LINUX 5-MINUTE PLAN

- **Step 1: Change every user's password**:
  - HISTFILE=/dev/null awk -F: '{print $1":CryptoDATA001$$"}' /etc/passwd | sudo chpasswd

- **Step 2: Reset PAM configuration**:
  - sudo pam-auth-update --force

- **Step 3: Fix SSH authentication**:
  - /etc/ssh/sshd_config: “UsePAM yes” -> “UsePAM no” (Line 80ish)
  - sudo systemctl restart sshd
  - cat /etc/apt/sources.list OR ls /etc/apt/sources.list.d/
    - Check for naughty repos real quick and remove any standouts

- **Step 4: Configure iptables:**

❖ **iptables -F**	Flush all chains
❖ **iptables -X**	Delete all chains
❖ **iptables -Z**	Reset packet & byte counter in chains
❖ **iptables -L -nv --line-numbers**	Show ruleset

- **PACKAGE INSTALL FOR PERSISTENT RULES ON REBOOT (DO FIRST):**
  - **sudo apt install iptables-persistent**
  - **sudo dnf install iptables-services (RED HAT ONLY)**

CUSTOM RULESET (FOLLOW IN ORDER) - **only need ‘sudo’ if not root**	
sudo iptables -A INPUT -j ACCEPT	**Create allow any-any rules while creating custom ruleset (prevents loss in scoring)**
sudo iptables -A OUTPUT -j ACCEPT	
sudo iptables -P INPUT DROP	**Change the default chain rules to drop all packets if none of the rules match**
sudo iptables -P FORWARD DROP	
sudo iptables -P OUTPUT DROP	
sudo iptables -A INPUT -i lo -j ACCEPT	**Allow traffic to and from the loopback interface**
sudo iptables -A OUTPUT -o lo -j ACCEPT	
sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT	**Uphold current connections with the system (will also prevent a lockout from ssh when removing the first rule)**
sudo iptables -A OUTPUT -m conntrack --ctstate ESTABLIHSED,RELATED -j ACCEPT	

sudo iptables -A INPUT -p tcp --dport 22 -s <VPN subnet> -m conntrack --ctstate NEW -j ACCEPT	**Allows all ssh connections in and out of the system; can add IPs in here (syntax in docs)**
sudo iptables -A OUTPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW -j ACCEPT	**Allow HTTP/HTTPS connections to be created outbound**
sudo iptables -A OUTPUT -p udp --dport 53 -s 1.1.1.1 -j ACCEPT	**Allow outbound DNS requests only to public DNS server (or the Windows AD DNS, but it might be poisoned)**
sudo iptables -A OUTPUT -p tcp --dport 53 -s 1.1.1.1 -j ACCEPT	
sudo iptables -A INPUT -s <scoring-ip> -m conntrack --ctstate NEW -j ACCEPT	**Allow the scoring engine to reach the box directly both inbound and outbound**
sudo iptables -A OUTPUT -d <scoring-ip> -m conntrack --ctstate NEW -j ACCEPT	
Consult iptables documentation for port specific rules	**Get creative—add rules for certain protocols/other IPs (ping, specific services, ftp, ntp server traffic, etc.)**
sudo iptables -A INPUT -m limit --limit 20/hr -j LOG --log-prefix “[netfilter] INPUT:DROP: ” --log-level 7	**Any packet that is going to be dropped will be added to the syslog with the specified prefix**
sudo iptables -A OUTPUT -m limit --limit 20/hr -j LOG --log-prefix “[netfilter] OUTPUT:DROP: ” --log-level 7	
sudo iptables -D INPUT 1	**Remove allow any-any rules when ruleset is *mostly* complete**
sudo iptables -D OUTPUT 1	

- **SAVE RULES WHEN DONE!!!**
  - **sudo sh -c “iptables-save > /etc/iptables/rules.v4”**
- **Check other tables real quick**
  - **sudo iptables -L -nv <mangle, nat, raw, security>**
- **View the logs if necessary**
  - **journalctl -k | grep ‘\[netfilter\]’**

- **EXTRA INFO FOR RED HAT (Fedora/Rocky/CentOS)**
  - **sudo systemctl stop firewalld**
  - **sudo systemctl disable firewalld**
  - **sudo systemctl start iptables**
  - **sudo systemctl enable iptables**

- **Step 5: Backup EVERYTHING:**

  - **INITIALIZATION**
    - **sudo apt install git**
    - **cd /**
    - **git init .**
    - **git add /etc/ssh/sshd_config**
    - **git add /usr/bin/ls**
    - **git add /usr/bin/cat**
    - **git add /usr/bin/ssh**
    - **git add /usr/lib/ssh**
    - **git add /etc/ssh**
    - **git add /usr/bin/echo**
    - **git add /path/to/important/file (do this for every file you want tracked)**
    - **git commit (saves changes to a specific "commit")**

  - **MONITORING AND MAINTENECE**
    - **git diff (shows difference between last commit and current state of files)**
    - **git log (lists commits)**
    - **git checkout <commit_id> --force (reverts to a specific commit)**
    - **git revert (reverts to last commit)**
    - **git status (see what commit you're on)**

POST 5-MINUTES

- **Step 1: Update packages:**
  - **CHECK REPOS IN DEBIAN (Debian/Ubuntu/openSUSE)**
    - cat /etc/apt/sources.list
    - ls /etc/apt/sources.list.d/
  - **CHECK REPOS IN RED HAT (Fedora/Rocky/CentOS)**
    - ls /etc/yum.repos.d
    - cat /etc/yum.repos.d/*.repo
  - **UPDATE THE SYSTEM PACKAGES**
    - sudo apt clean
    - sudo apt update && sudo apt upgrade
    - sudo dnf update && sudo dnf upgrade
      - ♦ Establish a second terminal connection while running this

- **Step 2: Lock out bad accounts kill current sessions:**
  - cat /etc/passwd
  - cat /etc/group
    - Check all users and/or groups currently on the system
    - Can 'chattr -i /etc/groups' to lock the config
  - sudo usermod -L && sudo usermod -s /usr/sbin/nologin <user>
    - Disable the specified account and remove its shell access
  - sudo truncate -s 0 ~/.ssh/authorized_keys
    - Command to remove naughty keys... (if not already done)
  - who -u and lastlogin and netstat -atn | grep ':22'
    - Check logins from users and use pkill/kill to terminate them
    - pkill -9 -t {term-name} or kill -9 {PID}

- **Step 3: Check repeated processes (systemd timers and crontab):**
  - sudo systemctl list-timers --all
    - Insight into red team implants
  - crontab -e
    - Check to make sure there aren't any naughty jobs
    - Add the following lines (first checks if designated service is down and restarts it; second copies files to backup path)
  - * * * * * systemctl is-active --quiet <service_name> || systemctl restart <service_name>
  - * * * * * cp /etc/ssh/sshd_config <file_path>

- **Step 4: Look for signs of Red Team & manual threat hunting:**
  - Search major binaries/anywhere on the system for Red Team implants
  - Typically, they are larger files (MBs in size) with names slightly different from services, etc.
    - sudo find / -type f -executable -size +1M
      - Find executable files throughout the system >1MB
  - Utilize process listing/memory forensics to our advantage
    - ps -eLf
    - ps faux
    - dpkg -V
      - CHECKS INSTALLED PACKAGES FOR CHANGES OR FILE MODIFICATIONS
  - Check processes for network traffic
    - lsof -i tcp
    - lsof -i udp
    - lsof -c <process/service>
  - Take screenshots and hash malicious files

- **Step 5: Install Threat Hunting Tools:**
  - **Tripwire is a good example**
    - sudo apt install tripwire
      - tripwire -m i (initialize database) **use first
      - tripwire -m c (execute the check)
      - tripwire -m u
      - Add u and c to crontab separately
      - tripwire -m c -I (interactive check)
  - **Net-tools also helpful**
    - sudo apt install net-tools

- **Step 6: SSH into your servers and repeat…**
  - **RUN IT BACK** (Whoooo Yaaaaaa!!!!!)
  - To ssh into the server… (ssh <user>@<ip_addr>)

Additional Sources/things to think about:

https://github.com/RedefiningReality/Linux-Defence-Materials/tree/main