ncae checklist

Must-do

- □ Delete malicious tools + stop cron/crond/cronie service
    - □ netcat / nc / ncat
    - □ ansible
    - □ cron/crond/cronie
    - □ at
- □ Check package integrity of all system binaries
    - □ Debian: `dpkg --verify`
        - □ `??5?????? - Failed md5 check`
        - □ `apt-install --reinstall <package> for ??5???? binaries`
    - □ CentOS: `rpm -Va --nomtimeg`
    - □ `Check binaries in $PATH && aliases in .bashrc`
- □ **Change logins, remove unnecessary users**
    - □ **/usr/bin/passwd and update-passwd**
    - □ **Verify /usr/sbin/nologin isn't just bash**
    - □ **SSH: move authorized_keys to unauthorized_keys**
- □ Static network config (i.e. assign ip addresses and configure router)
    - □ To get internet access, set the machine's DNS server to 8.8.8.8 -- typically this is set in /etc/resolv.conf
- □ UFW
    - □ Block all INCOMING connections that aren't score check services
        - □ sudo ufw default deny incoming
        - □ sudo ufw allow <port_number>
        - □ sudo ufw enable
        - □ sudo ufw status verbose
- □ Opensnitch (no rules at first) **Version: 1.5.2**
    - □ Kill established connections (look at netstat -plunet for them)
    - □ Set UI to listen for incoming connection
        - □ opensnitch-ui --socket [::]:50051
    - □ Enable daemon to forward events
        - □ In /etc/opensnitchd/default-config.json
        - □ Change address value
        - □ "Address":"**{IP_ADDR}:50051**"
        - □ Make a local back up of the rules file and place it in /opt/osrules or something
- □ Services - bring the machine online

- □ Check if there were any important cronjobs (reinstall cron if so)

  ```
  for user in $(cut -f1 -d: /etc/passwd); do crontab -u $user
  -l; done; for file in /etc/cron.*/*; do echo $file; cat
  $file; done
  ```

- □ Backups
  - □ /etc/
  - □ /var/log if it fits
  - □ /var/www or /var/www-html
  - □ Run this command on the machine whose files you want to back up externally to a backup machine:

    ```
     rsync -av -e ssh <path in local machine> <backup
    machine user>@<backup machine
    IP>:/opt/backups/<machine_name_directory_name>
    ```

Threat Hunting

- □ find / -type f -iname "*redteam*" -o -type d -iname "*redteam*" 2> /dev/null
    - □ Find anything with “redteam” on the filesystem

- □ ps aux | awk '$11 !~/^\/(sbin|bin|usr|var|lib|sys|proc|dev|tmp|run|root|home|etc)\// {print}'
    - □ Finds processes running in uncommon directories
    - □ Command goes in one line; No space between ‘!~’ and ‘/^’

- □ find / -type f -mtime **{DAYS}** 2>/dev/null
    - □ Finds all files created within {DAYS} days.

- □ **Debian:** comm -13 <(cat /var/lib/dpkg/info/*.list | sort -u) <(find / | sort -u)
    - □ Finds packages that are not installed using dpkg

- □ **CentOS:** comm -13 <(rpm -qla | sort -u) <(find / | sort -u)
    - □ Finds packages that are not installed using

- □ Linpeas/winpeas
- □ Ensure all versions of netcat (nc/ncat/netcat) are deleted
- □ Check /etc/hosts for malicious domains
- □ Sysdig to monitor stuff
- □ Install ClamAV
    - □ Have to wait a bit after install for signature db to populate
    - □ Likely don’t do a full filesystem scan, prob just if you’re suspicious of a particular file or certain directories
- □ In case of messed up aliases: Run `sysdig -c spy_users` in case aliases are messed up (see exactly what users are running)

Chatgpt script for killing shells from a specific IP

```
#!/bin/bash

# Replace <target_ip> with the specific IP address you want to target
target_ip="X.X.X.X"

# Find all user sessions associated with the target IP and terminate them
for session_info in $(w -h | grep "$target_ip" | awk '{print $1 ":" $2}')
do
    user=$(echo $session_info | cut -d: -f1)
    tty=$(echo $session_info | cut -d: -f2)

    echo "Terminating session for user '$user' on TTY '$tty' coming from IP '$target_ip'"
    pkill -9 -t $tty
done

echo "All sessions for IP $target_ip have been terminated."
```