 bytes from a region of size 14 [-Wstringop-overread]
  73 |        memcpy(wtmpfile, WTMP_FILENAME, 127);
     |        ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
c -DHAVE_LASTLOG_H   -D_FILE_OFFSET_BITS=64 -o ifpromisc ifpromisc.c
c  -o chkproc chkproc.c
c  -o chkdirs chkdirs.c
c  -o check_wtmpx check_wtmpx.c
c -static  -o strings-static strings.c
c  -o chkutmp chkutmp.c
ake[1]: Leaving directory '/home/uthaya_k/chkrootkit-0.58b'

──(uthaya_k㉿kali)-[~/chkrootkit-0.58b]
─$ 
```

send the output to a text file. Text file name should be your name.txt



```
┌──(uthaya_k㉿kali)-[~/chkrootkit-0.58b]
└─$ cat uthaya_k.txt
./chkrootkit needs root privileges

┌──(uthaya_k㉿kali)-[~/chkrootkit-0.58b]
└─$ sudo ./chkrootkit > uthaya_k.txt

┌──(uthaya_k㉿kali)-[~/chkrootkit-0.58b]
└─$ cat uthaya_k.txt
ROOTDIR is `/'
Checking `amd'... not found
Checking `basename'... not infected
Checking `biff'... not found
Checking `chfn'... not infected
Checking `chsh'... not infected
Checking `cron'... not infected
Checking `crontab'... not infected
Checking `date'... not infected
Checking `du'... not infected
Checking `dirname'... not infected
Checking `echo'... not infected
Checking `egrep'... not infected
Checking `env'... not infected
Checking `find'... not infected
Checking `fingerd'... not found
Checking `gpm'... not found
Checking `grep'... not infected
Checking `hdparm'... not infected
Checking `su'... not infected
Checking `ifconfig'... not infected
Checking `inetd'... not tested
Checking `inetdconf'... not found
Checking `identd'... not found
Checking `init'... not infected
Checking `killall'... not infected
Checking `ldsopreload'... not infected
Checking `login'... not infected
Checking `ls'... not infected
Checking `lsof'... not infected
Checking `mail'... not found
Checking `mingetty'... not found
Checking `netstat'... not infected
Checking `named'... not found
Checking `passwd'... not infected
Checking `pidof'... not infected
```



```
Checking `w55808'... not infected
Checking `wted'... chkwtmp: nothing deleted
Checking `scalper'... not infected
Checking `slapper'... not infected
Checking `z2'... chklastlog: nothing deleted
Checking `chkutmp'...  The tty of the following user process(es) were not fo
 in /var/run/utmp !
! RUID          PID TTY    CMD
! kaliuser     1394 pts/0  /usr/bin/zsh
! kaliuser    14534 pts/0  su - uthaya_k
! uthaya_k    14551 pts/0  -bash
! kaliuser    45316 pts/1  /usr/bin/zsh
! uthaya_k    50973 pts/0  sudo ./chkrootkit
chkutmp: nothing deleted
Checking `OSX_RSPLUG'... not tested

┌──(uthaya_k㉿kali)-[~/chkrootkit-0.58b]
└─$ 
```

Task 2 – Research on Windows Forensic Artifacts

1. Network Activity - System Resource Usage Monitor (SRUM)

Description: The System Resource Usage Monitor (SRUM) in Windows tracks and logs resource usage metrics, including network activity by applications and processes. It provides details such as data sent/received, network connections, and usage patterns over time.

Location:

- **Windows 10**: `%SystemRoot%\System32\sru\sru.db`
- **Windows 8**: `%SystemRoot%\System32\sru\sru.db`
- **Windows 7**: `%SystemRoot%\System32\sru\sru.db`
- **Windows XP**: Not applicable (SRUM is not available in Windows XP)

Comparison to Linux: In Linux, similar network activity monitoring can be achieved using tools like `iftop`, `nload`, and `vnstat`. These tools provide real-time and historical network usage statistics but typically do not log detailed application-specific network activity by default.

Value in Forensic Investigation: SRUM data is invaluable for forensic investigations as it:

- **Identifies Malicious Activity**: Helps detect unauthorized network connections or data exfiltration attempts by malicious software.
- **User Behavior Analysis**: Reveals patterns of network usage by users or applications, aiding in understanding intent and establishing timelines.
- **Evidence in Legal Proceedings**: Provides concrete evidence of network activities that can be used in legal proceedings, such as proving unauthorized access or data breaches.

2. File/Folder Opening - Shell Bags

Description: Shell Bags in Windows store metadata and viewing preferences for folders accessed by users. They maintain details such as folder size, view settings, and the last accessed timestamp.

Location:

- **Windows 10, 8, 7**: Registry keys under `HKEY_CURRENT_USER\Software\Microsoft\Windows\Shell\Bags` and `HKEY_CURRENT_USER\Software\Microsoft\Windows\Shell\BagMRU`.

Comparison to Linux: In Linux, similar metadata about folder access can be found in `recently-used.xbel` files in user directories (`~/.local/share/recently-used.xbel`). These files maintain a history of recently accessed files and folders.

Value in Forensic Investigation: Shell Bags are valuable as they:

- **Reconstruct User Activity**: Provide a detailed timeline of folder access, aiding in reconstructing user navigation paths and activities.
- **Evidence of Intent**: Help establish user intent or actions taken, such as accessing specific folders containing sensitive information.
- **Malware Analysis**: Identify folders accessed by malware during an attack, aiding in understanding its impact and propagation.

3. Account Usage - Last Password Usage

Description: Last Password Usage records the timestamp of the last successful authentication using a user's password. It helps track account activity and detect unauthorized access attempts.

Location:

- **Windows 10, 8, 7**: Event Logs (`Security` log) containing Event IDs such as 4624 (Successful Logon) and 4776 (Authentication Ticket Granted).

Comparison to Linux: In Linux, authentication events are logged in `/var/log/auth.log` or `/var/log/secure`. These logs record successful and failed authentication attempts, but may not explicitly track the last password usage timestamp.

Value in Forensic Investigation: Last Password Usage is crucial as it:

- **Audits Account Activity**: Logs timestamps of password usage, aiding in auditing user account access and monitoring for unauthorized logins.
- **Forensic Timeline**: Establishes a timeline of user authentication events, helping investigators trace user actions and interactions with the system.
- **Detection of Compromised Accounts**: Alerts to unauthorized usage of credentials, identifying potential insider threats or external breaches.