SIGPwny

Purple Team FA2025 • 2025-10-02

Native Linux Forensics

Bryce Kurfman

Announcements

- CyberForce Teams should be set
  - 2 rosters, 6 per team, some alternates

- Post October 3rd, we will announce CyberForce training with ITI faculty involved
  - If you are on the roster or are an alternate, you will need to attend these meetings

- If you are attending CyberForce, check your emails immediately and submit by TONIGHT
  - individual registration
  - image release form

ctf.sigpwny.com

sigpwny{3v3ryth1ng_15_a_f1l3}

Overview

- Introduction to Endpoint Forensics
- Key Linux Forensic Artifacts
  - Terminal History
  - Important logs
  - Service Configurations
- Evidence Collection
  - Memory Acquisition w/ AVML & LiME
  - Triaging w/ UAC
  - Discussion about Disk Acquisition
- Live Response

Introduction to Endpoint Forensics

Endpoint Forensics

- Endpoint forensics is the process of collecting, preserving, and analyzing digital evidence from **individual hosts** (endpoints)
  - Focuses on
- The aim is to identify attacker actions on compromised machines while preserving evidence
  - This provides visibility into any malware on the system, persistence mechanisms, data exfiltration, and other critical information
- We can learn a lot from network forensics, but we attain ***much*** more information from the host-level

Larger Picture

- Remember that finding and killing a suspicious process is not the end of an engagement
  - It takes more than **pkill -9 susprocess** to fully respond to and recover from a cyber attack
- Once we know what an attacker did, it is **much easier** to remediate
  - For example, once we have located all persistence mechanisms, say a malicious systemd service and a cron job that runs a malicious script on every startup, we can take clear action to remove these
- The information we can gain from a given endpoint like malware signatures or common techniques an attacker has already used **also feeds back** into our detection techniques

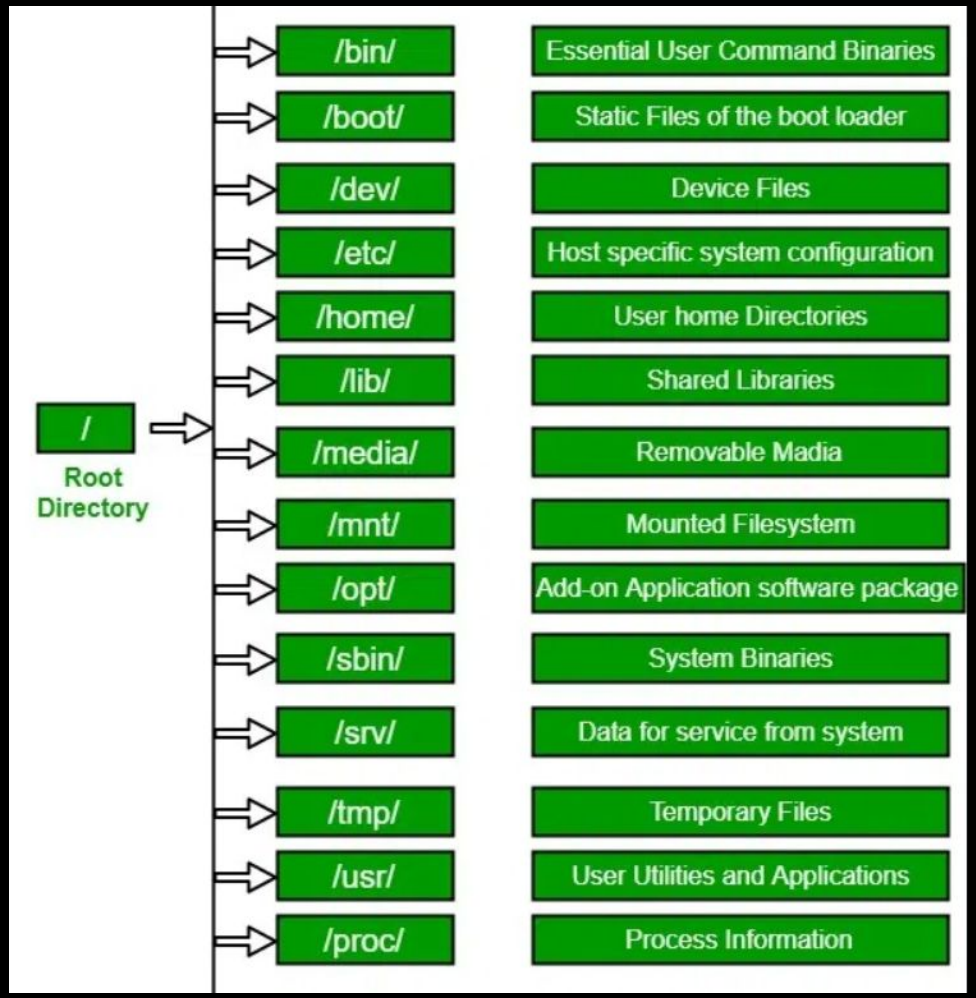
Chain of Custody

- Documenting every step of the acquisition process is mandatory for maintaining the chain of custody and ensuring evidence admissibility
  - This includes recording system details, timestamps, tool versions, hashes of acquired dumps, and any changes made during acquisition
- This is not as important to do for our purposes in competition or labs, but it’s always good to keep in mind to build good habits and awareness

Key Linux Forensic Artifacts

Linux Directory Layout

- Each directory has its own use case, so make sure you're familiar
- More on what to expect in each directory in a forensic image here



[Geeks for Geeks]

Artifacts

- Forensic artifacts are traces left behind on system that reveal user, process, and attack activity
- Artifacts provide evidence of compromise (logins, privilege escalation, lateral movement, etc.) and enable reconstruction of attacker actions and a **timeline** of when everything occurred
  - This all can be correlated with other sources like network traffic and evidence from other hosts to build a bigger picture of what went down

Command Line History

- ~/.bash_history or ~/.zsh_history
  - Logs the commands run by a user
  - Tracing user activity like what scripts they executed, privilege escalation attempts (running linpeas.sh is easily findable), or other malicious actions often starts here
  - You can sometimes find credentials entered in plaintext for different commands here

```
chmod +x output.py
output.py
./output.py
nano output.py
unzip x_2fuZUJ.txt
unzip x_2fuZUJ.zip
unzip archive.zip
cat x2fuZUJ.txt
cat X_2fuZUJ.txt
ftp --version
top | grep ftp
sudo apt install vsftpd
```

Linux Logs

- /var/log/ is the most important place where logs are stored, containing:
  - auth.log: logs related to login attempts, authentication failures, security related events, also every use of sudo
    - Useful for detecting brute force attacks, unauthorized logs, or privesc
  - syslog: general system events from system services, the kernel, and various applications
    - Records anomalies like unexpected service crashes, suspicious startup activities, etc.
  - cron.log: captures information about creation and execution of cron jobs
    - Can help create a timeline of suspect cron jobs
  - Also contains user.log, nginx/, boot.log, firewall and other service logs

SSH Configuration and Logs

- /etc/ssh
  - Contains all the SSH daemon settings that control:
    - How the server accepts connections (listening port, port forwarding, etc.)
    - What authentication methods are allowed (password vs. public/private key pair)
    - If root login in allowed, as well as if other users can login with SSH
  - Also contains the host's public (ssh_host_*_key.pub) and private (ssh_host_*_key) keys
    - Attackers can add and modify key as a persistence mechanism
    - It's also a privilege escalation path if an attacker can gain a private SSH key and login with a higher privileged user that way
- /var/log/auth.log
  - Documents remote access attempts (see slide 13 for more)

SSH Cont.

- In every user's home directory, there is a hidden .ssh folder which contains hashed names of all the known hosts that have connected in the past
  - Can't unhash them but if we know the names of what boxes should be connecting, we can hash those and compare to see if something is amiss
- Always make sure to check for **hidden** files and directories since those can often contain very useful information

Cron Jobs

- Cron jobs are scheduled tasks that run a command or script automatically at a specified time or on a recurring schedule
- /var/spool/cron/crontabs/ or /etc/crontab
  - Contains the cron tabs which hold all the cron jobs
  - Attackers often use cron jobs for persistence by running malware or opening backdoors at intervals

Package Management Logs

- /var/log/dpkg.log (Debian/Ubuntu)
  - Contains every package installation, upgrade, removal, and configuration action as well as timestamps
- /var/log/apt (Debian/Ubuntu)
  - apt acts as a front end to dpkg and simplifies package management by automatically handling dependencies
  - term.log contains the terminal output from apt operations, with package download URLs, errors, dependency resolution, etc.
- /var/log/dnf.log or /var/log/yum.log (RHEL/CentOS/Fedora)
  - Similar to the above but for different distros

Package Management Cont.

- When looking through package management logs, keep an eye out for:
  - Package with random or gibberish names
  - Packages installed from /tmp or unusual locations
  - Installations on hosts that shouldn’t have:
    - Development tools like gcc, make
    - Network tools like nmap
    - Cryptocurrency miners
    - Backdoor services like an OpenSSH or VNC server
  - Removals of security monitoring tools (auditd) , logging software (rsyslog), other system tools

Closing Note

- This is not by any means an exhaustive list of Linux artifacts, there are so many more out there that you can pull valuable data from
- Go and explore on the systems you have access to and find out where things are and what they do

Evidence Collection

Principles of Evidence Collection

- **Preserve**, don’t modify
  - Treat the live host as evidence, isolating it as much as possible and minimizing interactive changes
- Document Everything
  - Who ran what, when, and where
  - Record hashes
  - Keep chain of custody notes
- Verify and make sure actions are repeatable

Order of Operations

1. Capturing a full and unadulterated volatile memory dump of a suspected compromised machine **first** is very important
    a. Running processes, caches, active network connections, and decrypted data (encryption keys, passwords held by processes) exist only in RAM
    b. **Sophisticated** attackers can live entirely in memory and not drop any files to disk
    c. Try to acquire a memory dump ASAP so as little information is affected as possible
        i. All actions taken on a host *will* affect the memory in some way, so the less that you do before this, the truer the memory dump to its existing state
2. Take a triage image of the files and directories representing the most valuable forensic artifacts
    a. Making a full disk image is time consuming, and triaging allows us to get a quick start on the investigation

Memory Acquisition

- Acquire Volatile Memory for Linux (AVML)
  - Portable userland tool (static binary) that reads RAM via /proc/kcore and /dev/mem (More on AVML) for most common distros and kernels
  - Compile it once and you can port the binary around to different machine
    - Make sure to write the image to external storage or remote mount
  - sudo ./avml mem_output.lime
- Linux Memory Extractor (LiME)
  - LiME requires compiling a kernel module specific to each target machine's kernel which makes it quite annoying to use when doing analysis on many different machines
  - Try AVML first, but if that doesn't work, then move on to LiME or Linpmem.

Automating Triage

- Unix-like Artifacts Collector (UAC)
  - UAC automates the collection of artifacts from a wide range of Unix-like systems, ESXi, FreeBSD, Linux, macOS, OpenBSD, and others
  - Data collection is determined by easily customizable YAML profiles
  - Note that UAC comes with the AVML binary on it, so we can run **both** the memory acquisition and triage all in one command line!
    - ./uac -a ./artifacts/memory_dump/avml.yaml -p profiles/full.yaml /.../output
  - Always complement automated collection like this with manual inspection to see if anything extra funky is amiss that UAC might miss
  - Again, make sure to write the image to external storage or remote mount as to not affect the disk more than needed

Disk Imaging

- Once the memory is acquired and we have a triage image, then it is generally a good idea to take a full disk image
  - dd (larger block size for speed improvement)
    - dd if=/dev/sda of=/mnt/image_name bs=1M
  - Clonezilla (Instructions)
- We will not likely be doing this for any competitions or labs here given that it takes a lot of time and is hard to set up for multiple people

Live Response

During Competitions

- The competitions we are in simulate high tempo scenarios and make us do **immediate** threat hunting and containment, not full evidence collection and analysis
  - The goal is to rapidly identify suspicious hosts/processes, kill or isolate threats, remove the exploited vulnerability, and restore scored services.
- We’re currently planning out a large expansion of our capabilities for **CyberForce**, but more to come on that later

System Information

- date: Display current system date and time
- uptime: Show how long system has been running and load averages; detect unexpected reboots
- hostname: Display system hostname; confirm correct machine identity
- hostname -f: Display fully qualified domain name
- uname -a: Show all system information (kernel, architecture, OS version)
- cat /etc/os-release: Display detailed OS version and distribution information

User Activity and Information

- w: Show who is logged in and what they're doing; detect active attackers
- who: Display currently logged in users with login time and source
- cat /etc/passwd: List all user accounts with UID, home directory, and shell
- cat /etc/shadow: Display password hashes (requires root); check for unauthorized changes
- cat /etc/group: Show all groups and their members

Process Information

- ps aux: List all running processes with detailed information; find malicious processes
- ps auxf: Show process tree with parent-child relationships
- ps auxww: Show full command lines without truncation
- ps -u username: Show processes owned by specific user
- ps -p PID: Show information for specific process ID
- ps -C processname: Show processes with specific command name
- pstree: Display process tree structure
- pstree -p: Show process tree with PIDs
- pstree -a: Show process tree with command line arguments

Network Connections and Configurations

- ip route: Show routing table
- netstat -antup: Show all TCP/UDP connections with PIDs; identify backdoors and C2
- netstat -tulnp: Show listening TCP/UDP ports with PIDs
- netstat -i: Display network interface statistics
- netstat -s: Show network protocol statistics
- lsof +L1: Show deleted files still open

Firewalls

- iptables -L -n -v: List firewall rules with packet counts
- iptables -t nat -L -n -v: Show NAT table rules
- ip6tables -L -n -v: List IPv6 firewall rules
- iptables -L FORWARD -n -v: List forward chain rules
- iptables-save: Dump all iptables rules
- nft list ruleset: Show nftables configuration (modern systems)
- ufw status: Show UFW firewall status (Ubuntu)
- ufw status numbered: Show UFW rules with numbers

Open Files and Searching

- lsof: List all open files system-wide
- lsof /path/to/file: Show which processes have file open
- lsof -u username: Show files opened by specific user
- lsof -c processname: Show files opened by processes matching name
- lsof -p PID: Show files opened by specific process
- find / -type f -mmin -60: Find files modified in last 60 minutes
- find / -type f -perm -4000: Find SUID files (privilege escalation vectors)
- find / -type f -perm -2000: Find SGID files
- find / -type f -executable: Find executable files

Command History and Shell Information

- history: Show command history for current shell session
- cat ~/.bash_history: Display persistent bash history file
- cat ~/.zsh_history: Display zsh history file
- cat /root/.bash_history: Display root's command history (requires root)
- cat /etc/profile: Check system-wide shell configuration

Scheduled Tasks and Services

- crontab -l: List current user's cron jobs
- crontab -l -u username: List specific user's cron jobs
- cat /etc/crontab: View system-wide crontab
- ls -la /etc/cron.hourly/: List hourly cron scripts

- systemctl list-timers: List systemd timers (alternative to cron)
- systemctl show servicename: Show detailed service properties
- systemctl cat servicename: Display service unit file contents

SSH Configuration and Keys

- cat /etc/ssh/sshd_config: View SSH server configuration
- cat /etc/ssh/ssh_config: View SSH client configuration
- cat ~/.ssh/known_hosts: View known host keys
- ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub: Show SSH host key fingerprint

Resources

- The God Doc of Linux Forensics
  - Hal Pomeranz
- Practical Linux Forensics Appendix
  - Bruce Nikkel
- AzurePot Lab
  - CyberDefenders

Next Meetings

2025-10-07 • **This Tuesday**

- Windows & Windows Privilege Escalation
- Navigate Windows and how privileges escalate

2025-10-09 • **Next Thursday**

- Native Windows Forensics
- Checking out investigative aspects of Windows

2025-10-14 • **Next Tuesday**

- Active Directory I
- Fundamentals of Active Directory

ctf.sigpwny.com

`sigpwny{3v3ryth1ng_15_a_f113}`

Meeting content can be found at sigpwny.com/meetings.

SIGPwny