Readme - IDS

Thursday, October 23, 2025 10:20 PM

1. Prepare SD Card and OS

#Download and install Raspberry Pi OS Imager on other device w/ sd card

a. **Download and install** Raspberry Pi Imager on another device.
b. **Flash Raspberry Pi OS (Lite)** to the SD card.
c. **Pre-configure SSH and Wi-Fi** credentials in Imager before writing.

2. Check SD card health/speed:

#Anything sub 10 MB/s indicates poor SD card health

```
dd if=/dev/zero of=testfile bs=1M count=250 status=progress conv=fsync
```

3. Configure Swap Memory (for Zero 2 W or Low-RAM Models)

#On older models, like (Zero 2W): you'll need swap memory to install suricata/heavy apps

Create a 1 GB swap file

```
sudo fallocate -l 1G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile
```

#Confirm it's active

```
sudo swapon --show
```

4. Optional: Adjust Swap Settings

#Can modify size of default swapfile, due to swap being hard on SD & causing shortened lifespan

```
sudo nano /etc/dphys-swapfile
```

#Then to set default size to 512 with max size of 1GB

```
CONF_SWAPSIZE=512
CONF_SWAPFACTOR=2
CONF_MAXSWAP=1024
CONF_SWAPFILE=/var/swap
```

5. Optimize SD Card Longevity

#To reduce writes and ensure using RAM first to better protect SD card, in CLI:

```
sudo sysctl vm.swappiness=10
```

#To make this change permanent:

```
echo "vm.swappiness=10" | sudo tee -a /etc/sysctl.conf
```

6. System Updates

#Download updates/upgrades:

```
sudo apt-get update
sudo apt-get upgrade -y
```

7. Install and Configure Suricata

#Download and install suricata

```
sudo apt install suricata -y
```

#Enable Suricata on boot

```
sudo systemctl enable suricata
```

#Start Suricata to check successful installation

```
sudo systemctl start suricata
```

#Check status

```
sudo systemctl status suricata
```

8. Anticipating GUI in Future, Enable Eve JSON and Prepare Suricata Logs

#For a GUI option: several different ways. Local GUI on old model: EveBox. Can use Grafana/Prometheus later.

#Ensure EVE JSON logging is enabled in suricata.yaml

```
sudo nano /etc/suricata/suricata.yaml
```

#add eve-log below existing data found in 'outputs:'

```
outputs:
  - fast:
      enabled: yes
      filename: fast.log
      append: yes

  - eve-log:
      enabled: yes
      filetype: regular
      filename: /var/log/suricata/eve.json
      types:
        - alert
        - http
        - dns
        - tls
        - flow
```

#Save and close with ^X, y to save modified buffer, hit enter with filename

#This file is important as it dictates what rules and alerts are in effect.

9. Prepare Log Directory

#Ensure /var/log/suricata exists and is writeable

```
sudo mkdir -p /var/log/suricata
sudo chown root:root /var/log/suricata
```

10. Create Suricata User and Group

#Create user and group so logs are written safely without root. This creates group: suricata, user: suricata and gives full access to suricata, read/execute for group, none for other

```
sudo groupadd suricata
sudo useradd -r -g suricata -s /usr/sbin/nologin suricata
sudo mkdir -p /var/log/suricata
sudo chown suricata:suricata /var/log/suricata
sudo chmod 750 /var/log/suricata
```

11. Confirm Suricata user

#Check which user suricata is running under

ps aux | grep suricata

12. Restart Suricata

#Restart Suricata and logs should appear in: /var/log/suricata/eve.json which can be read by EveBox and other GUI tools

sudo systemctl restart suricata

#After setup, Suricata runs under non-root user for security

#confirm suricata can write logs:

sudo ls -l /var/log/suricata

You should see: eve.json, fast.log, stats.log, suricata.log

#/var/log/suricata *must* be writeable by suricata user prior to restart

13. Update Run Options

#I had to go into suricata.yaml and find Run Options, user and group were commented out

Remove # and change 'suri' to 'suricata'

#Restart suricata

sudo systemctl restart suricata

14. SECURITY ASPECTS & Live Deployment Preparation

#Confirm Suricata Logs Permissions

#IMPORTANT: Confirm all logs are now under suricata, not root:

#To view permissions:

sudo ls -l /var/log/suricata

#If there are still root privileges, convert all to be suricata group/user

sudo chown -R suricata:suricata /var/log/suricata

#check again to confirm

sudo ls -l /var/log/suricata

15. Ready for Live Capture

THIS MEANS WE ARE 'READY FOR LIVE CAPTURE'. Now it can be placed in the architecture with port mirroring to start capturing traffic.

- **Ethernet Setup:** Ensure the Pi's NIC is on the same VLAN or switch port as the mirrored traffic.
- **Port Mirroring:** Confirm the switch is actually mirroring the traffic you care about. Misconfigurations here are often why Suricata sees nothing.
- **Suricata Configuration:** In suricata.yaml, double-check af-packet or pfring capture interface settings point to the mirrored NIC.
- **Logs:** They will now start populating with network events; your external SSD plan can be applied later if needed.

IMPORTANT: Because of where this device sits, it's essentially a public-facing host even only running suricata. You will need to consider security here. Do not give the pi an IP address, implement UFW, SSH hardening and other security protocols.

16. Enable Firewall (UFW)

#Running a UFW to only allow critical services is crucial.

sudo apt install ufw

sudo ufw default deny incoming

```
sudo ufw default allow outgoing
sudo ufw allow ssh        # only if you need remote access
```

17. SSH Hardening

```
#If enabling SSH, limit SSH to reduce brute-force vulns. By dropping excessive login attempts automatically
    sudo ufw limit ssh
    sudo ufw enable
```

18. In Future, Allowing GUI Will Require Rules Update

```
#IMPORTANT: When deciding to run a web GUI which will call on the device for logs, you'll need to update the
UFW rules:
    sudo ufw allow 5636/tcp
#5636 is default EveBox port

#To check UFW status use:
    sudo ufw status verbose
```

#FUTURE security: consider using SSH keys instead of passwords

With my system being headless, I will enable SSH BUT if you do:

- Change the default password immediately.
- Use SSH key authentication instead of passwords.
- Implement Fail2Ban
- Consider changing the default port or restricting access by IP.
- Disable root login (PermitRootLogin no) in /etc/ssh/sshd_config.

19. Audit Running Services and Disable Unnecessary

```
#Check all services that are running, you only want suricata and essential system services:
    sudo systemctl list-unit-files | grep enabled

#To STOP and also disable a service
    sudo systemctl stop [service_name]      #Stops it now
    sudo systemctl disable [service_name]    #Prevents auto-start
            #extra step would be masking it, it cannot run until unmasked
                sudo systemctl mask [service_name]
            #unmask
                sudo systemctl unmask [service_name]
```

20. IMPORTANT: Implement Auto Security Updates

```
#Set updates to be automatic, download and set unattended-upgrades
    sudo apt update
    sudo apt install unattended-upgrades -y
    sudo dpkg-reconfigure --priority=low unattended-upgrades
#confirm it is running in the background successfully
    sudo systemctl status unattended-upgrades
#confirm it is set to run on boot, should return 'enabled'
    sudo systemctl is-enabled unattended-upgrades
```

21. Essential Services

#Essential services, do not disable:

Service	Purpose
ssh	Needed if you want remote management (optional if you never SSH)
systemd-journald	Core logging
systemd-logind	Manages user sessions
dbus	Core system communication bus
networking or dhcpcd	Handles network configuration, required for both static IP and DHCP
rsyslog	Logging (optional if using journald only, but recommended)
udev	Device management, mounts disks, USBs, etc.
cron	Scheduled tasks, useful for log rotation, updates, etc.
apt-daily.service & apt-daily-upgrade.service	Package updates if using unattended-upgrades
systemd-timesyncd	Time sync, important for logs and Suricata timestamps

22. Fail2Ban Notes (My XP)

#Tried to install fail2ban but device can't handle it so to remove half-config

```
sudo apt-get remove --purge fail2ban -y
sudo apt-get autoremove -y
sudo apt-get clean
sudo dpkg --configure -a
sudo apt --fix-broken install
```

Haven't done yet:

sudo ip link set eth0 promisc on

Ensure interface is on promiscuous mode, for ethernet

Limit log rule size with logrotate rules

```
sudo nano /etc/logrotate.d/suricata
```

#Update with

```
/var/log/suricata/*.log /var/log/suricata/eve.json {
   daily
   rotate 7
   compress
   missingok
   notifempty
   create 640 suricata suricata
   postrotate
      systemctl restart suricata > /dev/null 2>&1 || true
   endscript
}
```

Use Lite or ET Open rules instead of full rule set on old model like 2w

23. Setup External Drive for Logging; Plug in and Run to Identify Label:

lsblk

A. Optional Drive Format

#To format this drive, assuming name 'sda1', your label is found above with lsblk command

sudo mkfs.ext4 /dev/sda1 **(CAUTION! THIS WILL ERASE ALL EXISTING DATA)**

B. Mount & Auto-Mount

#Create mount point and mount drive, where 'sda1' is drive name found with lsblk

```
sudo mkdir -p /mnt/usb
sudo mount /dev/sda1 /mnt/usb
```

#mount automatically on boot by opening:

```
sudo nano /etc/fstab
```

#add this line:

```
/dev/sda1  /mnt/usb  ext4  defaults,noatime  0  2
```

#test without reboot

```
sudo umount /mnt/suricata-logs
sudo mount -a
```

C. Move Logs & Create SymLink

#Move existing logs and create symbolic link so writes to /var/log/suricata but data goes to USB

```
sudo systemctl stop suricata
sudo mv /var/log/suricata /mnt/usb/suricata
sudo ln -s /mnt/usb/suricata /var/log/suricata
```

#Ensure suricata can write to usb drive

```
sudo chown -R suricata:suricata /mnt/usb/suricata
```

#Restart suricata

```
sudo systemctl start suricata
sudo systemctl status suricata
```