Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

# Forensic Artifacts in Modern Linux Systems

Prof. Dr. Bruce Nikkel

# Purpose and Scope of Workshop

Describe things of forensic interest, show how to find and extract data from:

- hacked/compromised Linux servers
- criminal operated Linux servers (Command and Control)
- abused/misused Linux desktop systems (suspect users, victim users)
- seized and imaged systems (dead disk forensics)
- focus on modern Linux system artifacts (systemd, etc.)
- focus on artifacts independent of Linux distribution

*Not* the focus of this workshop:

- using Linux as an analysis platform (most of this analysis can also be done with encase/ftk/xways)
- how to use Linux based forensic tools
- live Linux system analysis and memory forensics
- Linux based mobile devices (Android)
- Application artifacts (browser, email, office, etc)

This is not exhaustive, there are many OS artifacts not covered here (obscure/rare artifacts, distro specific artifacts, etc.)

# Overview of Workshop

High level overview of workshop topics (from a forensic/investigative perspective):

- partitions and filesystems
- mbr/uefi, grub, initrd/initramfs
- linux file/directory layout
- systemd: boot/shutdown, services, scheduled tasks
- installed software and packages
- log files and systemd journal
- swap, cache, and persistent data
- system and user configuration
- desktop artifacts
- encryption and steganography
- conclusion

Workshop format: mix of theory/slides and demonstrations
Example disk images are shown as either "image.dd" or "/dev/sda"

# Partitions and Filesystems

Examples of typical storage devices:

- SATA drives: /dev/sda
- NVME drives: /dev/nvme0n1
- MMC/SD cards: /dev/mmcblk0
- (Virtual Machine: /dev/vda)

Examples of typical partition devices:

- /dev/sda1
- /dev/nvme0n1p1
- /dev/mmcblk0p1
- (/dev/vda1)

Most common partition schemes are DOS and GPT

- disktype /dev/sda
- mmls image.dd
- UEFI systems have GPT layout and use a system partition with a FAT filesystem for EFI boot files

# Partitions and Filesystems

Some examples of filesystems used by modern Linux:

- typical for installation: ext4, btrfs, xfs
- many others supported: fat, ntfs, ext2, ext3
- network filesystems: nfs, samba/cifs, sshfs
- pseudo filesystems: proc, sys, dev

Interesting artifacts about an EXT4 filesystem:

- when the filesystem was created
- last mounted, last written, last checked
- number of times mounted
- last repaired
- tune2fs -l /dev/sda1
- fsstat /dev/sda1

Network and Virtual filesystems are interesting in live system analysis, less for dead disk analysis (but we can try to find out some things, like when/where they were mounted)

# Partitions and Filesystems

The SleuthKit ("TSK") has filesystem analysis tools for:

- listing and extracting files, inodes, blocks
- identifying and extracting deleted files
- building timelines (MACB timestamps)
- extracting slack and unallocated areas for analysis
- other filesystem artifacts (journaling filesystems,etc.)

All the TSK commands grouped by function:

- Partition/volume analysis: mmcat, mmls, mmstat, fsstat, img_cat, img_stat
- Analyzing by blocks/sectors: blkcalc, blkcat, blkls, blkstat
- Analyzing by inodes: icat, ifind, ils, istat, tsk_recover
- Analyzing by filename: fcat, ffind, fls, fiwalk
- Journaling filesystems: jcat, jls, usnjls
- Timelines: mactime, tsk_gettimes
- Search and sort: jpeg_extract, sigfind, sorter, srch strings, tsk_comparedir, hfind, tsk_loaddb

These commands work on: attached devices, raw images (dd), and forensic images (EnCase/FTK).

# MBR/UEFI, Grub, initrd/initramfs

MBR - 512 byte boot sector, jumps to next stage loader

- can analyze boot sector for possible malware (boot sector viruses are rare today)
- dd if=image.dd of=bootsector.dd bs=512 count=1

UEFI - FAT system partition with files, more intelligent boot loading

- look for unusual efi binaries
- if you have access to mainboard, get UEFI variables stored in NVRAM

Grub artifacts (GRand Unified Bootloader)

- /boot/grub/grub.cfg or /boot/grub2/grub.cfg
- /etc/grub.d/* or /etc/default/grub
- can show list of previous OS installations, kernel parameters used, etc.

Kernel ramdisk (initrd or initramfs)

- debian: lsinitramfs -l /initrd.img
- fedora: lsinitrd -v /boot/initramfs-4.16.11-100.fc26.x86_64.img
- arch: lsinitcpio -v /boot/initramfs-linux.img
- suse: lsinitrd /boot/initrd
- if root filesystem is encrypted, may have interesting cleartext info

# Linux File System Layout

Directories of interest to forensic investigators:

- bootstrap configuration /boot (efi partition mounted on /boot/efi)
- system configs: /etc
- logs, cache, state: /var (especially /var/lib and /var/log)
- user data: /home and /root

Some directories are mountpoints for pseudo filesystems:

- /proc, /sys, /dev, /run
- not very useful for dead disk forensics

Other tips:

- be aware of hidden files/dirs (filenames starting with ".")
- the "FILES" section of manpages can indicate items of potential interest
- use forensic timelines to identify 'busy' directories

# Systemd Boot/Shutdown, Services, Scheduled Tasks

Systemd

- modern Linux system and service manager
- very consistent across distributions
- manages starting, stopping, restarting of daemons

Systemd configuration common locations:

- defaults: /usr/lib/systemd/ or /lib/systemd/
- custom: /etc/systemd/
- user: ~/.config/systemd

These directories contain systemd unit files, that configure or control:

- services and daemons
- sockets and devices
- mount points, automount points
- swap files and swap partitions
- start-up targets
- timers (scheduled jobs), watched file system paths

Provides forensic trace information about the system and user configuration

# Systemd Boot/Shutdown, Services, Scheduled tasks

Examples of things to look for as a forensic investigator:

- overview of services started on boot
- proxy and relay daemons
- strange services that could be backdoors or malcious code
- vpn tunnels (new: wireguard vpn, this is growing in popularity, look for /etc/wireguard/, the wg0 interface, or systemd wireguard files)
- service units for: bitcoin, torrent, tor, tunneled protocols, etc.

Scheduled jobs:

- traditional cron jobs: /var/spool/cron, /var/spool/anacron, /etc/cron.*/*, and /etc/crontab
- traditional at jobs (one time execution): /var/spool/at
- systemd timers (*.timer files)
- user and system jobs are separate (for cron and systemd)

Note: there are over 150 manpages describing systemd and various relevant files

# Installed Software and Packaging

Most popular packaging formats (not consistent across distributions):

- rpm (redhat and suse)
- apt/deb (debian/ubuntu, etc.)
- pacman/tar (arch, manjaro)

The interesting forensic artifacts in packaging systems are:

- list of installed software packages (package databases)
- removed software packages (install logs, previously downloaded packages)
- install and removal timestamps

Other packaging systems growing in popularity:

- snap/snappy
- flatpacks

Backups and archive files (ok, not packages, but...):

- tar snar files have a list of deleted, changed, created files from backups
- tar --show-snapshot-field-ranges

# Installed Software and Packaging

Debian based systems

- logs: /var/log/apt/*
- database: /var/lib/dpkg/* (especially the 'status' file)

Redhat and SuSe based systems

- logs: /var/log/dnf.rpm.log*
- database: /var/lib/rpm/*

Arch pacman based systems

- arch also has "AUR" or Arch User Repository user
- database: /var/lib/pacman/local/*/*
- logs: paclog command, /var/log/pacman.log

Note: users can bypass the packaging system and copies any files anywhere ('make install' for example).

# Log Files and Systemd Journal

Programs and daemons typically log to one of three places:

- traditional syslog (/var/log/messages or /var/log/syslog)
- systemd journal
- self written log files (usually in /var/log/*)

Traditional Linux logging:

- logs can be different levels of verbosity (debug, informational, etc.)
- a running linux kernel has a ring buffer log (dmesg)
- applications may separate error logs from transaction logs
- syslog messages are sent to a syslog daemon and saved to files

Systemd journal has features that are interesting for investigators:

- better recording of logs during early system initialization
- stderr output of a daemon is captured
- logs are stored in a structured binary format that can be filtered, searched, or exported
- Forward Secure Sealing (FSS) preserves integrity of the logs (like a forensic chain of custody)

# Log Files and Systemd Journal

Journalctl data and commands:

- location: /var/log/journal/$RANDOMSTRING/*
- system logs: system@*
- user logs (with UID): user-1000@*
- journalctl --root=/location/of/forensic/image/mount/
- journalctl --file=user-1000@
- journalctl --directory=/some/directory/with/journal/

Journalctl tips:

- logged boots: journalctl --list-boots
- kernel messages: journalctl --dmesg
- time periods: journalctl --since=2018-09-05 --until=2018-09-06
- more verbose: journalctl -ax
- search with "/", n-next, N-previous

# Log Files and Systemd Journal

What you might find in the logs and systemd journal:

- attached and mounted USB drives
- network interfaces and MAC addresses (NetworkManager)
- dhcp results with IPs addresses (NetworkManager)
- evidence of malicious activity and attacks (failed logins)
- successful logins (local and remote) and user sessions
- reboots, boots, daemon start/stop/restart
- virtual network interface creation (vpns/tunnels)
- application/daemon errors and messages
- user activity (pgp/gpg agent activity)
- notebook Lid close/open, power cable plugin

Files in /var/log/* are disappearing from use, so learn journalctl
Some systems may not keep a persistent copy of the journal across boots
Most systems still have utmp/wtmp files: last -f /var/log/wtmp

# Cache, Swap and Persistent data

Desktop systems using NetworkManager cache interesting things:

- /var/lib/NetworkManager/*
- dhcp leases and timestamps
- observed wifi bss ids

Desktop systems with Bluetooth cache interesting things:

- /var/lib/bluetooth/*
- paired bluetooth devices
- file timestamps reveal previous pairing activty

Lots of really great info in /var/lib, often with timestamps:

- depending on the software installed, all kinds of interesting system persistence and cached data
- example: switching from charging to discharging (/var/lib/upower/*)
- (hint, convert epoch timestamps to human time: date -d @1535347485)

# Cache, Swap and Persistent data

Temporary files and directories

- /tmp and /var/tmp may contain files
- (but may be deleted after boot or logout)
- swapfile or swap partition (see /etc/fstab)

If swap is the size of ram or larger, it can be used for hibernation:

- a hibernating system has a complete memory dump saved to disk
- check the end of journal to see if the system went into hibernation
- can be extracted with forensic tools (icat, dd, etc.)
- memory analysis can be done to find many artifacts:
  running processes, established network connections, possibly keys and passwords

Printers and printed pages

- attached and configured printers: /etc/cups/*
- print jobs from cupsd: /var/cache/cups/*
- /var/spool/cups/* and /var/log/cups/*

Large amounts of cached user data in /home/user/.cache, this contains application data (photo thumbnails for example)

# System and User Configuration

System and kernel:

- LSB (Linux Standards Base): /etc/lsb-release or o/etc/os-release
- kernel version: file vmlinuz
- kernel config/parameters grub.cfg and /etc/sysctl.*
- kernel modules: /etc/modprobe*, /etc/modules, /etc/modules-load*
- startup services/daemons (systemd units)

Systemd network config:

- default: /usr/lib/systemd/network/ or /lib/systemd/network
- custom: /etc/systemd/network/
- also distro specific (debian /etc/network/interfaces)

Crashed programs

- system may need to be configured to save core files
- /var/lib/systemd/coredump
- this is a memory dump of a process: established network connections, possibly keys and passwords
- manpage core(5)

# System and User Configuration

Users and groups

- traditionally in /etc/passwd and /etc/group
- some systems may use ldap
- UID and GID analyzed as filesystem meta data
- (Sleuthkit: mactime -p /etc/passwd -g /etc/group)
- users and groups may refer to people or processes
- difference between system and application activity is not always clear
- difference between system and user activity is not always clear

OS and user configuration files

- traditional Unix/Linux files in /etc
- gconf/dconf, systemd units
- dot files ~/.*
- dot files ~/.config/*
- user customized shells (.bashrc) and shell history
- each distro may have additional configuration artifacts that are interesting

# Desktop Artifacts

Freedesktop.org (formerly known as: X Desktop Group or XDG)

- XDG documentation and specifications at freedesktop.org
- Provides compatibility across distros and desktop environments
- KDE and Gnome most popular DEs

Key directories interesting to forensic investigators:

- systemd unit files for XDG: /etc/xdg/systemd
- $XDG_DATA_HOME, default ~/.local/share
- $XDG_CONFIG_HOME, default ~/.config
- $XDG_CACHE_HOME, default ~/.cache
- contains user's GUI data and configuration
- (there are also system-wide equivalent defaults)

# Desktop Artifacts

Interesting things we find here:

- autostarting GUI apps ~/.config/autostart/*
- contents of user's desktop: ~/Desktop (contains Desktop entry files)
- recently opened docs: ~/.recently-used or ~/local/share/recently-used.xbel
- thumbnails ~/.cache/thumbnails
- "Trash" ~/.local/share/Trash or ~/.Trash
- User override default apps: ~/.config/mimeapps.list
- application downloads: ~/Downloads

Other notes

- often 2 sets of timestamps: filesystem (MACB) and timestamps inside the files
- These directories and filenames may vary depending on the desktop and XDG variables
- X11 vs Wayland? These both operate below the XDG/freedesktop.org Environment, so it should (mostly) not matter

# Encryption and Steganography

Forensic examiners will find different types of encryption:

- application file encryption - protected PDF, office docs, etc
- individual file containers - GPG, Encrypted Zip
- directories - ecryptfs, ext4 encrypted sub-directories
- volumes - TrueCrypt/Veracrypt
- block devices - Linux LUKS
- drive hardware - OPAL/SED

Decrypting requires:

- password or passphrase
- cryptographic key string or key file
- smartcard or hard token

The forensic challenge is to find the decryption key
(some tools: John the Ripper, HashCat, bulk_extractor, $5 wrench)

# Encryption and Steganography

Steganography is considered a part of anti-forensics

- It hides data in non-obvious places
- least significant bits of color, sound, etc.
- tries to hide data in different slack areas
- Veracrypt allows hiding volumes inside volumes

Some tools:

- stegdetect
- stegsnow
- openstego
- busysteg
- gsteg
- photocrypt

# Conclusion

- Thanks for listening!
- If you know some additional Linux forensic artifacts not mentioned, please send them to me, I'll add them to my slides.
- Law Enforcement: You are welcome to contact me at BFH for Linux analysis support.
- Contact details: bruce.nikkel@bfh.ch
- These slides are available at: digitalforensics.ch