Linux Forensics Bits

Files

List files in a dir by time of modification

(note, this will not include deeply nested dirs)

```
> ls -lt
```

List most recently modified files

```
> find <start directory> -mtime -<days ago (int)> -ls
```

Other useful

```
stat
file
```

Check installed software against the package manager

(Warning: these yield many false positives and much noise)

On Ubuntu and Debian

```
> sudo apt install debsums
> sudo debsums_init
```

`> sudo debsums -ca` (will show all modified apt-installed files)

On redhat based(CentOS/Oracle Linux/Fedora)

```
> sudo rpm -Va
```

Code	Meaning
S	Different size
M	Different file mode
5	Different md5sum
T	Time of modification is more recent than installation
L	Link mismatch (probably noise)
U	Different ownership
G	Different group ownership
c	A config file
d	Documentation file
g	Misc. file type (probably noise, includes log files and such)

Processes

View running processes

```
> ps aux
```

View running processes as tree

```
> ps faux
```

View all processes for a user

```
> ps -u <username>
```

Users

Who’s on your system

```
> w
```

Latest Logins

```
> last
```

System Config

View SystemD timers

```
> systemctl list-timers --all
```

See Crontab jobs

- One of the fastest places to check is in `/var/log/cron`
- There are per-user crontabs and system-wide crontabs in `/etc`

```
> ls /etc | grep cron
> crontab -u <user> -l
```

eBPF

eBPF is a way to run specialized code inside of the Kernel. To work with eBPF, install the bpfcc-tools package

```
> apt install bpfcc-tools # Ubuntu and Debian
```

List active bpf programs (one or other of these commands)

```
> bpflist-bpfcc # as root
> bpflist
```

SystemD will probably have a couple, there may be others

eBPF Tracing

Besides looking for suspicious eBPF activity, you can use eBPF to monitor other programs. For this, use the bpftrace package. You will also need your Linux headers.

The bpf tracing toolkit comes with a number of useful example programs.

- tcpaccept.bt
- tcpconnect.bt

Networking

Open ports

```
> ss -tulpn
```

Active connections

```
> ss -tuapw
```

Watch for particular connections

- Use ngrep, like grep for network traffic
- Similar to tcpdump
- Probably will need to install it (`apt install ngrep`, `yum install ngrep`)

```
> sudo ngrep -d any dst port 22 # watch for incoming ssh connections
> sudo ngrep -d any port 53 # watch for DNS
> sudo ngrep -d any host 1.2.3.4 # watch for traffic from ip 1.2.3.4
```

Show open network connections by file descriptor (very useful to see what is active):
(install lsof with “lsof” package on Centos, Ubuntu)

```
> lsof -nPi
```

Dump netfilter configuration (watch out for rules in other tables)

```
> iptables -L -n -v
> iptables -L -n -v -t nat
> iptables -L -n -v -t mangle
> iptables -L -n -v -t raw
> iptables -L -n -v -t security
```

Also do for ip6tables ^
Dump Nftables

```
> nft list ruleset
```

See the nftables guides for more...

Routing Configuration:

```
> ip rule show
> ip route show table all
> ip link show
> ip addr show
> ip netns list # note that this only shows named netns, so containers may
not
                   appear
```

Iproute2 cheatsheet: https://baturin.org/docs/iproute2/

Verify DNS configuration in /etc/resolv.conf

Crontab

Check /var/spool/cron/crontabs for user crontabs
Check /etc/crontab for system cron