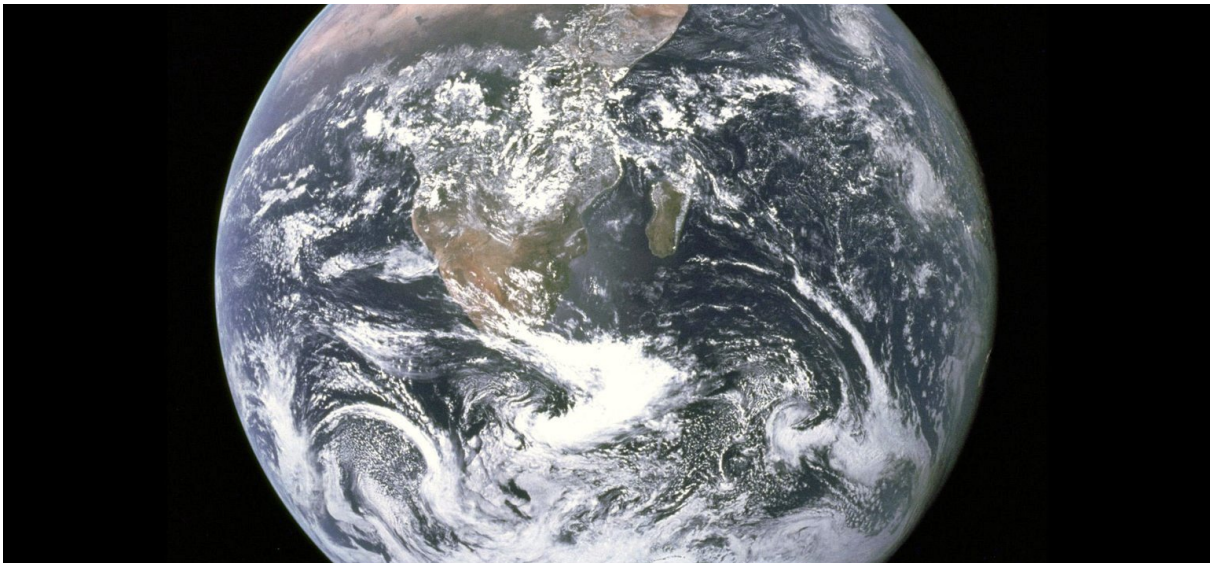


Unit - 3

Linux Forensic

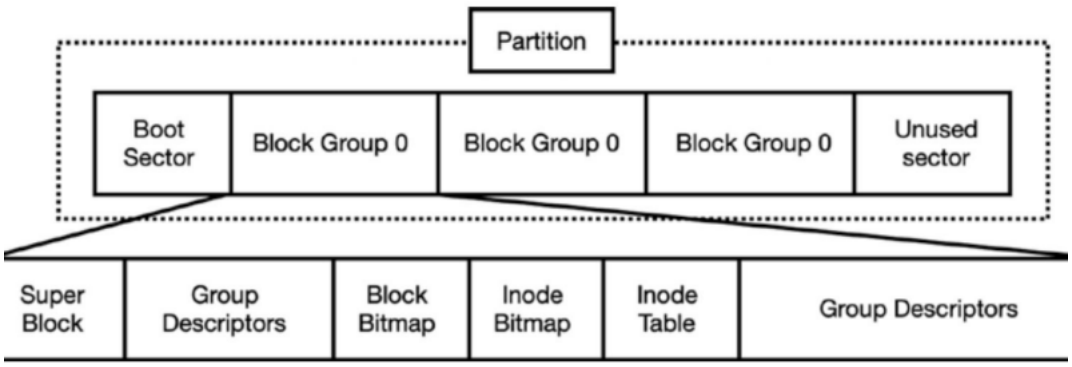
In Linux System we use Ubuntu for servers and cloud computing, Debian for stability, and Kali Linux for penetration testing and digital forensic analysis.

Here is a list of a few popular Linux Distributions that are commonly used.

- Red Hat Linux
- Ubuntu
- Fedora
- Debian
- SUSE
- Mint
- Arch Linux
- Linux Lite

Default file system for modern Linux system is EXT4

Whenever a file is created or saved, it gets indexed by a number or inode. These inodes have multiple attributes attached to it, which is the metadata



EXT4 File System Components

Boot Block

- Contains the bootstrap code
- Located on the main hard disk
- UNIX/Linux computer has only one boot block

Superblock

- Specifies disk geometry and available space
- Keeps track of all inodes
- Part of metadata
- Manages the file system

Inode Blocks

- First data after the superblock
- Assigned to every file allocation unit
- Created or deleted as files or directories are created or deleted

Data Blocks

- Store directories and files on a disk drive
- Linked directly to inodes

Inodes

Contents

- File and directory metadata
- Links data stored in data blocks

Assigned Inode Contains:

- Mode and type of file or directory
- Number of links to a file or directory
- UID and GID of the file's or directory's owner
- Number of bytes in the file or directory
- File's or directory's last access time and last modified time
- Inode's last file status change time
- Block address for the file data
- Indirect, double-indirect, and triple-indirect block addresses for the file data
- Current usage status of the inode
- Number of actual blocks assigned to a file
- File generation number or version number
- Continuation inode's link

Inode Pointers

- First inode has 13 pointers
  - Pointers 1 to 10 are direct pointers to data storage blocks
  - Pointer 11 is an indirect pointer, links to 128 pointer inodes, each pointing to 128 blocks
  - Pointer 12 is a double-indirect pointer
  - Pointer 13 is a triple-indirect pointer

Inode pointers in Linux file system

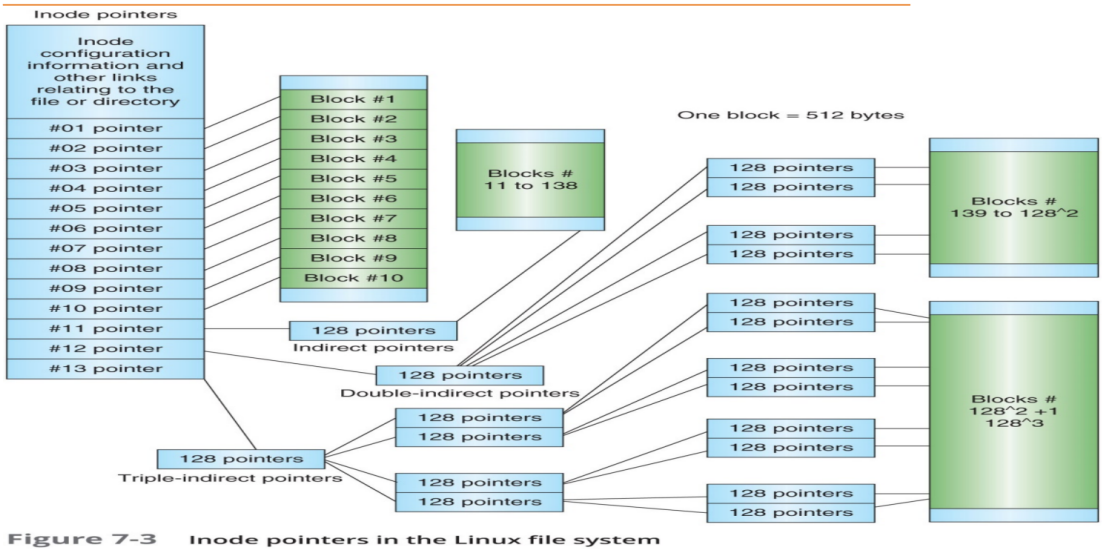


Figure 7-3 Inode pointers in the Linux file system

File System Layer

Components

- **Superblock:**
  - Located 1024 bytes from the start of an Ext file system
  - Contains:
    - Layout information of the file system
    - Block and inode allocation information
    - Metadata indicating the last mount or read time
- **Group Descriptor Table:**
  - Found in the block immediately following the superblock
  - Contains allocation status information for each block group on the file system

File Name Layer

- File names stored as directory entries
- Directory entries stored in blocks filled with directory entries
- Each directory entry contains:
  - File name
  - Address of the inode associated with the file
  - Flag indicating whether the name refers to a directory or a normal file

Note:

- **Device Files:**
  - Linux has special device files
  - Encountered when examining a Linux system

Metadata Layer

- Metadata (MAC TIMES) for files stored in inodes
- Forensically interesting items in Ext inodes:

- File's size and allocated blocks
- Ownership and permissions information
- Time stamps associated with the file

- Each inode has a link count, indicating the number of file names referring to it
- Ownership information includes:
  - User Identifier (UID)
  - Group Identifier (GID)

Dataunit Layer

- Data units in Ext file systems called blocks
- Blocks are 1, 2, or 4K in size as denoted in the superblock
- Each block has an address and is part of a block allocation group
- Block addresses and groups start from 0 at the beginning of the file system and increment
- Pointers to allocated blocks for a file are stored in the inode
- Current Linux kernels fill block slack space with zeroes when writing data, leaving no "file slack"
- Allocation strategy places blocks in the same group as the inode to which they are allocated

Deleted Data

- For each directory entry pointing to an inode, the inode's link count is incremented
- When directory entries pointing to an inode are removed, the inode's link count is decremented
- When all directory entries pointing to an inode are removed, the inode's link count is zero, and it is considered "deleted"
- On Ext2 systems, the process stops here, making recovery easy
- On Ext3 systems, when the link count of an inode is zero, the block pointers are zeroed out, severing the link between metadata and data
- Deleted blocks and inodes remain preserved until reallocated and overwritten, effectively "freezing" the data
- Attackers may exploit this by placing malware in low-use areas, allowing deleted blocks and inodes to remain preserved

```
user@ubuntu:~$ stat file1
  File: 'file1'
  Size: 11          Blocks: 8      IO Block: 4096 regular file
Device: 801h/2049d  Inode: 452126    Links: 1
Access: (0644/-rw-r--r--) Uid: ( 1000/ user) Gid: ( 1000/ user)
Access: 2010-10-19 21:06:36.534649312 -0700
Modify: 2010-10-19 21:06:34.798639051 -0700
Change: 2010-10-19 21:06:34.798639051 -0700
```

The fifth line contains the information of interest—
The "Access: (0644/-rw-r—r--)" item are the permissions, and the rest of the line is the ownership information. This file is owned by User ID 1000 as well as Group ID 1000. W

- Linux permissions are divided among three groups, and three tasks.
- Files and directories can be read, written, and executed.
- Permissions to perform these tasks can be assigned based to the owner, the group, or the world (aka anyone with access to the system).
- This file has the default permissions a file is assigned upon creation.
- Reading from left to right, the owner (UID 1000) can read and write to the file, anyone with a GID of 1000 can read it, and anyone with an account on the system can also read the file.

Hidden Files On Linux systems

Hidden Files On Linux systems, files are "hidden" from normal view by beginning the file name with a dot (.)

These files are known as dotfiles and will not be displayed by default

/tmp

—it is a shared scratch space, and as such all users have write permissions to this directory.

It is typically used for system-wide lock files and nonuser-specific temporary files

```
drwxrwxrwt 13 root root 4.0K 2010-10-15 13:38 tmp
```

Explanation:

- **Type**: `d` (indicates it's a directory)
- **Permissions**:
  - Owner: `rwx`
  - Group: `rwx`
  - Others: `rwt`
- **Number of links**: `13`
- **Owner**: `root`
- **Group**: `root`
- **Size**: `4.0K`
- **Last modified date**: `2010-10-15 13:38`
- **Name**: `tmp`

The sticky bit is set because there's a `t` in place of `x` at the end of the permission string.

Files under a directory with the sticky bit set can only be deleted by the user that owns them (or the root user)

Stickiness overrules other permissions

User Accounts

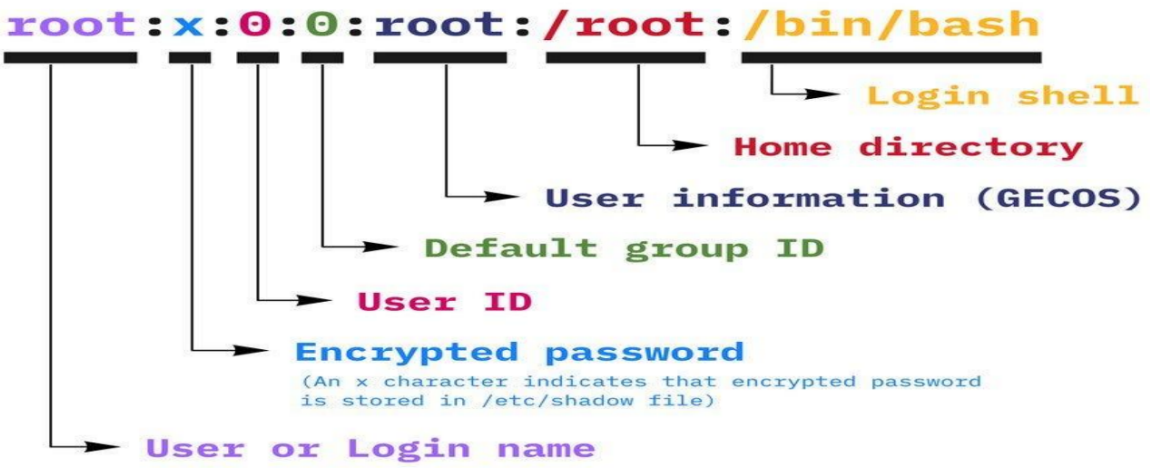
The first place to begin looking for information related to user accounts is the “/etc/ passwd” file.

The passwords for user accounts are generally stored in the “/etc/shadow” file.

Here is a typical entry in the `/etc/passwd` file with a description of each field:

```
forensics:x:500:500::/home/forensics:/bin/bash
```

1. **Username**: `forensics`
2. **Hashed Password Field** (Deprecated): `x` (Previously, the hashed password would be stored here, but it's deprecated now and usually set to `x` to indicate that the password is stored in the `/etc/shadow` file.)
3. **User ID**: `500`
4. **Primary Group ID**: `500`
5. **GECOS Comment Field**: (Generally used for the user’s full name or a more descriptive name for a service account, but it's empty here.)
6. **User's Home Directory Path**: `/home/forensics`
7. **Initial Login Program (Default Shell)**: `/bin/bash`



The `/etc/passwd` file used to hold the password hashes, but due to security concerns, the password hashes were moved to the `/etc/shadow` file. The `/etc/passwd` file now contains placeholders (often 'x') in the password field, indicating that the actual password hash is stored elsewhere. This improves system security by limiting the exposure of password hashes to only those who need them.

The “/etc/group” file

Here's an explanation of the fields in a typical entry in the `/etc/group` file:

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
wheel:x:10:root
```

1. **Group Name**:
   - `root`
   - `bin`
   - `daemon`
   - `wheel`
2. **Group Password Hash**:
   - `x` (Password-protected groups are not typically used)
3. **Group ID**:
   - `0` for `root`
   - `1` for `bin`
   - `2` for `daemon`
   - `10` for `wheel`
4. **Group Members** (comma-separated list):
   - `root` for `root` group
   - `root,bin,daemon` for `bin` group
   - `root,bin,daemon` for `daemon` group
   - `root` for `wheel` group

The “/etc/shadow” file

It contains hashed user passwords and password-related information

here's the breakdown of the entries in the `/etc/shadow` file:

```
root:$1$gsGAI2/j$jWMnLc0zHFtlBDveRqw3i/:13977:0:99999:7:::
bin:*:13826:0:99999:7:::
gdm:!!:13826:0:99999:7:::
user:$1$xSS1eCUL$jrGLlZPGmD7ia61kIdrTV.:13978:0:99999:7:::
```

1. **Username**:
   - `root`
   - `bin`
   - `gdm`
   - `user`
2. **Password Field**:
   - `$1$gsGAI2/j$jWMnLc0zHFtlBDveRqw3i/`: Encrypted password for `root`
   - `: No password for `bin`
   - `!!`: No password for `gdm`
   - `$1$xSS1eCUL$jrGLlZPGmD7ia61kIdrTV.`: Encrypted password for `user`
3. **Last Password Change**:
   - `13977` for `root`
   - `13826` for `bin` and `gdm`
   - `13978` for `user`
4. **Minimum Password Age**:
   - `0` for all users
5. **Maximum Password Age**:
   - `99999` for all users
6. **Password Warning Period**:
   - `7` for all users
7. **Inactive**:
   - Blank for all users
8. **Expiration Date**:
   - Blank for all users
9. **Reserved**:
   - Blank for all users

The `*` and `!!` in the password fields for daemon accounts (`bin` and `gdm`) indicate that these accounts do not have encrypted passwords. These accounts are not meant for interactive logins, so they have a null or invalid password field.

Scheduling Tasks

On Linux systems there are two main mechanisms for scheduling a job to be run in the future: at and cron.

The at command is used to run a task once, at a specific point in the future.

at Command:

- Used to run a task once, at a specific point in the future.
- at jobs can be found under `/var/spool/cron`.

cron Process:

- Used to schedule repeating tasks.
- System cron jobs are stored in:
  - `/etc/crontab`
  - `/etc/cron.hourly`
  - `/etc/cron.daily`
  - `/etc/cron.weekly`
  - `/etc/cron.monthly`
- User-added scheduled tasks are found in `/var/spool/cron`, including jobs added by the `at` command.

Cron jobs are an effective way for an attacker to maintain persistence on a compromised system, so verifying these jobs is critical in an intrusion investigation.

Bad block inode

Keeps track of disk's bad sectors

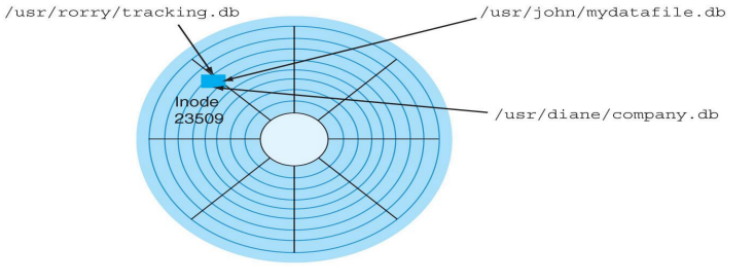
To find bad blocks on a Linux computer, use the following commands:

- `badblocks` (must log in as root to use)
- `mke2fs` and `e2fsck`

Hard Links and Symbolic Links

A hard link is a pointer that allows accessing the same file by different filenames

The filenames refer to the same inode and physical location on a drive.



Link count

A field inside each inode that specifies the number of hard links

Command used: ls –ia

If two files have the same inode number, the link count is two. If one file is deleted, the link count drops by one.

Symbolic links

- Symbolic links (also known as "soft links" or "symlinks") are simply pointers to other files and aren't included in the link count.
- Symbolic links have an inode of their own, which isn't the same as the inode of the item they're pointing to.
- Unlike hard links, symbolic links depend on the continued existence of the destination they're pointing to.
- Symbolic links can point to items on other drives or other parts of the network; they simply need an absolute path.
- Unlike hard links, which point to their destination with an inode number, symbolic links identify their destination by name and path.
- If a name and path no longer exist, the symbolic link stops working.

Forensic Process for Linux Systems

A forensic investigator will follow the same protocol for the forensic examination of a Linux system as for Windows.

Forensic Artifacts

Directory	Description
/bin	The essential command binaries
/boot	Files required for the system bootloader
/dev	Device files
/etc	System configuration files
/home	Home directories
/lib	Shared libraries and kernel modules
/media	Mount points for removable media
/opt	Add-on application packages
/root	Root user home directory
/sbin	System binaries
/tmp	Temporary files
/var/logs	Centralized repository of log files

Artifacts	Location
User profile	/home/$USER
System and Application logs	/etc
Operating system information	/etc/os-release
Operating system install	/root/install.log
Host/ Computer name	/etc/hostname
IP address, DNS	/var/log
Time Zone Information	/etc/timezone
User login History	/var/log/auth.log
Recently Accessed files	/home/username/ local/share/recently-used.xbel
Command History	$HOME/.bash_history

lists of several Linux system files containing information about users and their activities

System file	Contents
/etc/exports	File systems exported to remote hosts; might include remote drive mappings
/etc/fstab	File system table of devices and mount points
/var/log/lastlog	User's last logon
/var/log/wtmp	Logon and logoff history information
/var/run/utmp	Current user's logon information
/var/log/dmesg	System messages log
/var/log/syslog	System log, occasionally called system.log or kernel.log
/etc/shadow	Master password file, containing hashed passwords for the local system
/etc/group	Group memberships for the local system
/etc/passwd	Account information for the local system

Core top-level directories of a Linux system

Directory	Contents
/usr	Most applications and commands are in this directory or its subdirectories bin (stands for "binary" and contains binary files required at boot time) and sbin (which requires superuser permission to run the binaries in it).
/etc	Most system configuration files are stored in this directory.
/home	The home directories for all users, usually named after their usernames.
/root	The home directory for the root user (superuser), which is kept separate from other user home directories.
/dev	Device files that act as stand-ins for the devices they represent, as described in Chapter 3; for example, /dev/sda is the first non-IDE disk drive on the system, usually the main hard drive.
/var	Subdirectories such as log (often useful for investigations), mail (storing e-mail accounts), and spool (where print jobs are spooled).

Directory	Description
/bin	The essential command binaries
/boot	Files required for the system bootloader
/dev	Device files
/etc	System configuration files
/home	Home directories
/lib	Shared libraries and kernel modules
/media	Mount points for removable media
/opt	Add-on application packages
/root	Root user home directory
/sbin	System binaries
/tmp	Temporary files
/var/logs	Centralized repository of log files

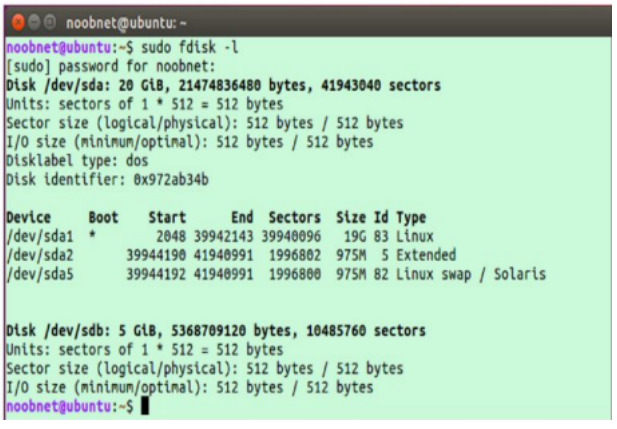
Challenges in Linux Digital Forensics:

1. **Data Distribution:**
   - Linux lacks a central Registry like Windows, scattering data across the OS and requiring collection from multiple sources.
2. **Metadata Zeroing:**
   - Metadata for files is zeroed when deleted, complicating data recovery.
3. **Linux System Usage:**
   - Linux is less commonly used in home systems compared to Windows, resulting in fewer specialized forensic tools.

4. **Limited User Base:**
   - Linux's primarily advanced computing usage poses challenges for cyber forensic examiners.
5. **Diverse Distributions:**
   - Various Linux distributions with unique features complicate forensic analysis due to OS configuration variations.
6. **Tool and Technique Adaptation:**
   - Cyber forensic experts must study Linux systems to identify important artifacts and use compatible tools.
7. **File System Compatibility:**
   - Compatibility issues with some forensic tools arise due to the adoption of the EXT4 file system in modern Linux systems.
8. **Command-Line Tools:**
   - Most Linux forensic tools are command-line-based, potentially posing usability challenges for some users.
9. **Future Tool Development:**
   - With evolving trends, more user-friendly Linux forensic tools are expected to emerge, addressing usability challenges and catering to a broader user base.

Windows	Linux
Windows has a central Registry that is used for collecting and storing the configuration settings of Windows components, installed hardware & software applications, etc.	Linux does not have a central Registry like Windows. The data is scattered across the OS, which has to be collected from multiple sources.
Windows supports FAT (with its variations) or NTFS file systems.	Linux supports EXT (with its variations) file system.
Most of the tools are GUI based and easy to understand or use.	Most of the Linux tools are command line and not GUI based, and hence they are not the easiest ones to use.
In Windows, you can have many user accounts with administrative privileges.	Linux has only one administrative account called root. Root account has complete control of the system.
In Windows, you can find file permissions in the Security tab of Properties section of My Computer, and they are kept in Registry.	In Linux, by running the **ls l** command on a directory or on a particular file, you can view these file permissions.
Windows has a Recycle Bin folder to store deleted files, and these deleted files can be recovered from it.	Linux distributions have Trash functions that contain deleted files of the particular user.

Listing Partitions



Here's the information presented in a clearer format:

- Use `fdisk -l` to list all available drives and partition information.
- Disk names in Linux are stored in alphabetical order.

- Example:
  - Two hard drives:
    - `/dev/sda` (20GB) with partitions `/dev/sda1`, `/dev/sda2`, and `/dev/sda5`.
    - `/dev/sdb` (5GB).
- After obtaining the list of hard disks, create a dd image of the disk or a hard drive or flash drive.
- `dd` is a command-line utility for Unix operating systems used for copying and converting files.
- To create an image of a partition on a Linux system, use the following command:

```
dd if=/dev/sdb of=image.001 bs=2M status=progress
```

  - `if=/dev/sdb`: Read from partition `/dev/sdb`.
  - `of=image.001`: Write the contents of partition `/dev/sdb` to `image.001` file.
  - `bs=2M`: Read and write 2048 (2MB) of file at a time.
  - `status=progress`: Show the status of the number of bytes copied to the file.

```
sudo insmod ./lime-4.15.0-29-generic.ko "path=../Linux_ Memory.mem format=raw“
```

  To capture RAM contents

Mac Forensics

Mac systems use an SSD in place of an HDD

System	Description
MacOS X	- Introduced as the 10th major version of Apple's operating system, denoted by the letter "X" representing the number 10.
	- Had a distinct code base compared to its predecessor, Macintosh, being based on the NeXTSTEP operating system code base.
	- Core of Mac OS X is Darwin, an open-source software.
	- Implemented preemptive multitasking and memory protection to improve the system's ability to run multiple applications simultaneously without interference or corruption.
OS X	- In 2012, the name of the operating system was shortened from Mac OS X to OS X.
	- OS X featured a new user interface design, including deep color saturation, text-only buttons, and a minimal, ‘flat’ interface.
macOS	- In 2016, the name of the operating system was changed from OS X to macOS to maintain the branding of Apple’s other primary operating systems like iOS, watchOS, and tvOS.
	- Introduced features like Siri on macOS and optimized storage.
	- Provided greater integration with Apple’s iPhone and Apple Watch.
	- Introduced the Apple File System (APFS) as a replacement for the HFS+ file system.

An Overview of Mac File Structures

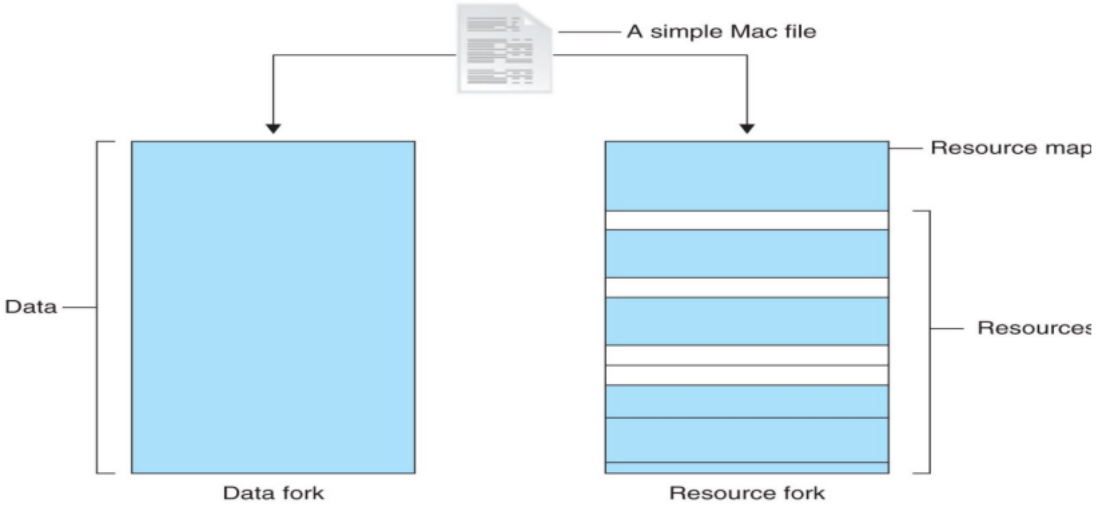
In Mac, a file consists of two parts:

1. **Data Fork:**
   - Contains user-created data such as text or spreadsheets.
   - Applications read and write to the data fork.
2. **Resource Fork:**
   - Contains additional information such as menus and dialog boxes.

Additionally:

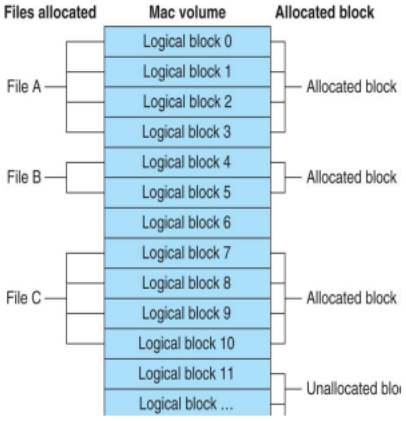
- **Volume:**

- Any storage medium used to store files.
- It can be all or part of the storage media for hard disks.

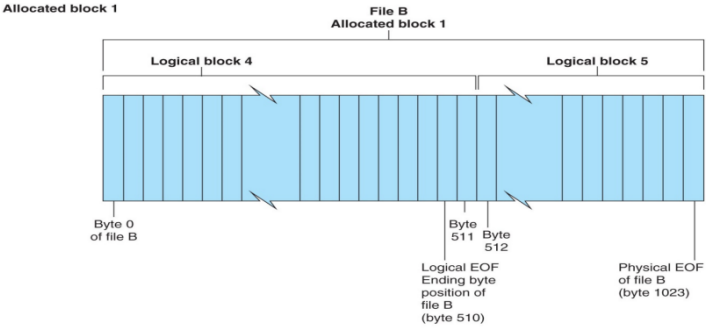


Here's an overview of Mac file structures:

- **Volumes:**
  - Consist of allocation and logical blocks.
  - Logical blocks cannot exceed 512 bytes.
  - Allocation blocks are a set of consecutive logical blocks.



- **End of File (EOF) Descriptors:**
  - Logical EOF: Actual ending of the file.
  - Physical EOF: Number of bytes allotted on the volume for a file.



- **Clumps:**
  - Groups of contiguous allocation blocks to reduce fragmentation.
- **File System Components (Older Macintosh OSs):**
  - Boot Blocks:
    - First two logical blocks (0 and 1).
  - Master Directory Block (MDB) or Volume Information Block (VIB):
    - Stores all information about a volume.
  - Volume Control Block (VCB):
    - Stores information from the MDB when OS mounts.
- **Extents Overflow File:**
  - Stores any file information not in the MDB or a VCB.
- **Catalog:**
  - The listing of all files and directories on the volume.

Differences between Linux and macOS file systems:

1. **Directory Structure:**
   - Linux:
     - `/home/username`
     - `/root`
   - macOS:
     - `/Users/username`
     - `/private/var/root`
     - `/home` directory exists but is empty in macOS.
2. **User Access:**
   - macOS users have limited access to other user accounts' files.
   - The guest account is disabled in macOS.

Forensic View of macOS:

- **Understanding File System Components:**
  - Know the location of file system components and how files and file components are stored.
- **Application Settings:**
  - Stored in three formats: plaintext, plist files, and SQLite database.
  - Plist files are preference files for installed applications on a system.
- **FileVault:**
  - Used to encrypt and decrypt a user's `/Users` directory.

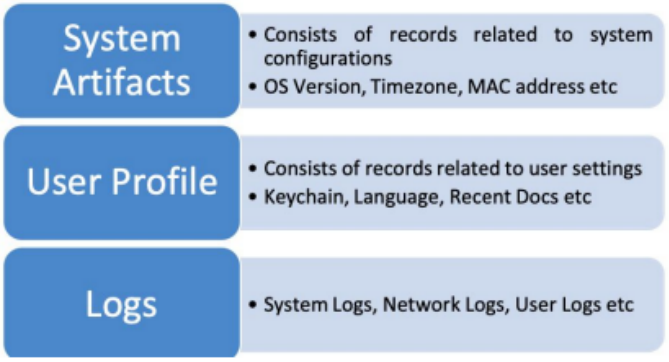
Files useful for investigation:

- `/System/Library/CoreServices/SystemVersion.plist`:
  - Contains the OS version.
- `/Library/Preferences/SystemConfiguration/NetworkInterfaces.plist`:
  - Shows all existing network interfaces.

    - Recent interface usage is listed in `/private/var/db/dhcpclient/leases`.
- `/private/var/db/DirectoryService/flatfile.db`:
    - List of users on a system (used before Mac OS X v10.7).
    - Similar to the Linux/UNIX `/etc/passwd` file.
- `/private/var/db/dslocal/nodes/Default/users`:
    - Contains users' plist files in Mac OS X after v10.7.
- `/private/var/db/shadow/hash`:
    - Contains account passwords.

Forensic Artifacts

Artifacts are useful objects or area within a computer system that hold useful information about various activities performed by a user on the computer system, and these artifacts differ from one OS to another



System artifacts consist of records related to system configurations like OS version, MAC Address, Time Zone, etc.

OS Version	• /System/Library/CoreServices/SystemVersion.plist
MAC Address	• /private/var/log/daily.out
Timezone	• /Library/Preferences/.GlobalPreferences.plist
Language	• /Library/Preferences/.GlobalPreferences.plist
Start-up	• /Library/LaunchAgents/
Folders	• /Library/LaunchDaemons/ • /System/Library/LaunchAgents/ • /System/Library/LaunchDaemons/

User Profiles These files contain data related to user activity on a system.

User Folder (Default)	• Desktop files -- ~/Desktop/ • Download folder -- ~/Downloads/ • Library -- ~/Library/ • Document folder -- ~/Documents/ • Deleted files -- ~/.Trash/
Recent folders	• ~/Library/Preferences/com.apple.finder.plist
DOCK – Persistent apps	• ~/Library/Preferences/com.apple.dock.plist
Recent Documents	• ~/Library/Preferences/com.apple.recentitems.plist
Safari Browsing History	• /username/Library/Safari/History.plist
Apple Mail	• Desktop/Library/Mail
USB devices	• /private/var/log/system.log

Keychain in macOS Forensics:

- **Keychain Overview:**
  - macOS has its own password management system called Keychain.
  - Stores sensitive information such as user credentials, passwords, certificates, and other secure entities.
- **Keychain Files:**
  - macOS uses a Keychain file to store credentials used by the operating system.
  - Additional file for each user in the system.
  - Keychain encrypts and stores passwords, while secure notes on other entities are in plain text.
- **Keychain Access Application:**
  - Mac application used to manage and restore passwords stored in Keychain.
- **Deleted Files:**
  - Deleted files are moved to the Trashes folder.
  - Files deleted at the command line do not appear in the trash.

System Keychain Contains:

- Apple ID and Password
- Wi-Fi passwords
- VPN, FTP, and SSH passwords
- Passwords to iTunes backup
- Passwords to social networks
- iWork document passwords
- AirPort and TimeCapsule passwords
- Passwords to mail accounts
- Passwords to social networking websites

Logs

Mac also stores logs of system and user activity.

System Logs	• /private/var/log/asl/YYYY.MM.DD.U[XX].asl • /private/var/log/DiagnosticMessages/YYYY.MM.DD.asl • /private/var/log/system.log • /private/var/log/zzz.log
Shutdown Logs	• /private/var/log/com.apple.launchd/launchd- shutdown.system.log
Network Status	• /private/var/log/daily.out
Bootup Time	• /private/var/log/System.log (find 'BOOT_Time')
Filesystem Logs	• ~/Library/Logs/fsck_hfs.log
VMWare Logs	• /Library/Logs/VMWare

File Systems in macOS:

- **HFS (Hierarchical File System):**
  - Introduced in 1984 with the original Macintosh.
- **HFS+ (Hierarchical File System Plus):**
  - Introduced 13 years later.
  - Served as a major file system upgrade for the Mac and became the primary file system.
- **Apple File System (APFS):**

- Introduced in 2016 with macOS High Sierra, replacing HFS+.
- Optimized for SSDs in macOS with encryption as its primary feature.
- New features include:
  - Snapshot
  - Copy-on-write metadata
  - Space sharing
  - Fast directory sizing
  - Cloning for files and directories
  - Automatic safe-save

Characteristics of Apple File System (APFS):

- Supports cloning of files and directories, allowing efficient file copies on the same volume without occupying additional storage space.
- Supports the snapshot feature to capture the state of a system, creating a read-only instance of the file system.
- Uses a 'copy-on-write' metadata scheme to ensure that updates to the file system are crash-safe.
- Supports TRIM operations.
- Allows space sharing by having multiple logical drives in the same container, with free space available to all volumes in that container.
- Supports full disk encryption with options for single-key encryption or multi-key encryption models for each volume in a container.
- Uses AES-XTS or AES-CBC encryption methods, depending on the hardware.
- The multi-key encryption model ensures user data integrity, even when a device's physical security is compromised.

Challenges in macOS Forensics:

- **Strong Encryption Standards:**
  - Apple's implementation of stronger encryption standards on macOS and iOS creates secure environments but poses challenges for forensic investigations.
- **Secure Delete Feature:**
  - Apple's secure delete feature allows Mac users to overwrite a system's free space, making data recovery nearly impossible.
- **FileVault:**
  - FileVault, an in-built feature in macOS, provides users with a secure location to store their data.
  - Accessing data in FileVault requires bypassing encryption or obtaining the password, making it inaccessible to forensic examiners unless FileVault is disabled.
- **iCloud Backup:**
  - Apple's iCloud allows users to back up their device data to the cloud.
  - All users with an iCloud account have an Apple ID, enabling them to upload and download data from iCloud and sync all their macOS devices.
  - Obtaining the Apple ID and password provides access to data associated with all synced devices, potentially providing access to a vast amount of information.

Information to Collect During MacBook Forensics Investigation:

- **Case Background/Bring Your Own Device (BYOD) Policy:**
  - Determine if any evidence needs to be acquired and analyzed according to the organization's policy.
- **Device Details:**

        - Make, model, capacity, etc.
        - Decryption key/password to decrypt the hard drive before/after handing it over for imaging.
    - **Access Credentials:**
        - Admin username and password.
        - FileVault Password or Recovery Key (if enabled) for unlocking the device.
    - **Original Charger:**
        - Ensure the original charger of the device is collected.
    - **iCloud Credentials:**
        - Apple ID and Password for extraction of the recovery key from iCloud.
    - **Adapter for Latest MacBooks:**
        - Multiport adapter to connect USB hard drive and charge the device (for MacBooks with only one USB C port).
    - **Boot Settings for Apple T2-based MacBooks:**
        - Disable secure boot and enable booting from external media before handover.
    - **File System Information:**
        - Determine the file system (HFS, HFS+, APFS) of the machine.

Tools for MacBook Forensics:

Acquisition and On-site Verification:

- MacQuisition
- Guymager (Kali/CAINE)
- OSXpmem
- OSXcollector
- FTK cli
- Blacklight
- Arsenal Recon (Image mounting)
- APFS for Windows – Paragon Software (Image mounting)
- Plaso (Open Source) for timeline analysis
- Plist Viewer (OSForensics)

Imaging Tools:

- Guymager
- MacQuisition

MacQuisition:

- Intuitive user interface for forensically acquiring a bit-by-bit image of a Mac device.
- Features include:
    - Easy identification of source device(s).
    - Configuration of destination location.
    - Command-line option for advanced examiners.
    - Case, exhibit, and evidence tracking and notes.
    - Automatic generation of MD5, SHA1, and SHA 256 hashes.
    - Advanced features for hash and block customization, along with extension naming options.
    - Two fast, compact flash readers, UDMA 1394a adapter, and USB 2.0.

Guymager:

- Open-source tool for acquiring a device and creating a forensic image.
- Features include:
  - Simple user interface.
  - Multithreaded data compression and pipelined design for faster and more reliable operation.
  - Generates dd, E01, and AFF image formats.

Network Forensics

It involves monitoring, recording, analyzing, and interpreting network traffic.

This language of networking is called Protocol.

A Protocol is a set of rules that defines how communication should be carried out.

Protocols can be categorized as:

- **Public Standards:**
  - Have a known format and can be used by anyone.
  - Often encountered on the Internet.
- **Proprietary:**
  - Owned by a company.
  - Often encountered when dealing with a vendor's product.

Communication Complexity:

- With networks as complex as those we use today, it's necessary to use and combine multiple protocols to achieve efficient communication.
- No network relies on a single protocol; instead, they rely on a stack of protocols where each protocol defines one stage of the communication.

1. Protocol Analysis:

Protocol analysis involves:

- Examining packets belonging to a certain protocol to understand how it works.
- Ranges from examining fields for well-known protocols to analyzing packets of unusual protocols.

Challenges:

- Products may not fully follow protocol standard documentation.
- Preparatory protocols may not have official documentation, requiring reliance on reverse engineering attempts.

Tools for Protocol Analysis:

- **Packet Details Markup Language (PDML):**
  - XML-based language describing packets from the application layer to the data link layer.
  - Easy to parse and analyze for extracting data.
- **Wireshark:**
  - Essential for network administrators and security professionals.
  - Automatically decodes captured packet bytes.
  - Allows writing custom decoders (custom dissectors) for custom protocols.
- **tshark:**
  - Command-line interface tool similar to Wireshark.
  - Shares many features with Wireshark but lacks a graphical interface.

2. Flow Analysis:

Flow analysis is the process of:

- Analyzing a group of related packets.
- Looking for patterns and/or anomalies.
- Helps reconstruct incidents and the data exchanged during those events.

Key Points:

- Packet analysis involves examining separate packets, regardless of their relationship to each other.
- Flow analysis focuses on analyzing related packets to understand the complete picture of the communication.

Challenges:

- Related packets may not be sequential in the captured traffic.
- Computers may not send packets belonging to the same flow sequentially.

Example:

- One common example of flow analysis is the "Follow TCP/HTTP Stream" feature in Wireshark.
- This feature shows all messages exchanged during a TCP session, providing a complete picture of the connection without the need to manually piece together individual packets.

3. File Carving & Data Extraction:

File carving is the process of:

- Extracting useful data or application-level protocol payloads from a stream/flow of packets.
- Can be done automatically or manually.

Key Points:

- The goal of analyzing flows and packets is to extract information, which could be:
  - Simple protocol fields such as Source IP or destination Port.
  - Files transferred using application-layer protocols such as FTP or HTTP.

Process:

- Identify the flow containing the file of interest.
- Look for the signature of the file within the flow.
- Export the bytes of the file from the PCAP file to a separate file.

Tools:

- **Manual Extraction:**
  - Tools such as Wireshark can be used to manually extract files.
- **Automatic Extraction:**
  - Tools like xplico or TCPextract can automatically analyze a PCAP file and extract files from it.

4. Statistical Flow Analysis:

Statistical flow analysis is the process of:

- Combining network flow analysis with statistical methods to examine the full picture of network activities.

Key Points:

- Enables effective examination of network activities by combining network flow analysis with statistical methods.
- Requires logging features enabled on main networking devices to log information about every packet going through.
- Details about network flow are usually stored in what is called flow record.

Applications:

- **Improving Network Performance:**
  - Gathered data can be used to enhance network topology for better quality of service.
- **Forensic Investigation:**
  - Identify infected machines within a network by analyzing statistical patterns.
  - Useful in data exfiltration cases to confirm or deny the existence of data leakage.
- **User Identification:**
  - Unique patterns associated with each user on the network can be identified.

Flow Records:

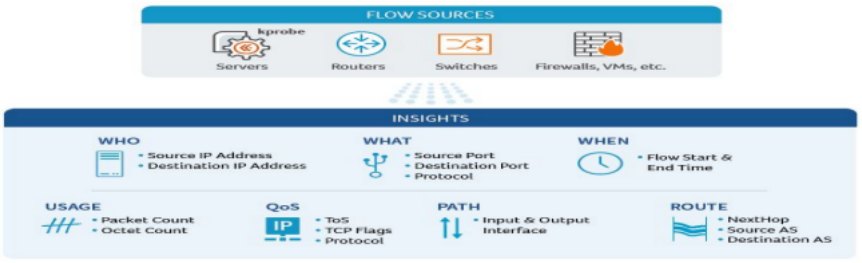
- Contains information such as source and destination, port numbers, date and time, and upper layer protocol.

Components of Statistical Flow Analysis:

1. **Sensor:**
   - Device (e.g., router) working as a gateway for a network segment, responsible for creating flow records.
2. **Collector:**
   - Server used to store flow records.
3. **Aggregator:**
   - Central node connecting different collectors together to avoid decentralization.
4. **Analyzer:**
   - Reads and analyzes flow records.

Flow Record Formats:

- There is no universal format to represent flow records. Various proprietary and open-source solutions use different formats.
- **Cisco Netflow:**
  - Developed for billing purposes.
  - Most Cisco routers can be turned into Netflow sensors with a few commands.



- **Other Flow Record Formats:**
  - **IPFIX (Netflow V10):**
    - Considered the successor of Netflow.
  - **S-flow:**
    - Standard protocol adapted by the Internet Engineering Task Force.
    - Offers advanced statistical features over Cisco's Netflow.

Data Storage:

- Netflow doesn't store the entire packet, only information about the packet header.
- Reduces storage space required for flow records. For example, a flow record for 8 GBs of packets needs only 30 MBs for storage.

Evolution of Netflow:

- Netflow has evolved from a simple billing protocol to a powerful statistical tool for performance and monitoring purposes.

Challenges:

- **Choosing Sensor Placement:**
  - Selecting the right device to install the sensor is challenging.
  - Installing at a very high point in the network hierarchy may result in overload and unnecessary data to filter.
  - Installing at a low point (network access device) may cause the sensor to miss relevant data for investigation.

Challenges of Network Forensics:

- **Volatility of Network Data:**
  - Network forensic evidence tends to be more volatile and harder to acquire compared to other forensic fields.
- **Locating Evidence:**
  - Due to the distributed nature of computer networks, it may be challenging to locate the best source of evidence.
- **Fragmented Evidence:**
  - Investigators may need to collect and piece together evidence from different parts of the network to get a complete picture.
- **Limited Storage on Networking Devices:**
  - Networking devices such as switches and routers do not have large storage capacities and do not store the packets that pass through them.
- **Difficulty in Seizing Devices:**
  - Removing or seizing a networking device is more challenging compared to other types of devices.
  - For instance, removing a hard disk from a computer is simpler in terms of business continuity compared to removing a router from a network.

Network Evidence Acquisition:

Sources of Network Evidence:

1. **Transmission Medium:**
   - Packet may traverse different mediums such as uTP, fiber optics, coaxials, or wirelessly.
   - Techniques and tools for acquiring packets vary depending on the medium.
   - Wiretapping is used to acquire data from wired networks, while wireless networks are easier to monitor.
2. **Encryption:**
   - Encryption of frames can pose a challenge for investigators.
   - Even if frames are encrypted, investigators may still extract useful forensic data from headers.
   - IPsec ESP encapsulates the whole packet, only the data link layer header is retrievable.
3. **Network Infrastructure Devices:**
   - Hubs and switches are common networking devices used to acquire evidence.
   - Hubs:
     - Frames on a hub network are delivered to every machine on the network.
     - Promiscuous mode on a network interface card can be used to collect frames.
   - Switches:
     - Span ports and port mirroring are used to copy frames going through a switch.
     - Port mirroring is a feature that can be enabled through the switch operating system.

    - Port mirroring allows the administrator to specify source ports and a destination port.
    - CAM (content addressable memory) stores MAC addresses and corresponding port information.
    - Switch configuration information such as VLAN tables and port security tables can be valuable for investigations.
    - Another way to ensure you receive all frames going through a switch, is to overload the switches Cam table with fake Mac addresses. Once the table is filled with Mac addresses that do not exist, the switch won't be able to know which ports the actual devices are connected to. Once that happens, the switch will start broadcasting all the frames over the network.
4. **Routers:**
    - Routers provide information on the path a packet took within the network.
    - Routers can also be configured to work as an access controller for the network using network access lists (ACLs).
    - Routing tables and configurations may contain relevant evidence.
    - Routers can export logs to remote servers using protocols like Netflow.
5. **DHCP Servers:**
    - DHCP server logs provide information on IP address assignments.
    - IP addresses are crucial for identifying devices used by attackers.
    - DHCP logs can help identify devices by their MAC addresses.
6. **DNS Servers:**
    - DNS server logs are valuable for investigations.
    - DNS requests can help identify machines trying to connect to specific websites or servers.

Challenges:

- **Volatility of Data:**
    - Networking devices have limited memory, and evidence such as CAM and routing tables are volatile.
    - Switches may clear CAM tables when they run out of memory.
- **Complexity of Environment:**
    - Network environments are more complex compared to other environments, making evidence acquisition challenging.
- **Encryption:**
    - Encrypted frames can hinder the extraction of payload and origin information.
- **Device Seizure:**
    - Seizing networking devices is more difficult compared to other devices like computers due to business continuity concerns.

Berkeley Packet Filter (BPF):

- **Purpose:**
    - Helps separate useful packets relevant to the investigation from noise packets.
    - Enables the selection of specific frames to capture, reducing resource consumption.
- **Functionality:**
    - **Raw Interface:**
        - Provides a raw interface to data link layers.
        - Allows raw link-layer packets to be sent and received.
    - **Promiscuous Mode:**

        - If the network interface driver supports promiscuous mode, BPF allows the interface to receive all packets on the network, even those destined for other hosts.
    - **Filtering:**
        - Supports packet filtering, enabling a user space process to specify which packets it wants to receive.
        - For example, a tcpdump process may only want to receive packets that initiate a TCP connection.
- **Advantages:**
    - BPF returns only packets that pass the filter specified by the process, avoiding the copying of unwanted packets from the operating system kernel to the process and greatly improving performance.
- **Implementation:**
    - Seen in action when using tools like Wireshark.
    - Before starting the capture, a BPF can be inserted to specify the type of package to capture.

Forensic Footprints in Network Forensics:

- **Scouring the Internet:**
    - Investigators scour the internet to obtain tracks of the hacker/attacker.
- **Packet Analysis:**
    - Data travels in the form of packets in cyberspace.
    - Packets contain valuable information such as source, destination, and contents.
    - Investigators analyze these packets for traces left by hackers/attackers.
- **Logging Feature:**
    - Almost all network devices come with a logging feature.
    - Traffic passing through the device gets digitally logged.
- **Network Log Mining:**
    - Process of extracting logs from networking devices.
    - Involves identification, extraction, arrangement, and examination of log data.
- **Packet Analysis:**
    - Single packets are studied for details in captured traffic analysis.
    - Helps determine whether the traffic is generated via a genuine source or was created via bots.
- The process of extracting logs from networking devices is known as network **log mining.**
- In the event that a networking-related crime hacker/attacker might have left some traces, investigators need to analyze these. Such traces are also called **footprints.**

Seizure of Networking Devices:

- **Handling Networking Devices:**
    - Networking devices contain crucial data useful in cybercrime investigations.
    - These devices are sturdy and durable.
- **Steps to Investigate Networking Devices:**
    1. Switch off the device and power supply.
    2. Disconnect cables and pack the device in anti-static material.
    3. Fill out the chain of custody form and document the chronological history of electronic evidence.

Forensic Examination of Networking Devices like Firewalls:

- **Information to Collect:**
    - Traffic allowed and blocked on the firewall.

- Bandwidth and protocol usage, high CPU usage, and exceeding limits.
- Bytes transferred (large files).
- Detected attack activities, such as attacks from specific sources.
- Administrator access, including login failed attempts.

Forensic Techniques:

- **Session Identification:**
  - Analyzing collected logs to understand how attackers gained access to the network.
- **Pattern Discovery and Analysis:**
  - Attempting to crack the pattern of an attacker.
  - Includes resolution and backtracing.
  - *Resolution:*
    - Extracting salient rules, patterns, and statistics by eliminating irrelevant data.
  - *Backtracing:*
    - Reconstructing an event from the end to the start.

Network Forensic Artifacts:

- **DHCP (Dynamic Host Configuration Protocol)**
- **NTP (Network Time Protocol)**
- **DNS (Domain Name Server)**
- **Web Proxy logs**
- **Firewalls**
- **IDS (Intrusion Detection System) and IPS (Intrusion Prevention System)**
- **Evidence from software-based firewalls and mail clients like MS Outlook, Outlook Express, Eudora, etc.**

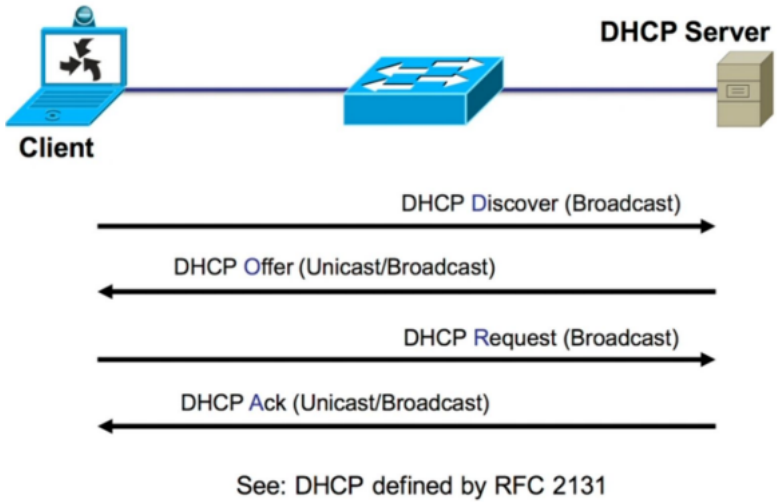
Network Attacks:

- **Characteristics:**
  - Network attacks are often hardware and software independent, targeting networking protocols themselves.
  - They can occur on different layers of the OSI model.
- **Types of Network Attacks:**
  - **Application Layer Attacks:**
    - Rogue DHCP and DNS servers.
  - **Transport Layer Attacks:**
    - Denial-of-service attacks like SYN-Flood.
  - **Network Layer Attacks:**
    - IP spoofing and ICMP redirection.
  - **Data Link Layer Attacks:**
    - ARP poisoning, port stealing, and MAC flooding.

DHCP Starvation Attack:

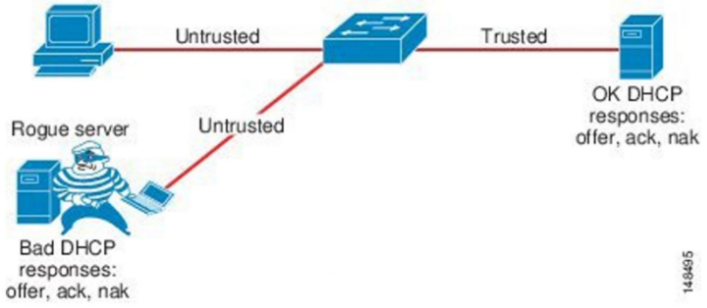
- **Overview:**
  - DHCP (Dynamic Host Configuration Protocol) automates the IP address assignment process.
- **Attack Description:**
  - In DHCP starvation, attackers perform the DHCP IP request process multiple times with spoofed MAC addresses until they consume all available IP addresses in the pool.

- **Detection:**
  - Signs of DHCP starvation attack include many DHCPDISCOVER messages requested from non-existing MAC addresses.
- **Prevention:**
  - Network administrators can prevent such attacks by using port security and DHCP rate limiting.



Rogue DHCP Server Attack:

- **Description:**
  - In this attack, a fake DHCP server is installed within the network.
- **Impact:**
  - The rogue DHCP server provides clients with incorrect IP configurations, including IP address, default mask, and default gateway.
- **Example:**
  - If the attacker sends their IP address to clients as the default gateway, the clients will send internet traffic to the attacker.

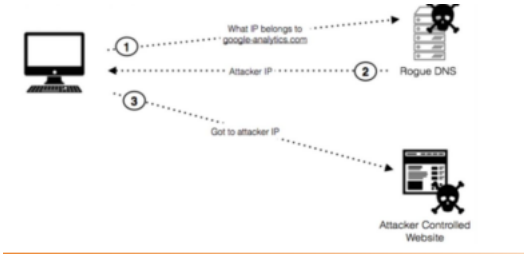


  - **Prevention:**
    - Rogue DHCP server attacks can be prevented by activating DHCP snooping on network switches.
    - DHCP snooping allows administrators to specify which port on the switch is allowed to relay DHCP offer messages, preventing attackers from sending fake DHCP offers from an untrusted port.

  Rogue DNS Server Attack:

  - **Description:**
    - In a rogue DNS server attack, the attacker installs a rogue DNS server and sends spoofed answers to victims' DNS queries.

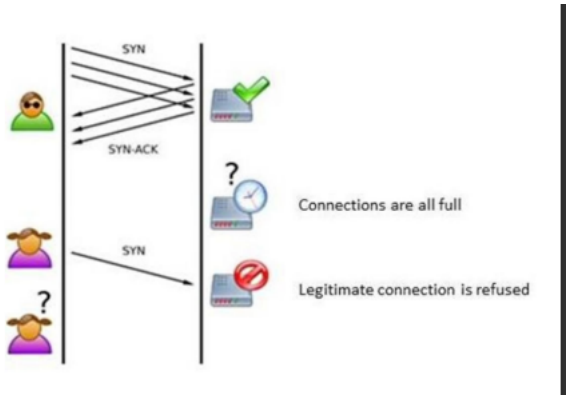
- **Impact:**
  - Attackers can redirect victims to a website of their choosing.
  - For example, attackers can redirect victims trying to log in to their bank accounts to a fake login page, stealing their credentials.



Transport Layer Attacks:

- **Description:**
  - Attacks target transport layer protocols like TCP and UDP.
- **Examples:**
  - Denial-of-service attacks like SYN flood.
  - Session hijacking attacks where the attacker impersonates the victim's client.

SYN Flood Attack:



- **Description:**
  - SYN flood attacks exploit the three-way handshake mechanism in the TCP protocol.
- **Execution:**
  - Attackers send various SYN segments with fake source IP addresses to the victim's machine's port.
  - When the SYN message is received, the victim's application starts allocating memory and processing time for the new communication.
- **Impact:**
  - Continuous SYN flooding consumes the victim's machine's resources, making it harder for new clients to connect.

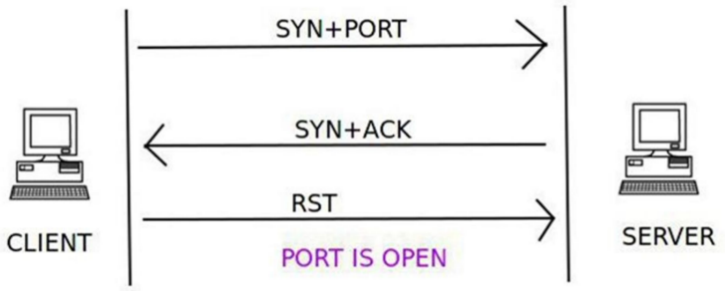
Port Scanning:

- **Overview:**
  - Port scanning is often the last phase before an actual attack.
- **Legality:**
  - Conducting unauthorized port scanning can lead to legal consequences.
- **Purpose:**

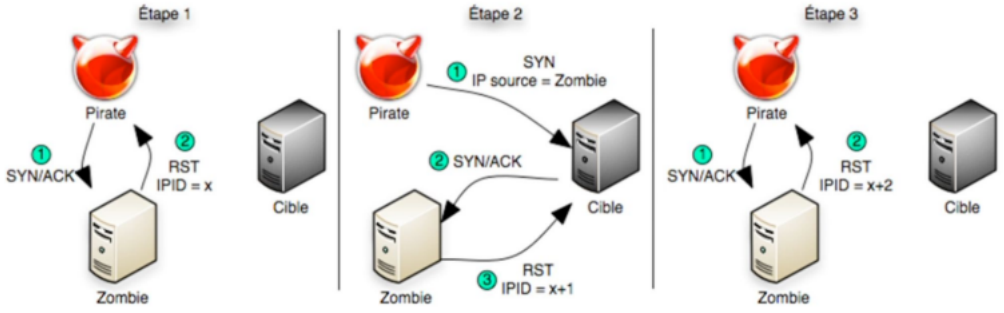
- Port scanning allows attackers to determine which ports are open on the victim's machine, providing them with information for a potential attack.

- **Recommendation:**
  - Network investigators should be familiar with various port scanning techniques.

Port Scanning Techniques:



1. **Full TCP Scan:**
   - Description: Completing a full three-way handshake with the desired port.
   - Method: Attacker completes the three-way handshake and terminates the session immediately.
2. **Half Open Scan (SYN Scan or Stealth Scan):**
   - Description: Attacker does not complete the three-way handshake, terminating the session after receiving the ACK message.
   - Purpose: Faster than a full TCP scan.
3. **Zombie Scan:**
   - Description: Attacker impersonates a third machine on the network and probes ports using spoofed IP addresses.
   - Method:
     - Attacker can spoof the reply coming from the victim to the zombie machine.
     - Alternatively, attacker can ping the zombie machine and calculate the IP ID differences to determine whether the zombie replied to the victim's SYN-ACK.
   - Result: Network administrator may attribute the scanning to the zombie machine instead of the attacker.



TCP Session Hijacking Attack:

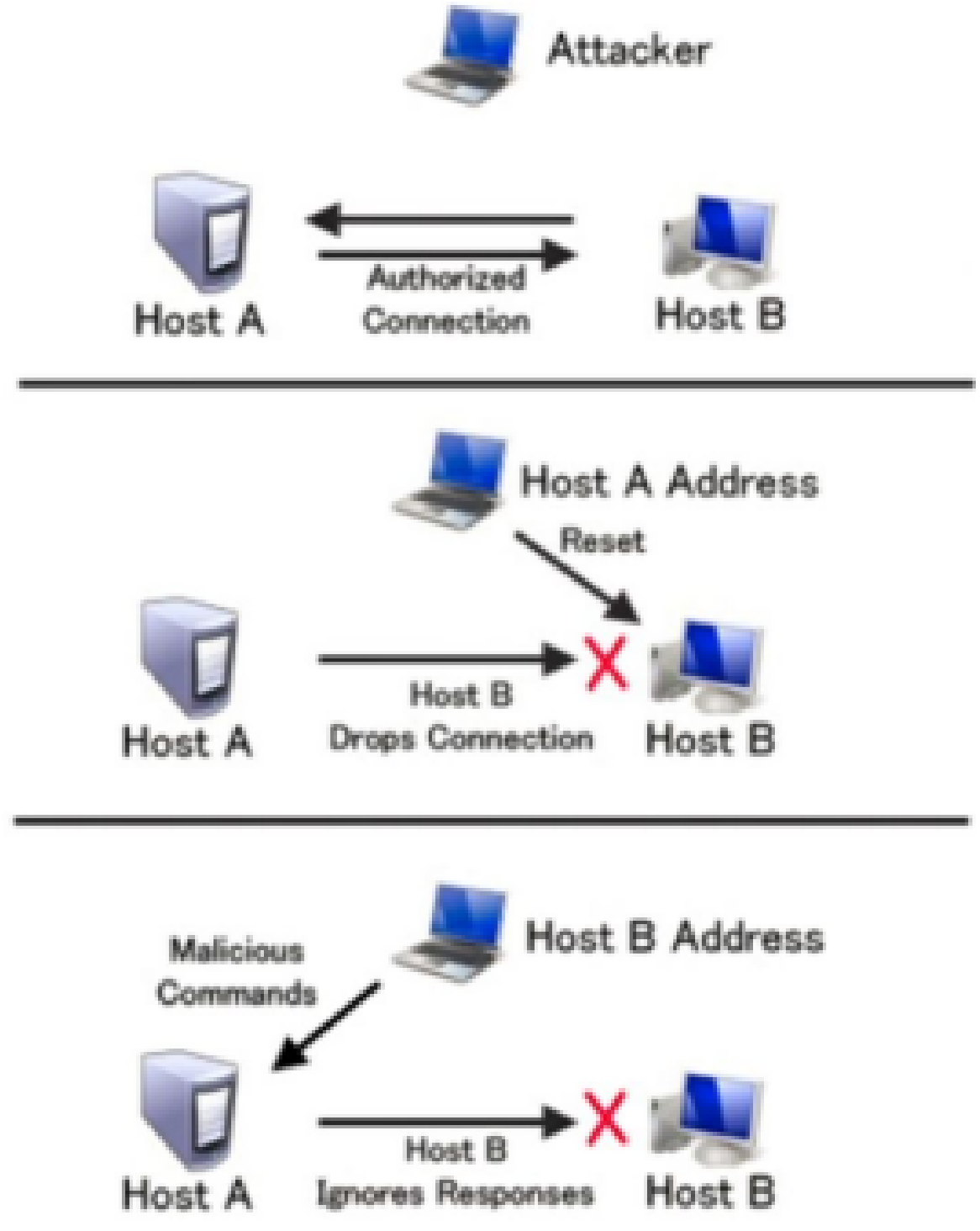


Figure 2: Session Hijacking

- **Description:**
  - Attacker acts as a man in the middle to desynchronize the connection between the victim and the server.
  - Exploits TCP's reliance on acknowledgment number and sequence number to maintain the connection.

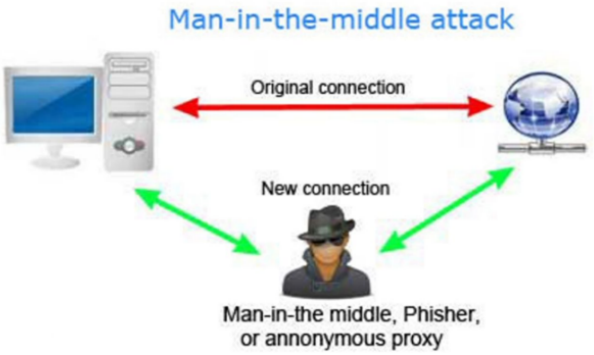
- **Execution:**
  - Attacker destabilizes the TCP session by sending a spoofed segment.
  - Once the connection is destabilized, attacker can send messages with the correct sequence and acknowledgment number on behalf of the victim.

Data Link Layer Attacks:

- **MAC Flooding:**
  - Description: A denial-of-service attack conducted at the data link layer.
  - Method:
    - Every MAC address on every port uses a table called the CAM table.
    - When the CAM table gets full, the switch drops the table and starts filling it again.
    - Attacker sends a huge number of frames with fake MAC addresses, filling the CAM table and forcing the switch to drop and rebuild it.
  - Result: Switch behaves like a hub, forwarding frames to all ports on the network.

ARP Poisoning Attack:

- **Description:**
  - ARP (Address Resolution Protocol) poisoning is a common attack conducted on local networks.
  - Exploits the trusting nature of the ARP protocol within the network.
- **Execution:**
  - Attacker becomes a man in the middle between the router and the clients by poisoning ARP tables.
  - Allows the attacker to intercept, view, and extract unencrypted data transmitted between the router and the clients.



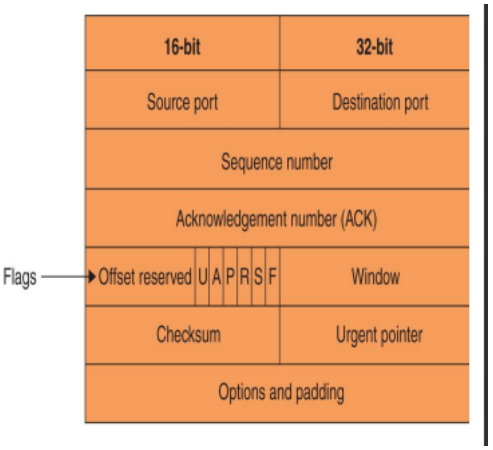
Attack	Layer	Description	Prevention
Rogue DHCP Server	Application	Attacker installs a rogue DHCP server to provide incorrect IP configurations to clients.	Use DHCP snooping on network switches.
Rogue DNS	Application	Attacker installs a rogue DNS server to redirect victims to malicious websites.	Use DNSSEC, DNS filtering, and DNS monitoring.
SYN Flood	Transport	Floods victim with SYN segments to exhaust its resources.	Implement SYN cookies, rate limiting, and firewalls.
Port Scanning	Transport	Determines open ports on a victim machine.	Use firewalls, intrusion detection/prevention systems.
TCP Session Hijacking	Transport	Attacker impersonates client or server to disrupt TCP session.	Use encryption, strong authentication, and intrusion detection/prevention systems.
MAC Flooding	Data Link	Floods switch's CAM table with fake MAC addresses, turning it into a hub.	Use port security, dynamic ARP inspection, and switch port authentication.
ARP Poisoning	Data Link	Attacker exploits ARP protocol to become a man-in-the-middle, intercepting network traffic.	Use ARP spoofing detection tools, static ARP entries, and VLAN segregation.

Network Tools

Tool	Description
Splunk	Log analysis platform for searching, monitoring, and analyzing machine-generated big data.
Spiceworks	IT management software including network monitoring, help desk, inventory, and more.
Nagios	System, network, and infrastructure monitoring software.
Cacti	Network graphing solution designed to harness the power of RRDTool's data storage and graphing functionality.

Packet Analyzers

- **Definition**: Devices or software that monitor network traffic.
- **Layers**: Most work at layer 2 or 3 of the OSI model.
- **Functionality**: Some perform packet captures, some are used for analysis, and some handle both tasks.
- **Formats**: Most tools follow the Pcap (packet capture) format.
- **Combination**: Many investigators capture with tcpdump and then analyze the capture in Wireshark.
- **Identification**: Packet analyzers can identify packets by examining the flags in their TCP headers.



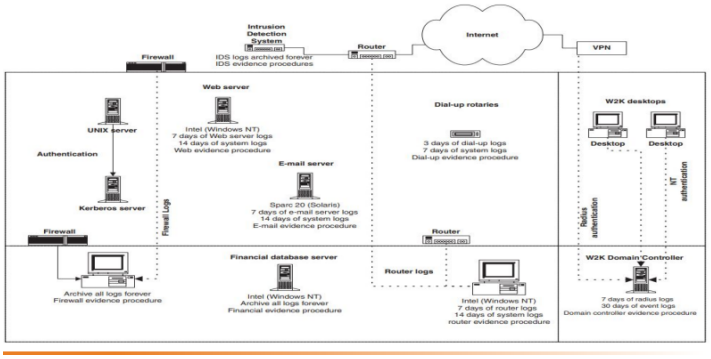
Tools

Tool	Description	Purpose
Tcpslice	Extracts information from large Libpcap files	Extraction of specific data from large network capture files
Tcpreplay	Replays network traffic recorded in Libpcap format	Testing network devices such as IDSs, switches, and routers with recorded network traffic
Etherape	Graphically views network traffic	Visualization of network traffic in a graphical format
Netdude	GUI tool for inspecting and analyzing large tcpdump files	Easy-to-use interface for analyzing large tcpdump files
Argus	Real-time flow monitor for security, accounting, and network management	Session data probe, collector, and analysis tool
Wireshark	Real-time network packet analyzer and capture tool	Capturing and analyzing network packets in real-time
Network Miner	Packet capturing tool/passive network sniffer	Detects operating systems, open ports, sessions, etc., without sending traffic on the network
Xplico	Extracts application data using Port Independent Protocol Identification (PIPI)	Parses and analyzes pcap files to extract application data efficiently

Digital Evidence Map

- **Definition**: A methodical search tool facilitating the identification of digital evidence within a network.
- **Importance**: Helps investigators identify missing evidence and navigate the network efficiently.
- **Benefits**:
  - Helps investigators realize when something is missing from the network.

- Saves time by allowing investigators to quickly locate evidence.
- Avoids the need to spend hours trying to obtain evidence by providing a clear path to the required data.



Digital Evidence Map

- **Purpose**: Facilitates a methodical search of digital evidence.
- **Importance**: Helps identify missing evidence within the network.
- **Benefits**: Enables investigators to locate evidence efficiently.

Incident Response in Network Forensics

- **Preparation**: Organizations need to have a plan, tools, and personnel to effectively respond to a breach.
- **NIST Guidelines**: Incident response life cycle includes preparation, prevention, detection and analysis, containment, eradication and recovery, and post-incident activity.
- **Response Team**: Form a multidisciplinary Computer Incident Response Team including management, information security, IT support, legal, public affairs/media relations, and digital forensics.
- **Outsourcing**: If digital forensics capabilities are lacking internally, outsource this function in advance.

Network Investigation Challenges

- **Identifying Hackers**: Challenging due to spoofed IP addresses, disabled logging functions, and time delays in discovering breaches.
- **Log Availability**: Logs can be deleted intentionally by hackers or may not be generated at all.
- **Jurisdictional Issues**: Trails can cross state, national, and international boundaries, leading to legal and cooperation challenges.

Web Browser Forensics

Web forensics focuses on extracting and analyzing digital evidence related to the user's internet activity.

On the server side, access logs and proxy logs in web servers may prove very useful in tracking a certain user's web activities.

While data such as cash, accessed URLs, searched keywords are looked for on the client side.

Internet forensics include the analysis of both volatile data, such as cached content and temporary files and non-volatile data downloaded file and browser related data files.

To safeguard the privacy of end-users, various web browsers offer a special configuration known as Private Browsing (Firefox) or Incognito Mode (Google Chrome). When activated, this mode allows users to browse the web without storing local data that could reveal their previous web activity on their device.

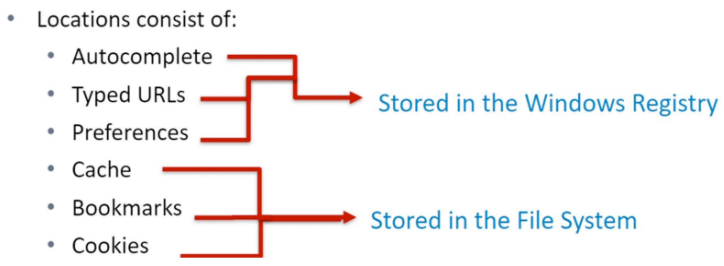
Here are some key features of this mode:

1. **Data Deletion**:
   - Browsing history
   - Cookies
   - Form and search bar entries
   - Download list history
   - Entered passwords
   - Offline web content

   All this information is deleted upon closing the browser.
2. **Tracking Protection**:
   - Tracking protection is activated automatically.
   - It prevents websites from tracking user browsing history across multiple sites.

By combining these features, Private Browsing or Incognito Mode offers users a way to browse the web with increased privacy and security.

Internet Explorer



Windows default web browser, Internet Explorer (IE), comes preinstalled with all versions of Windows. Its main registry key is located at `HKEY_CURRENT_USER\\Software\\Microsoft\\Internet Explorer`.

Under this key, there are several subkeys:

1. **Main**:
   - Location: `HKEY_CURRENT_USER\\Software\\Microsoft\\Internet Explorer\\Main`
   - This key stores IE configuration settings such as the home page, search bar, default search engine, etc.
2. **TypedURLs**:
   - Location: `HKEY_CURRENT_USER\\Software\\Microsoft\\Internet Explorer\\TypedURLs`
   - This key maintains a list of the URLs the user types in the address bar in IE.

index.dat

Internet Explorer (versions 9 and below) uses a file called `index.dat`. This file serves as a database file to improve the overall performance of IE. It indexes various contents such as browsing history, search queries, cookies, and recently opened files to offer a more customized experience for the user.

- **Purpose**:
  - Improve IE performance by indexing browsing history and other contents.
- **Contents**:
  - URLs visited using IE

    - Search queries
    - Cookies
    - Recently opened files
- **Functionality**:
    - Autocomplete web addresses in the browser address bar by retrieving browsing history from `index.dat`.
- **Location (Windows 7)**:
    - `\\Users\\<username>\\AppData\\Roaming\\Microsoft\\Windows\\Cookies\\index.dat`
    - `\\Users\\<username>\\AppData\\Roaming\\Microsoft\\Windows\\Cookies\\Low\\index.dat`
- **More Locations**:
    - Index.dat files in various Windows versions: www.milincorporated.com/a2_index.dat.html

WebCacheV01.dat

Newer versions of Internet Explorer (versions 10 and 11), which come preinstalled with Windows 8 and 10, do not use index.dat files. Instead, they use a file called `WebCacheV01.dat` to store all user browsing information.

- **Purpose**:
    - Store user browsing information.
- **Contents**:
    - Browsing history
    - Cache data
- **Location**:
    - `\\Users\\<username>\\AppData\\Local\\Microsoft\\Windows\\WebCache\\WebCacheV01.dat`
    - or
    - `System32\\Config\\SystemProfile\\AppData\\Local\\Microsoft\\Windows\\WebCache\\WebCacheV01.dat` (after turning on "Show hidden files and folders" in Windows settings)

Microsoft Edge Web Browser

Microsoft Edge (code name Spartan) is the replacement for the Internet Explorer browser and is the default browser for Windows 10.

- **Description**:
    - Lightweight web browser
    - Integrated with Cortana feature in Windows 10
    - Allows users to complete tasks using voice commands
- **Forensics Perspective**:
    - With the transition to Windows 10, more users are expected to use Microsoft Edge instead of Internet Explorer. Therefore, knowing where this browser stores its data is essential for forensic analysis.

Cortana

Cortana is a voice-activated personal assistant, similar to Apple's Siri, developed by Microsoft. It was first introduced in Windows phone version 8.1 and later integrated into Windows desktop with the release of Windows 10.

- **Functionality**:
    - Provides personalized experience for Windows 10 users
    - Offers suggestions during searches
    - Remembers events
    - Sends emails on the user’s behalf

    - Searches the Web
    - Checks weather forecasts, and more
- **Cumulative Learning**:
    - Learns user's habits and attitudes through cumulative learning
    - Provides more accurate results with increased user interaction

Cortana - From a Digital Forensics Perspective

- **Information Stored**:
    - Web searches
    - Geolocation data (latitude/longitude of triggered location-based reminders)
- **Usage Considerations**:
    - Not always enabled on all Windows machines due to privacy concerns
- **Storage Locations**:
    - ESE Databases:
        - `\\Users\\<username>\\AppData\\Local\\Packages\\Microsoft.Windows.Cortana_xxxx\\AppData\\Indexed D\\IndexedDB.edb`
        - `\\Users\\<username>\\AppData\\Local\\Packages\\Microsoft.Windows.Cortana_xxxx\\LocalState\\ESEDatabase_CortanaCoreInstance\\CortanaC`
    - Voice Command Recordings:
        - `\\Users\\<username>\\AppData\\Local\\Packages\\Microsoft.Windows.Cortana_xxxx\\LocalState\\LocalRecorder\\Speech`
    - **Voice Command Recordings Folder**:
        - Stores WAV audio files of voice command recordings issued by a user to Cortana to perform a task.

Firefox

Firefox is a free, open-source web browser developed by Mozilla and is one of the most widely used web browsers globally.

- **Data Storage**:
    - Firefox does not use the Windows registry like Internet Explorer. Instead, it stores its web history, download history, and bookmarks in a central database file named `places.sqlite`.
    - Access Firefox profile: `%APPDATA%\\Mozilla\\Firefox\\Profiles\\`

places.sqlite

- **Contents**:
    - Bookmarks
    - Visited websites
    - Download history

Tools for Forensic Analysis:

- DB Browser for SQLite (http://sqlitebrowser.org): Browse SQLite database tables and their content.
- MZHistoryView (www.nirsoft.net/utils/mozilla_history_view.html): Displays a list of previously visited websites from the `places.sqlite` database.

formhistory.sqlite and cookies.sqlite

- **formhistory.sqlite**:
    - Stores search keywords used in Firefox search bar and searches entered into web forms.
- **cookies.sqlite**:

- Stores cookies planted by visited websites, including login usernames, passwords, and website preferences.

Tools for Forensic Analysis:

- MZCookiesView (www.nirsoft.net/utils/mzcv.html): Displays all cookies stored in a Firefox cookie file; results can be exported into various formats.
- DB Browser for SQLite

key4.db and logins.json

- **key4.db and logins.json**:
  - Stores encrypted passwords.

Tools for Forensic Analysis:

- PasswordFox (www.nirsoft.net/utils/passwordfox.html): Displays usernames and passwords stored by Firefox.
  - Requires supplying master password if set.
  - Can view passwords of another profile by selecting the target profile folder.

Other Firefox Data Locations:

- `permissions.sqlite`: Stores Firefox permissions for individual websites.
- `search.json.mozlz4`: Holds user-installed search engines.
- `prefs.js`: Stores Firefox preferences.
- `addons.json`: Views installed add-ons on Firefox.
- `extension-data` (Folder): Holds data generated by installed extensions (add-ons).

Google Chrome

- **Description**:
  - Fastest and most used web browser on desktop computers worldwide.
  - Based on Chromium, an open-source browser project developed by Google.
- **Chromium-based Browsers**:
  - Many third-party web browsers are based on the Chromium project, such as Vivaldi, Yandex browser, Cent browser, and Opera browser.
  - Investigative techniques used with Google Chrome can often be applied to these Chromium-based browsers.

Data Storage in Google Chrome

- **Configuration Settings and User Data**:
  - Stored in SQLite databases.
  - Databases are files without extensions.
- **Profile Location**:
  - Default Profile:
    - `\\Users\\<username>\\AppData\\Local\\Google\\Chrome\\User Data\\Default`
  - Additional Profiles:
    - Each profile has its own folder.
    - Location: `\\Users\\<username>\\AppData\\Local\\Google\\Chrome\\User Data\\Profile x` (where `x` is any positive integer beginning from 1).

Determining Profile Folder Location

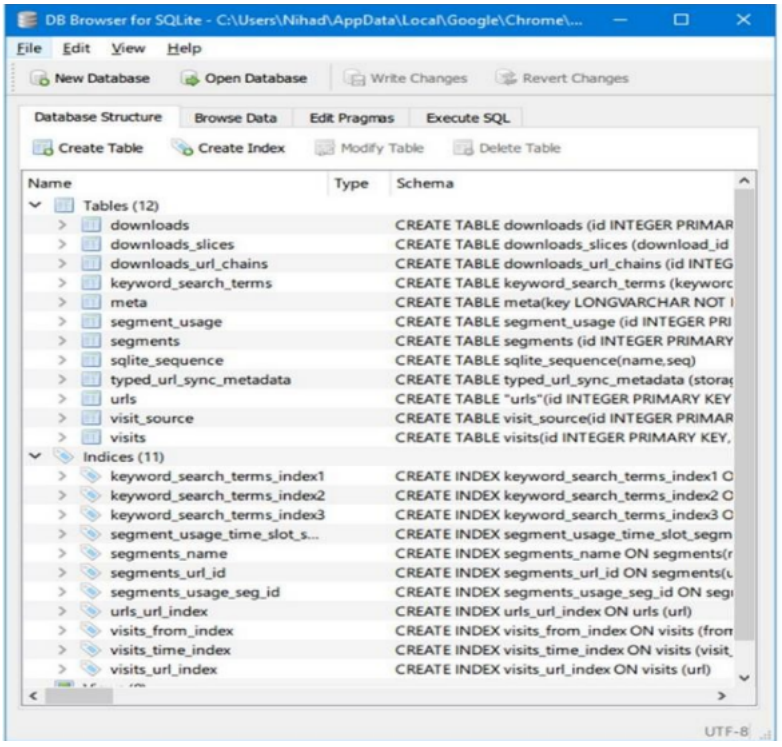
- **Steps**:
  - Open Google Chrome.
  - Type `chrome://version` in the browser address bar and press Enter.

- Check the "Profile Path" in the resulting window.

Google Chrome stores its configuration settings, apps, bookmarks, and extensions in its profile folder. It can have more than one profile, each with its own folder. The default profile can be found at `\\Users\\<username>\\AppData\\Local\\Google\\Chrome\\User Data\\Default`. If there are additional profiles, they can be found at `\\Users\\<username>\\AppData\\Local\\Google\\Chrome\\User Data\\Profile x`, where `x` is any positive integer beginning from 1.

History

- **Description**:
  - Google Chrome stores user browsing history, downloads, keywords, and search terms in the "History" database file located under the Chrome user's profile.
- **Examination Tools**:
  - DB Browser for SQLite can be used to examine the "History" database file.
- **Tables in "History" Database**:
  - There are 12 tables in the "History" database file.



Downloads

- **Description**:
  - Stores information about downloaded files.
- **Examination**:
  - Use the "Downloads" table under the "Browse Data" tab in DB Browser for SQLite.
  - Time information is displayed using Google Chrome value stamps (also known as Webkit format). Use DCode tool to convert it into a readable form.
- **Tools for Examination**:
  - Nirsoft's ChromeHistoryView tool (www.nirsoft.net/utils/chrome_history_view.html) can be used to reveal Chrome history by reading the "History" file of the Google Chrome web browser.

Cookies

- **Description**:

- Google Chrome stores cookies information in the "Cookies" file located under the Chrome user's profile.

- **Examination**:
  - Use DB Browser for SQLite to view the contents of the "Cookies" file and show detailed information about saved Chrome cookies.

These files can be examined using DB Browser for SQLite, and Nirsoft's ChromeHistoryView tool can be used to reveal Chrome history.

Additional Data Files

Top Sites

- **Description**:
  - Stores top websites visited by Google Chrome.
  - Contains two tables: `meta` and `thumbnails`.
  - Information is stored in the `thumbnails` table.

Shortcuts

- **Description**:
  - Supports the autocomplete feature of Google Chrome.
  - Contains two tables: `meta` and `omni_box_shortcuts`.
  - `omni_box_shortcuts` table holds autocomplete text and URLs.

Login Data

- **Description**:
  - Contains three tables: `login`, `meta`, and `stats`.
  - `login` table holds usernames and sometimes encrypted passwords for various websites.
- **Tool for Revealing Passwords**:
  - ChromePass by Nirsoft (www.nirsoft.net/utils/chromepass.html) reveals all usernames and passwords stored by the Google Chrome web browser.

Web Data

- **Description**:
  - Stores login credentials of users (without passwords).
  - Chrome stores login passwords in another file called "Login Data" in newer versions.

Bookmarks

- **Description**:
  - A bookmark (or "favorite") is a URL stored by a user for later retrieval.
  - Bookmarks database file in Google Chrome holds a user's current bookmarks.
  - Contents of this file can be viewed using Windows Notepad.
- **Date/Time Information**:
  - Convert the "date_added" value associated with a bookmark into a readable format using the DCode tool.

Additional Data Files

Bookmarks.bak

- **Description**:
  - Holds recent backups of Chrome bookmarks.

- Gets overwritten periodically each time Google Chrome launches.

- **Forensic Value**:
  - Deleted bookmarks can be found in this file if a suspect deletes them before closing Chrome.
- **Recommendation**:
  - Avoid launching Google Chrome until a copy of this file is saved in a safe location to prevent overwriting.

Cache Folder

- **Description**:
  - Holds frequently accessed static contents like images and parts of HTML files.
  - Helps to load web pages faster by loading parts of contents from the local cache folder instead of downloading from the origin server.
- **Tool for Extraction**:
  - ChromeCacheView by Nirsoft (www.nirsoft.net/utils/chrome_cache_view.html) automates the extraction process of Google Chrome cache.

Forensic View - Google Chrome

- **Information Stored**:
  - Google Chrome stores a lot of personal information about its user.
  - Investigating these artifacts helps in drawing a complete timeline of a user's online activities and understanding their intentions or interests.

Additional Tools for Investigating Web Browsers

1. **WebCacheImageInfo** (www.nirsoft.net/utils/web_cache_image_info.html):
   - Searches and lists all JPEG images with EXIF metadata information stored inside the cache folder of major web browsers (IE, Firefox, and Google Chrome).
2. **ImageCacheViewer** (www.nirsoft.net/utils/image_cache_viewer.html):
   - Scans cache folder in major web browsers and lists all images found inside.
3. **BrowserAddonsView** (www.nirsoft.net/utils/web_browser_addons_view.html):
   - Displays all add-ons/extensions installed on major web browsers (Chrome, Firefox, and IE).
4. **MyLastSearch** (www.nirsoft.net/utils/my_last_search.html):
   - Scans web history in major browsers, cache folder, and retrieves all search queries made previously.
5. **WebBrowserPassView** (www.nirsoft.net/utils/web_browser_password.html):
   - A password recovery tool that reveals passwords stored in major web browsers.
6. **Web Historian** (www.webhistorian.org):
   - Browser extension for Google Chrome to visualize web browsing history stored within Google Chrome. Shows graphical circles of the number of days a website was visited, keyword search terms, most active browsing hours of a day, and days of the week.