

# System Hacking

---

⚡ This chapter has [practical labs](#)

## Goals:

---

1. **Gaining Access** - Uses information gathered to exploit the system
  - Password Attacks:
    - Non-electronic attacks
    - Active online attacks
    - Passive online attacks
    - Offline attacks
2. **Escalating Privileges** - Granting the account you've hacked admin or pivoting to an admin account
3. **Executing Applications** - Putting back doors into the system so that you can maintain access
4. **Hiding Files** - Making sure the files you leave behind are not discoverable
5. **Covering Tracks** - Cleaning up everything else (log files, etc.)
  - **clearev** - Meterpreter shell command to clear log files (issued inside Metasploit Framework)
  - Clear MRU list in Windows
  - In Linux, append a dot in front of a file to hide it

## Password Attacks

---

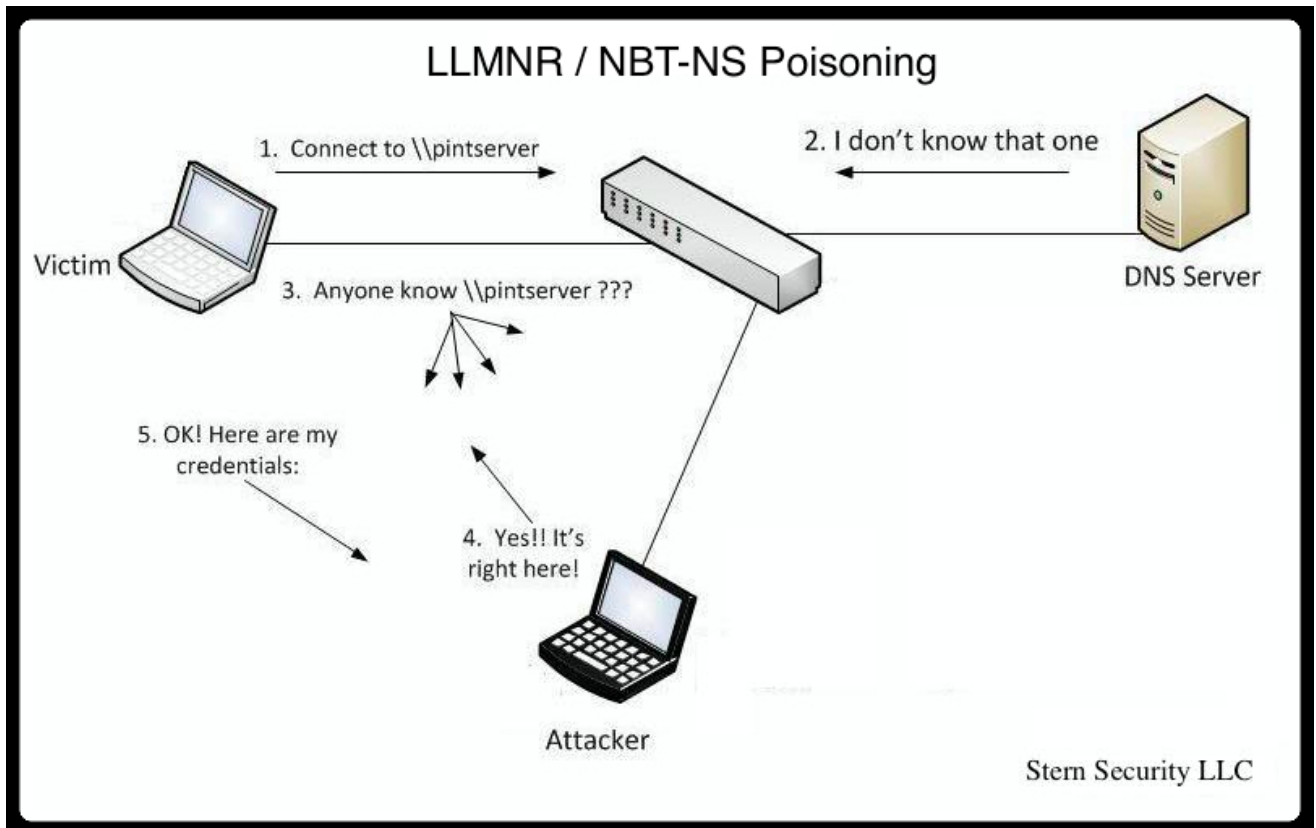
⚡ Check out the practical labs on [Dumping and Cracking SAM hashes \[1\]](#), [Rainbow Tables Basics \[2\]](#) and [LLMNR/NBT-NS \[3\]](#).

### Non-electronic - Non-technical attacks.

- Social engineering attacks - most effective.
- Shoulder surfing
- Dumpster diving
- Snooping around
- Guessing

**Active online** - done by directly communicating with the victim's machine.

- Includes **Dictionary** and **Brute-force attacks**, hash injections, phishing, Trojans, spyware, keyloggers and password guessing
- **LLMNR / NBT-NS Poisoning** - attack based off Windows technologies that caches DNS locally. Responding to these poisons the local cache. If an NTLM v2 hash is sent over, it can be sniffed out and then cracked.
  - ⚡ **LLMNR/NBT-NS practical lab**
  - LLMNR uses UDP 5355
  - NBT-NS uses UDP 137
  - Responder is the tool to sniff the access logs from LLMNR / NBT-NS



- **Keylogging** - process of using a hardware device or software application to capture keystrokes of a user
- Active online attacks are easier to detect and take a longer time
- **Tools for Active Online Attack:**
  - Medusa
  - Hydra
  - NBNSpoof
  - Pupy
  - Metasploit

- **Responder** - **LLMNR and NBT-NS responder**, it will answer to *specific* NBT-NS (NetBIOS Name Service) queries based on their name suffix. By default, the tool will only answers to File Server Service request, which is for **SMB**.
- Can combine "net" commands with a tool such as **NetBIOS Auditing tool** or **Legion** to automate the testing of user IDs and passwords
  - **Tools for NetBIOS attack:**
    - **Hydra**
    - **Metasploit**

## Passive online - Sniffing the wire in hopes of intercepting a password in clear text or attempting a replay attack or man-in-the-middle attack

- **Tools for Passive Online Attack:**
  - **Cain and Abel** - Can poison ARP and then monitor the victim's traffic; Also used for cracking hash passwords (LM, NTLM), sniff network packets for password, sniff out for local stored passwords, etc.
  - **Ettercap** - MITM tool for LAN's, DNS Spoofer; Help against SSL encryption; Intercept the traffic on a network segment, capture passwords, and conduct an active eavesdropping against a number of common protocols.
  - **KerbCrack** - built-in sniffer and password cracker looking for port 88 Kerberos traffic
  - **ScoopLM** - specifically looks for Windows authentication traffic on the wire and has a password cracker



**Services/Protocols that uses Clear text:**

Service	Port
FTP	20/21
TELNET	23
SMTP	25
HTTP	80
POP3	110
IMAPv4	143
NetBIOS	139,445
SNMP	161,162
SQLnet	1521

## Offline - when the hacker steals a copy of the password file (Plaintext or Hash) and does the cracking on a separate system.

- **Dictionary Attack** - uses a word list to attack the password. Fastest method of attacking
  - **Wordlists** - A wordlist or a password dictionary is a collection of passwords stored in plain text. It's basically a text file with a bunch of passwords in it. One popular example of wordlist is the [rockyou.txt](#) containing 14,341,564 unique passwords.
  - You also can generate your own wordlist with given parameters like length, combining letters and numbers, profiling etc.
    - Tools for generate Wordlists:
      - CeWL
      - crunch
- **Brute force attack** - Tries every combination of characters to crack a password
  - Can be faster if you know parameters (such as at least 7 characters, should have a special character, etc.)
- **Hybrid attack** - Takes a dictionary attack and replaces characters (such as a 0 for an o) or adding numbers to the end
- **Rainbow tables** - Uses pre-hashed passwords to compare against a password hash. Is faster because the hashes are already computed.
- **Tools for cracking password files (CLI):**
  - [John the Ripper](#) - Works on Unix, Windows and Kerberos; Compatible with MySQL, LDAP and MD4.
  - [Hashcat](#) - Advanced password recovery tool; Provides several options like hash modes OS's, documents, password managers... (MD5, SHA-family, RIPE-MD, NTLM, LM, BitLocker, OSX, MD5 salted or iterated, and the list goes on).

```

hashcat (v6.2.1) starting...

CUDA API (CUDA 11.3)
=====
* Device #1: NVIDIA GeForce RTX 2080 Ti, 10137/11264 MB, 68MCU

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Optimizers applied:
* Optimized-Kernel
* Zero-Byte
* Precompute-Init
* Early-Skip
* Not-Iterated
* Prepend-Salt
* Single-Hash
* Single-Salt
* Brute-Force
* Raw-Hash

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1100 MB

e983672a03adcc9767b24584338eb378:00:hashcat

Session.....: hashcat
Status.....: Cracked
Hash.Name.....: SolarWinds Serv-U
Hash.Target.....: e983672a03adcc9767b24584338eb378:00
Time.Started.....: Sun May 23 11:43:13 2021 (1 sec)
Time.Estimated...: Sun May 23 11:43:14 2021 (0 secs)
Guess.Mask.....: ?a?a?a?a?a?at [7]
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 24620.9 MH/s (32.19ms) @ Accel:32 Loops:1024 Thr:1024 Vec:1
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 31606272000/735091890625 (4.30%)
Rejected.....: 0/31606272000 (0.00%)
Restore.Point....: 0/857375 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:35840-36864 Iteration:0-1024
Candidates.#1...: 4{,erat -> cyr ~}t
Hardware.Mon.#1..: Temp: 62c Fan: 31% Util:100% Core:1920MHz Mem:7000MHz Bus:16

Started: Sun May 23 11:43:12 2021
Stopped: Sun May 23 11:43:15 2021


```


- Tools for cracking password files (GUI):

- Cain & Abel - Windows software; Cracks hash passwords (LM, NTLM), sniff network packets for password, sniff out for local stored passwords, etc.
- Lophcrack - Paid software; Extract and crack hashes; Uses brute force or dictionary attack;
- 0phcrack - Free open-source; Cracks Windows log-in passwords by using LM hashes through rainbow tables.
- Rainbowcrack - Rainbow tables generator for password cracking

- **Legion** - Legion automates the password guessing in NetBIOS sessions. Legion scans multiple IP address ranges for Windows shares and also offers a manual dictionary attack tool.
- **KerbCrack** - Crack Kerberos passwords.
- **Mimikatz** - Steal credentials and escalate privileges (Windows NTLM hashes and Kerberos tickets(Golden Ticket Attack); 'Pass-the-hash' and 'Pass-the-ticker').
- **fgdump** - Dump SAM databases on Windows machines.
- **Pwdump7** - Dump SAM databases on Windows machines.
- **CHNTPW** - chntpw is a software utility for **resetting** or **blanking local passwords used by Windows NT, 2000, XP, Vista, 7, 8, 8.1 and 10**. It does this by editing the SAM database where Windows stores password hashes.
  - Physical access** to victim's computer
  - Startup on BIOS and allow boot to CD or USB
  - Modify the SAM user account information through the CHNTPW

 **rtgen**, **winrtgen** - Tools for generate your own rainbow tables.

 **SAM (Security Account Manager)** is a database file **present in Windows machines that stores user accounts and security descriptors for users on a local computer**. It stores users passwords in a hashed format (in LM hash and NTLM hash). Because a hash function is one-way, this provides some measure of security for the storage of the passwords.

 **/etc/shadow** is where **hashed password data** is stored in **Linux systems** (only users with high privileges can access).

 **Password attack countermeasures:**

- **Length of passwords** is good against **brute-force attacks**.
- **Password complexity** is good against **dictionary attacks**.

## Authentication

- **Three Different Types**
  - **Something You Are** - Uses biometrics to validate identity (retina, fingerprint, etc.)
    - Downside is there can be lots of false negatives
    - **False acceptance rate (FAR) - Type II** - Likelihood that an unauthorized user will be accepted (This would be bad)
    - **False injection rate (FRR) - Type I** - Likelihood that an authorized user will be rejected

- **Crossover error rate (CER)** - Combination of the two; the lower the CER, the better the system
- **Active** - requires interaction (retina scan or fingerprint scanner)
- **Passive** - Requires no interaction (iris scan)
- **Something You Have** - Usually consists of a token of some kind (swipe badge, ATM card, etc.)
  - This type usually requires something alongside it (such as a PIN for an ATM card)
  - Some tokens are single-factor (such as a plug-and-play authentication)
- **Something You Know** - Better known as a password
  - Most systems use this because it is universal and well-known
- **Two-Factor** - When you have two types of authentication such as something you know (password) and something you have (access card)
- **Strength of passwords** - Determined by length and complexity
  - ECC says that both should be combined for the best outcome
  - Complexity is defined by number of character sets used (lower case, upper case, numbers, symbols, etc.)
- **Default passwords** - always should be changed and never left what they came with. Databases such as cirt.net, default-password.info and open-sez.me all have databases of these

## Windows Security Architecture

---

- Authentication credentials stored in SAM file
- File is located at `C:\windows\system32\config`
- Older systems use LM hashing. Current uses NTLM v2 (MD5)
- Windows network authentication uses Kerberos

### LM Hashing

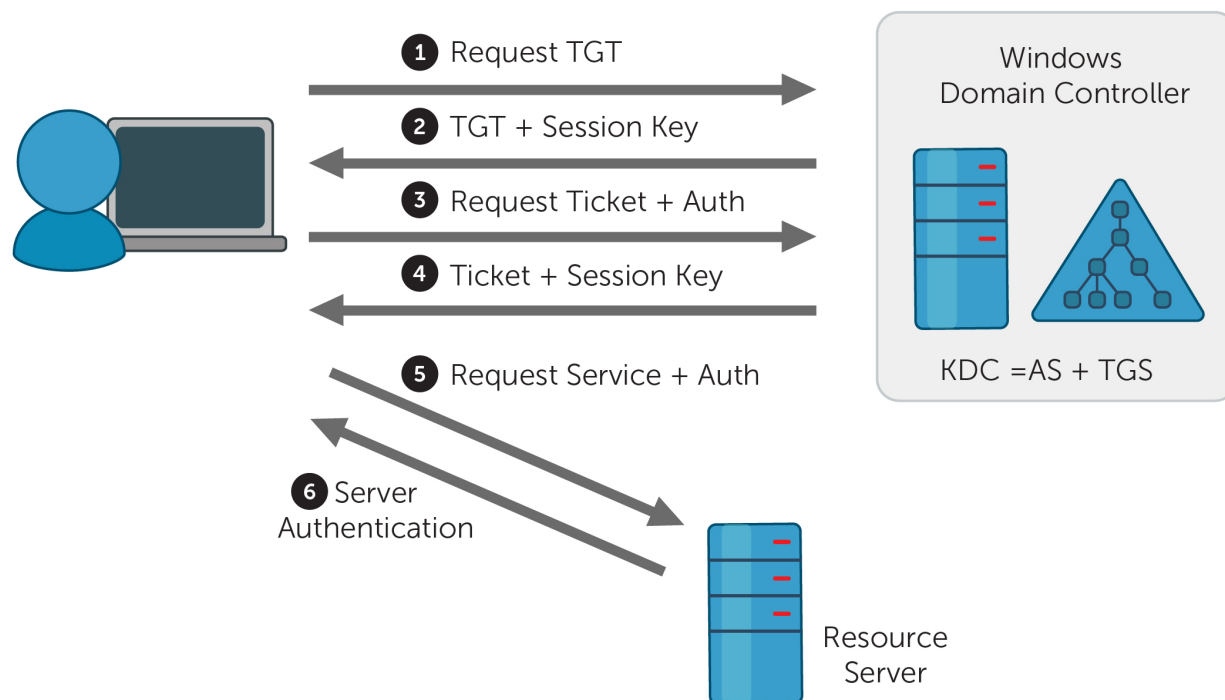
- Splits the password up. If it's over 7 characters, it is encoded in two sections.
- If one section is blank, the hash will be `AAD3B435B51404EE`
- Easy to break if password is 7 characters or under because you can split the hash
- SAM file presents as `UserName:SID:LM_Hash:NTLM_Hash:::`

### Ntds.dit

Database file on a domain controller that stores passwords

- Located in %SystemRoot%\NTDS\Ntds.dit or
- Located in %SystemRoot%\System32\Ntds.dit
- Includes the entire Active Directory

## Kerberos for Active Directory Domain Services (AD DS)



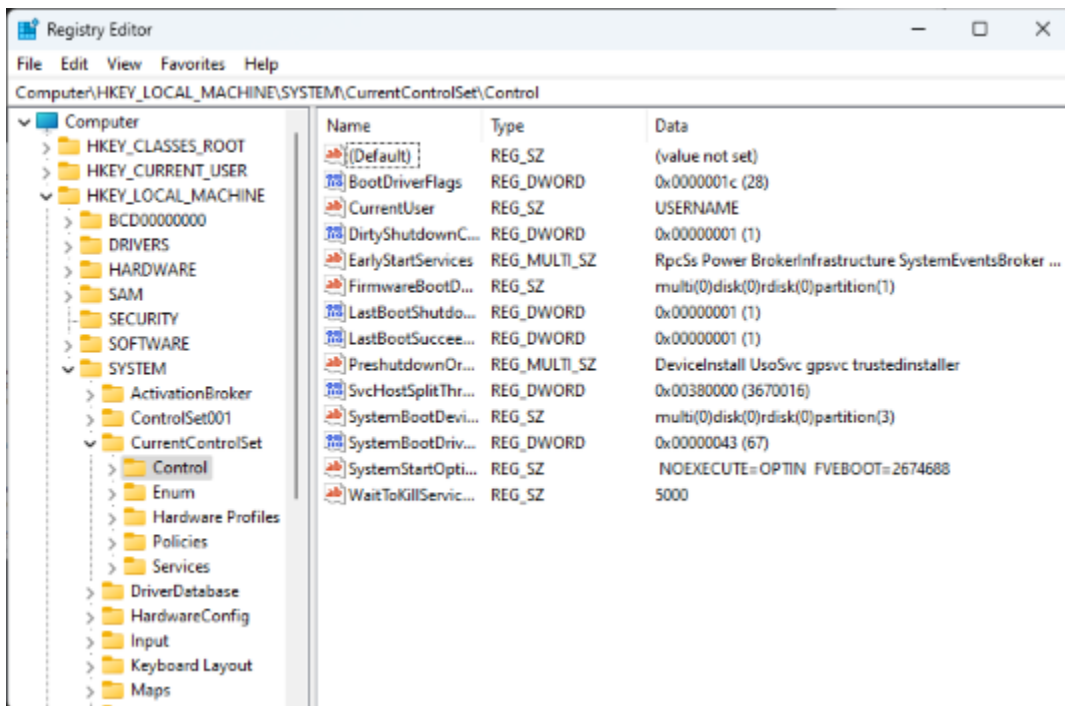
- Steps of exchange
  - i. Client asks **Key Distribution Center (KDC)** for a ticket. Sent in clear text.
  - ii. Server responds with **Ticket Granting Ticket (TGT)**. This is a secret key which is hashed by the password copy stored on the server.
  - iii. If client can decrypt it, the TGT is sent back to the server requesting a **Ticket Granting Service (TGS)** service ticket.
  - iv. Server sends TGS service ticket which client uses to access resources.
- Tools
  - KerbSniff
  - KerbCrack
  - Both take a long time to crack



Uses TCP/UDP Port 88

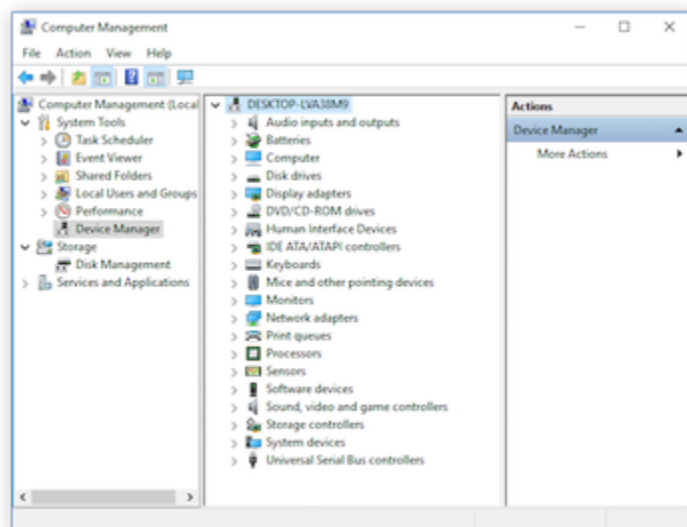
## Registry





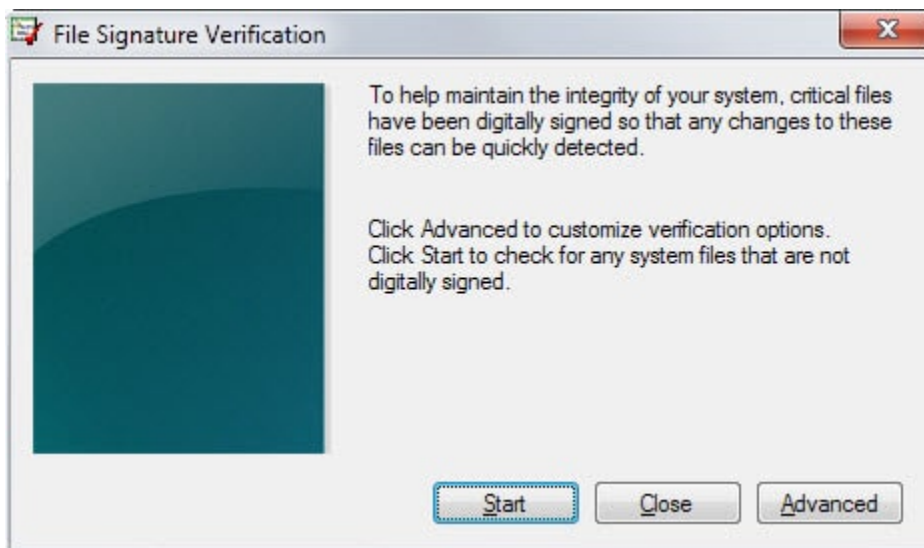
- Collection of all settings and configurations that make the system run
- Made up of keys and values
- Root level keys
  - HKEY\_LOCAL\_MACHINE (HKLM) - information on hardware and software
  - HKEY\_CLASSES\_ROOT (HKCR) - information on file associates and OLE classes
  - HKEY\_CURRENT\_USER (HKCU) - profile information for the current user including preferences
  - HKEY\_USERS (HKU) - specific user configuration information for all currently active users
  - HKEY\_CURRENT\_CONFIG (HKCC) - pointer to HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Hardware Profiles\Current
- Type of values
  - REG\_SZ - character string
  - REG\_EXPAND\_SZ - expandable string value
  - REG\_BINARY - a binary value
  - REG\_DWORD - 32-bit unsigned integer
  - REG\_LINK - symbolic link to another key
- Important Locations
  - HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
  - HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices
  - HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
  - HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
- Executables to edit
  - regedit.exe
  - regedt32.exe (preferred by Microsoft)

# MMC



- Microsoft Management Console - used by Windows to administer system
- Has "snap-ins" that allow you to modify sets (such as Group Policy Editor)

## Sigverif.exe

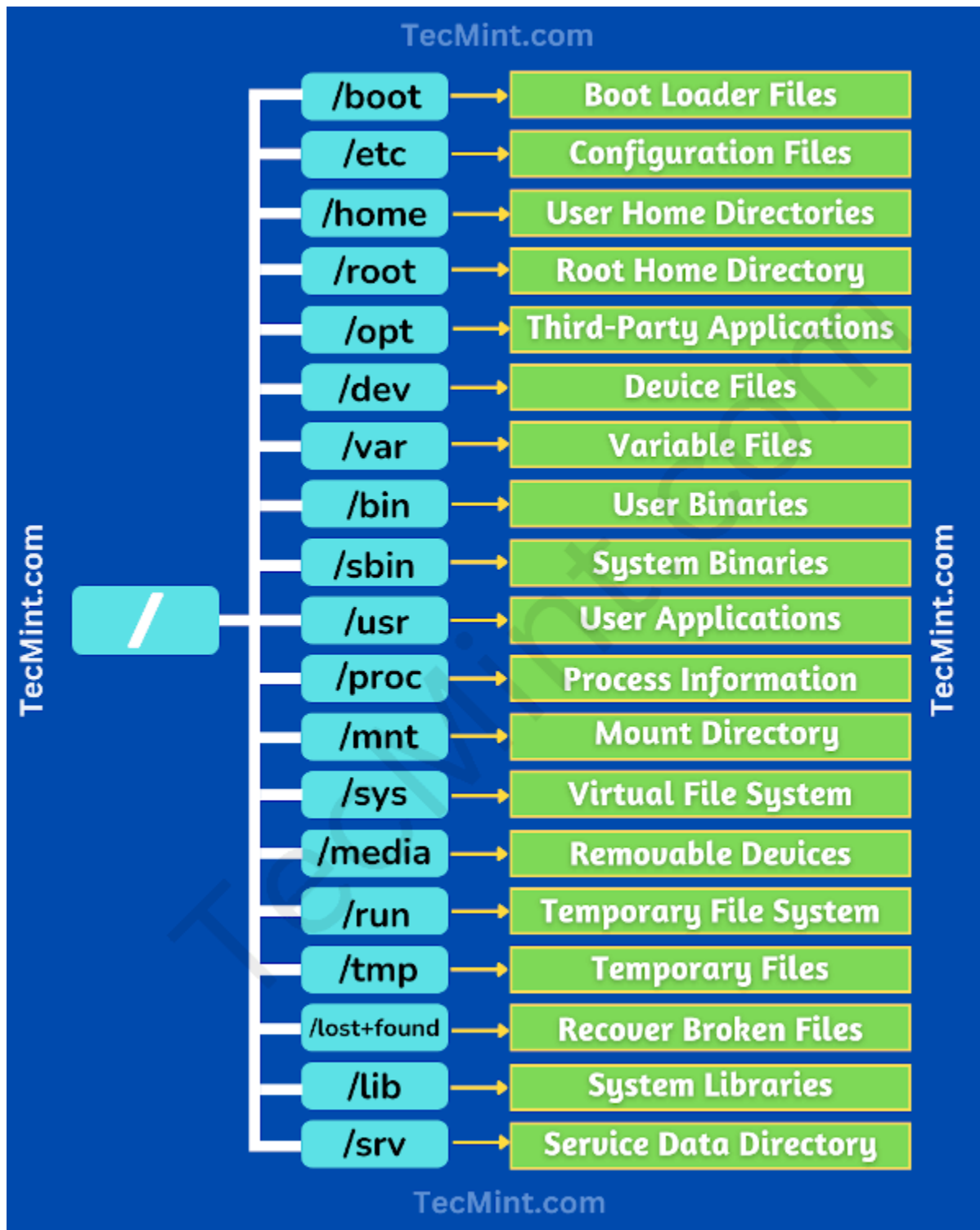


- 
- File Signature Verification (Sigverif.exe) detects signed files and allows you to:
  - View the certificates of signed files to verify that the file has not been tampered with after being certified.
  - Search for signed files.
  - Search for unsigned files.

# Linux Security Architecture

## Linux Directory Structure

- Linux root is just a slash (/)
  - Important locations
    - / - root directory
    - **/bin** - basic Linux commands
    - **/dev** - contains pointer locations to various storage and input/output systems
    - **/etc** - all administration files and passwords. Both password and shadow files are here
    - **/home** - holds the user home directories
    - **/mnt** - holds the access locations you've mounted
    - **/sbin** - system binaries folder which holds more administrative commands
    - **/usr** - holds almost all of the information, commands and files unique to the users
-



## Linux Common Commands

Command	Description
<code>adduser</code>	Adds a user to the system
<code>cat</code>	Displays contents of file
<code>cp</code>	Copies
<code>ifconfig</code>	Displays network configuration information

Command	Description
<code>kill</code>	Kills a running process
<code>ls</code>	Displays the contents of a folder. <code>-l</code> option provides most information.
<code>man</code>	Displays the manual page for a command
<code>passwd</code>	Used to change password
<code>ps</code>	Process status. <code>-ef</code> option shows all processes
<code>rm</code>	Removes files. <code>-r</code> option recursively removes all directories and subdirectories
<code>su</code>	Allows you to perform functions as another user (super user)

- Adding an ampersand after a process name indicates it should run in the background.

- `pwd` - displays current directory

- `chmod` - changes the permissions of a folder or file

- Read is 4, write is 2 and execute is 1

- | Read | Write | Execute |
|------|-------|---------|
| r--  | -w-   | --x     |
| 4    | 2     | 1       |

- First number is user, second is group, third is others

- when you issue the `ls` command with `-la` flag on Linux, you can see the permissions. As you can see below the file have a permission for everyone (777), will be like this:

- `rw-rw-rwx` ---> user
    - `rw-rw-rwx` ---> group
    - `rw-rw-rwx` ---> others

- Another example - 755 is **everything** for users, read/execute for group, and read/execute for others

- `rw-r-xr-x` ---> user
    - `rw-r-xr-x` ---> group
    - `rw-r-xr-x` ---> others

- You also can set permissions like: `chmod g=rw` (set read/write for groups).

- Root has UID and GID of 0 - you can see this information by issuing the command `id`.

```
root@kali:~# id
```

- `uid=0(root) gid=0(root) groups=0(root)`

- First user has UID and GID of 500 (Fedora and CentOS); in most Linux systems the **non-root/normal user** are **UID and GID of 1000**.

- `normal-user@kali:~# id`

- `id`  
`uid=1000(kali) gid=1000(kali)`  
`groups=1000(kali),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev`

- Passwords are stored in **/etc/shadow** for most current systems

- **/etc/passwd** stores passwords in hashes.

- `cat /etc/passwd`

- `root:x:0:0:root:/root:/bin/bash`  
`daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin`  
`bin:x:2:2:bin:/bin:/usr/sbin/nologin`  
`sys:x:3:3:sys:/dev:/usr/sbin/nologin`  
`sync:x:4:65534:sync:/bin:/bin/sync`  
`(...)`

- **/etc/shadow** stores passwords encrypted (hashed and salted) and is only accessible by root

- `sudo cat /etc/shadow`

- `root:!:18390:0:99999:7:::`  
`daemon*:18390:0:99999:7:::`  
`bin*:18390:0:99999:7:::`  
`kali:$6$a/53BntOdPOaghAx$VCAdR3Af97cYTtWCtDp9iksacL3gj2Sgrb12EMix0ITuxc5j0Qp1lbaRi.jNl`  
`(...)`

## Privilege Escalation and Executing Applications

⚡ Check out the [practical lab on PrivEsc](#)

**Vertical** - Lower-level user executes code at a higher privilege level (*e.g.: common user to root/administrator*).

**Horizontal** - executing code at the same user level but from a location that would be protected from that access

- Crack the password of an admin - primary aim
- Taking advantage of an OS vulnerability
  - One way to perform a priv esc is using CVE's in order to perform local shells, c shells, web shells and so on.
  - Examples:
    - Linux: [DirtyCow](#) race-condition vulnerability;
    - Windows: [EternalBlue](#) exploits the old Samba version 1 to leverage a Remote code execution (RCE);
- **DLL Hijacking** - replacing a DLL in the application directory with your own version which gives you the access you need
- In Linux machines is possible to look for **crontabs** and find misconfigurations on privileges.
- In Linux, **insecure sudo** can lead a privilege escalation to root; You can check this by typing: `sudo -l`. If there's any system command that allows **NOPASSWD option** this may lead to escalation.
- Nmap old versions you can start **interactive mode** and issue the `!/bin/bash` to elevate root privileges.
- Use a tool that will provide you the access such as Metasploit
- Social engineering a user to run an application
- ECC refers executing applications as "owning" a system
- **Executing applications** - starting things such as keyloggers, spyware, back doors and crackers

## Covert data gathering

---

**Keyloggers** - record keys strokes of a individual computer keyboard or a network of computers.

- Keylogger when associated with spyware, helps to transmit your information to an unknown third party.

- **Types of Keyloggers:**
- **Hardware keylogger**
  - PC/BIOS embedded
  - Keyboard
  - External device
    - PS/2 and USB
    - Acoustic/CAM
    - Bluetooth
    - Wi-Fi
  - **Hardware Keylogger Tools:**
    - KeyGrabber - electronic device capable of capturing keystrokes from PS/2 USB keyboard.
- **Software keylogger**
  - Application
  - Kernel
  - Hypervisor-based
  - Form Grabbing based (records from web form data)
  - **Software Keylogger Tools:**
    - KeyCarbon
    - Keyllama Keylogger
    - Keyboard logger
    - KeyGhost

**Spywares - watching user's action and logging them without the user's knowledge.**

- Hide its process, files and other objects
- Spywares can steal user's PII, monitor activity, display annoying pop-ups, redirect web pages to ads, change the browser's settings, steal passwords, modify the DLLs, change firewall settings and so on.
- **Types of spyware:**
  - Desktop
  - Email
  - Internet
  - Child-Monitoring
  - Screen Capturing



- USB
- Audio and Video
- Printers
- Mobile devices / Telephones / Cellphones
- GPS
- **Spyware Tools:**
  - [SpyAgent](#) - allows you to secretly monitor and record all activities on your computer, which is completely legal.
  - [Power Spy](#) - allows you to secretly monitor and record all activities on your computer, which is completely legal.
  - **mSpy** - GPS spyware that trace the location of particular mobile devices.
  - **USBDeview** - monitors and analyzes data transferred between any USB device connected to a computer.

## Defending against Keyloggers and Spywares

- Restrict physical access to computer systems
- Use anti-keylogger between the keyboard and its driver
- Use pop-up blocker and avoid opening junk emails
- Use anti-spyware/antivirus
- Firewall and anti-keylogging software(Zemana AntiLogger)
- Update and patch!
- Recognize phishing emails
- Host-based IDS
- Automatic form-filling password manager or virtual keyboard

## Hiding Files

⚡ Check out the practical labs(2) on [Hiding Files using NTFS streams](#) and [Steganography](#)

- In Windows, you can use **Alternate Data Stream** (ADS) to hide files:
  - Hides a file from directory listing on an NTFS file system
    - `type badfile.exe: > plaintext.txt:badfile.exe`
    - Next create a symlink `mklink normalApp.exe readme.txt:badfile.exe`
  - You can also clear out all ADS by copying files to a FAT partition
  - To show ADS, `dir /r` does the trick;
    - You can use `streams` from **Sysinternals** to show streams.
    - Also you can use **FTK (Forensics ToolKit)** to look for this
- You can also hide files by attributes

- In Windows: `attrib +h filename`
- In Linux, simply add a `.` to the beginning of the filename ( `.file.tar` )
- **Can hide data and files with steganography**
  - Tools for steganography:
    - CLI (Linux):
      - `steghide`
    - GUI (Windows):
      - Snow
      - OpenStego
      - OpenPuff



## Steganography:

- **Steganography** - practice of concealing a message inside another medium so that only the sender and recipient know of its existence
- **Ways to Identify**
  - Text - character positions are key - blank spaces, text patterns
  - Image - file larger in size; some may have color palette faults
  - Audio & Video - require statistical analysis
- **Methods**
  - Least significant bit insertion - changes least meaningful bit
  - Masking and filtering (grayscale images) - like watermarking
  - Algorithmic transformation - hides in mathematical functions used in image compression
- **Tools**
  - QuickStego
  - gifshuffle
  - SNOW
  - Steganography Studio
  - OpenStego

## Rootkits

---

- Software put in place by attacker to obscure system compromise
- Hides processes and files
- Also allows for future access
- **Examples**
  - Horsepill - Linux kernel rootkit inside initrd
  - Grayfish - Windows rootkit that injects in boot record
  - Firefex - multi-component family of malware

- Azazel
- Avatar
- Necurs
- ZeroAccess
- **Hypervisor level** - rootkits that modify the boot sequence of a host system to load a VM as the host OS
- **Hardware** - hide malware in devices or firmware
- **Boot loader level** - replace boot loader with one controlled by hacker
- **Application level** - directed to replace valid application files with Trojans
- **Kernel level** - attack boot sectors and kernel level replacing kernel code with back-door code; most dangerous
- **Library level** - use system-level calls to hide themselves
- One way to detect rootkits is to map all the files on a system and then boot a system from a clean CD version and compare the two file systems

## Covering Tracks

---

Clearing logs is the main idea behind covering tracks.

1. Find and clear the logs.
2. Falsify/Modify logs.

### On Linux:

- Linux keep the **command line history** on `.bash_history` file
  - To clear out the command line history use `rm -rf` to force remove. You also can use `shred -zu` that deletes the file and **overwrite on memory**.
  - You can also use `history -c` to clear all command line history on entire system or `history -w` to clear out all session history.
- **Turn off the command logs:**
  - `export HISTSIZE=0`
  - `echo $HISTSIZE` will return 0 limiting the number of commands which can be saved in `$HISTFILE`.
- **clearev** - Meterpreter shell command to clear log files (issued inside Metasploit Framework)

### Most common logs on Linux:

- `/var/log/messages` or `/var/log/syslog/`

- General messages, as well as system-related information.
- `/var/log/auth.log` OR `/var/log/secure`
  - Store authentication logs, including both successful and failed logins and authentication methods.
- `/var/log/boot.log`
  - Related to booting and any messages logged during startup.
- `/var/log/maillog` OR `var/log/mail.log`
  - stores all logs related to mail servers.
- **Clearing and Modifying logs on Linux:**
  - It is possible to echo whitespace to clear the event log file:
    - `echo " " > /var/log/auth.log`
  - Also you can perform this by using 'black hole dev/null':
    - `echo /dev/null > auth.log`
  - To tamper/modify the log files, you can use `sed` stream editor to delete, replace and insert data.
    - `sed -i '/opened/d' /var/log/auth.log` - this command will delete every line that contains the '**opened**' word, that refers to opened sessions on Linux system.

## On Windows:

- To clear out all **command line history**:
  - On **Cmd Prompt**: press [ alt ] + [ F7 ]
  - On **PowerShell**: type `Clear-History`

In Windows, you need to clear **application**, **system** and **security logs**.

- **Auditpol** for changing settings on log files (used for manipulate audit policies).
- Main commands:
  - `auditpol /get /category:*` --> display all audit policies in detail if is enable (*Object Acces, System, Logon/Logoff, Privilege Use, and so on*).
  - `auditpol /clear` --> reset (disable) the system audit policy for all subcategories.
  - `auditpol /remove` --> Removes all per-user audit policy settings and disables all system audit policy settings.

⚡ Check out the [practical lab on Auditpol](#)

- **MRU** (Most Recently Used) programs that registry recently used programs/files and saves on Windows Registry.
- Is possible to manually clear the logs on Event Viewer.

## Conclusion on Covering Tracks

- Option is to corrupt a log file - this happens all the time
- Best option is be selective and delete the entries pertaining to your actions.
- **Can also disable auditing ahead of time to prevent logs from being captured**
- Tools:
  - ccleaner --> automate the system cleaning, scrub online history, log files, etc. [Windows]
  - MRUblaster [Windows]
  - Meterpreter on MSF have **clearev** to clear all event logs remotely. [Kali Linux using MSF]