



**Prifysgol Cymru**

Y Drindod Dewi Sant

**University of Wales**

Trinity Saint David

*Trawsnewid Addysg, Trawsnewid Bywydau*  
*Transforming Education, Transforming Lives*

# Guest Lecture

Kamadchisundaram Sureshkumar (Suresh)  
Lecturer in Computing and Training Manager  
(Cybersecurity) – WorldSkills UK

# Hacking Ethically!!!



## *Advanced Cybersecurity - Continuation* (ACCB6000)

Applied Computing - WISA

# VM Download

\\192.168.135.200\ACCB6000\$

## Previous sessions...

- **Understanding Information Security, Risks involved, assessing, managing and monitoring Risks, Cyber risk resilience**
- **Purpose and role of Hackers, Penetration Testing, Various Laws and Standards**
- **Understanding Footprinting and Reconnaissance Concepts, Methodology.**
- **Using various Tools, Taking Countermeasures**
- **Network Scanning, Tools and Techniques**
- **Banner Grabbing**
- **Scanning – PenTesting**
- **Enumeration Concepts, Techniques, Countermeasures**

So far, we discussed about Information security, the threats and attack vectors. In continuation to that, I hope *you all have understood the hacking concepts and the reasons to stay ethical*. Add more insight, we also *discussed various security laws and standards*. To conclude, we covered various “Risk” concepts like risk factors, cyber risk resilience in the process of enhancing security.

In continuation, we discussed *Footprinting concepts focusing on different methodologies like Search Engines, Google Hacking Techniques etc.* giving a walkthrough on Website Footprinting, Email Footprinting, Competitive Intelligence, Network Footprinting etc.



## Previous sessions...

### **Understanding and Assessing Vulnerabilities**

- **What is Vulnerability? How is it classified?**
- **Assessing Vulnerabilities.**
- **Managing Vulnerabilities**

### **Social Engineering**

- **Understand Social Engineering Concepts, Techniques**
- **Understanding Threats, Impersonation**
- **Countermeasures**

### **Lab/ Tutorials**

- **Activity 1 Getting Ready**
- **Activity 2 Information Gathering**
- **Activity 3 Scanning and Enumeration**
- **Activity 4- Vulnerability Assessment**
- **Activity 5 – Social Engineering**

Continuing the earlier lectures, we discussed on Network Scanning, different tools and methodologies used like Banner Grabbing and mitigation methods etc. following that enumeration concepts, its different tools and techniques were discussed. we continued focusing on assessing vulnerabilities within a network (either it is our target to attack or defend), different tools used to assess and techniques

**During the tutorials, under isolated lab environment you might have practiced different Footprinting, scanning, enumerating methods using various tools and assessed the vulnerabilities.**

## ***Gaining access and escalating privileges***

- ***Hacking Methodology (Based on CEH)***
- ***Understanding techniques to gain access to the system***
- ***Techniques to escalate privileges***

## ***Lab/ Tutorials***

- ***Activity 6 – Exploiting and Testing***

*During this week, with applied knowledge and understanding on **Footprinting, Network Scanning, Enumerating the target, and assessing vulnerabilities**, we shall move forward focusing on the hacking methodology. We shall understand various techniques to gain access, escalate privileges, create and maintain remote access to the target.*

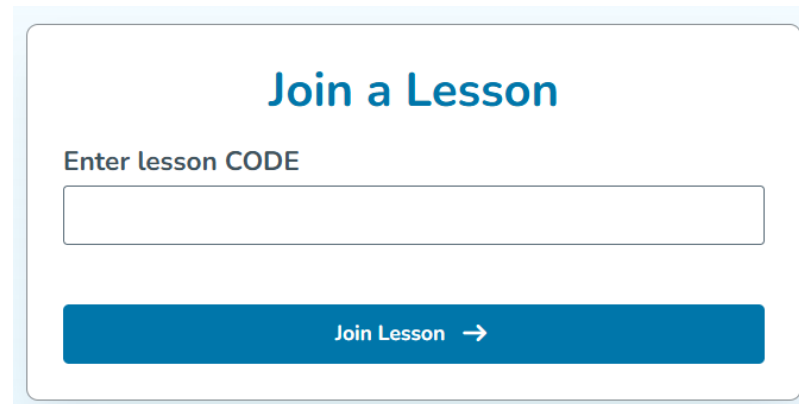
***During the tutorials, under isolated lab environment you will be using **different techniques to gain access and escalate privileges**.***

# Recap on last week's lesson 6

Please use your computer or mobile device to browse the following link:

<https://nearpod.com/student/>

You will see:



Join a Lesson

Enter lesson CODE

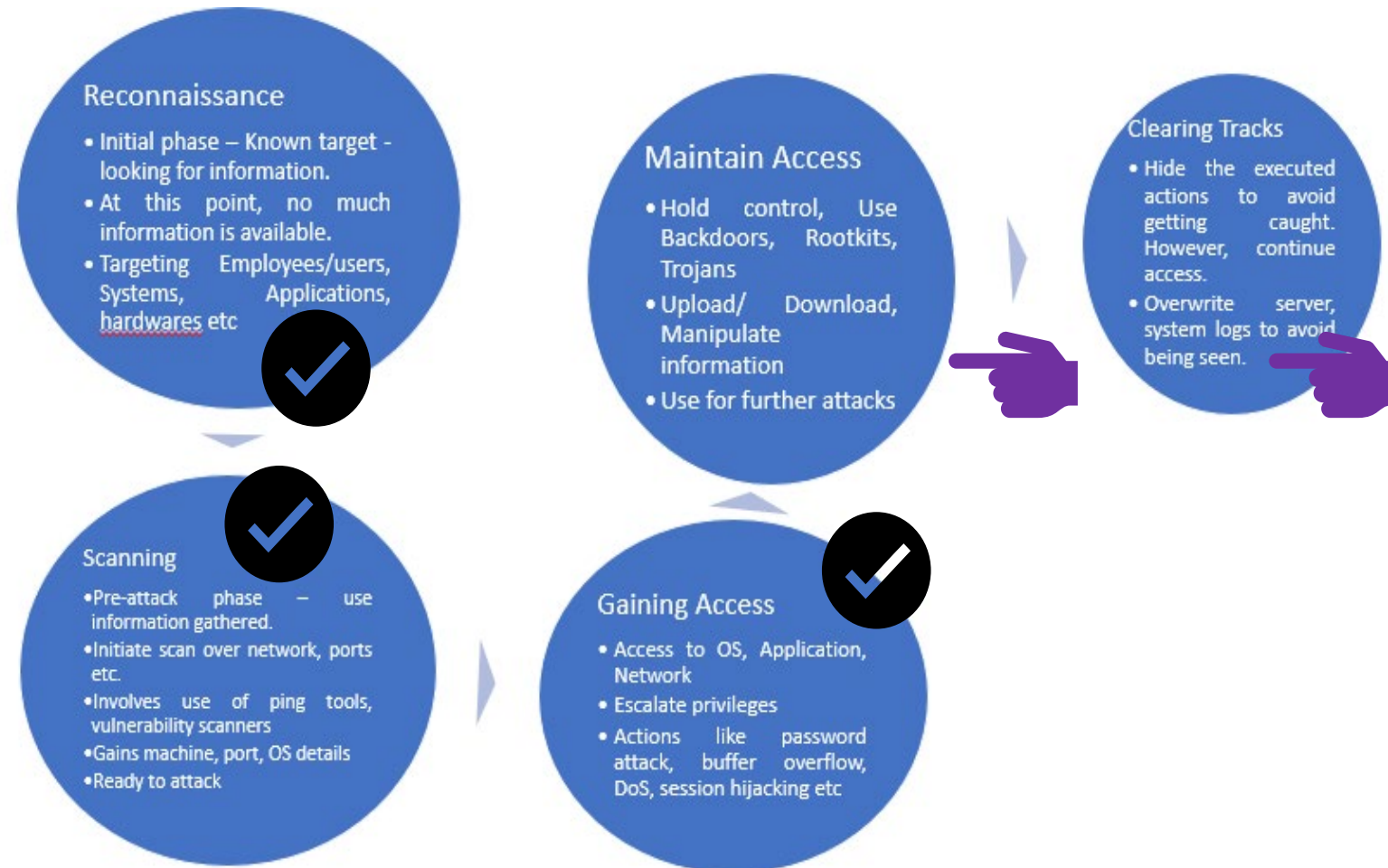
Join Lesson →

\*The lesson code will be given.

# Hacking - Methodology

## ○ Do you remember the process?

- ✓ Footprinting and Reconnaissance
- ✓ Scanning
- ✓ Enumeration
- ✓ Vulnerability Analysis
- Gaining Access
- Maintaining Access
- Clearing Logs



# Hacking - Methodology

- Do you remember the process?
  - ✓ Footprinting and Reconnaissance
  - ✓ Scanning
  - ✓ Enumeration
  - ✓ Vulnerability Analysis
  - Gaining Access
  - Maintaining Access
  - Clearing Logs

## Gaining Access

Cracking Passwords ∞ Escalate Privileges

## Maintaining Access

Executing Applications ∞ Hiding Files

## Clearing Logs

Cover your tracks



# Hacking - Methodology

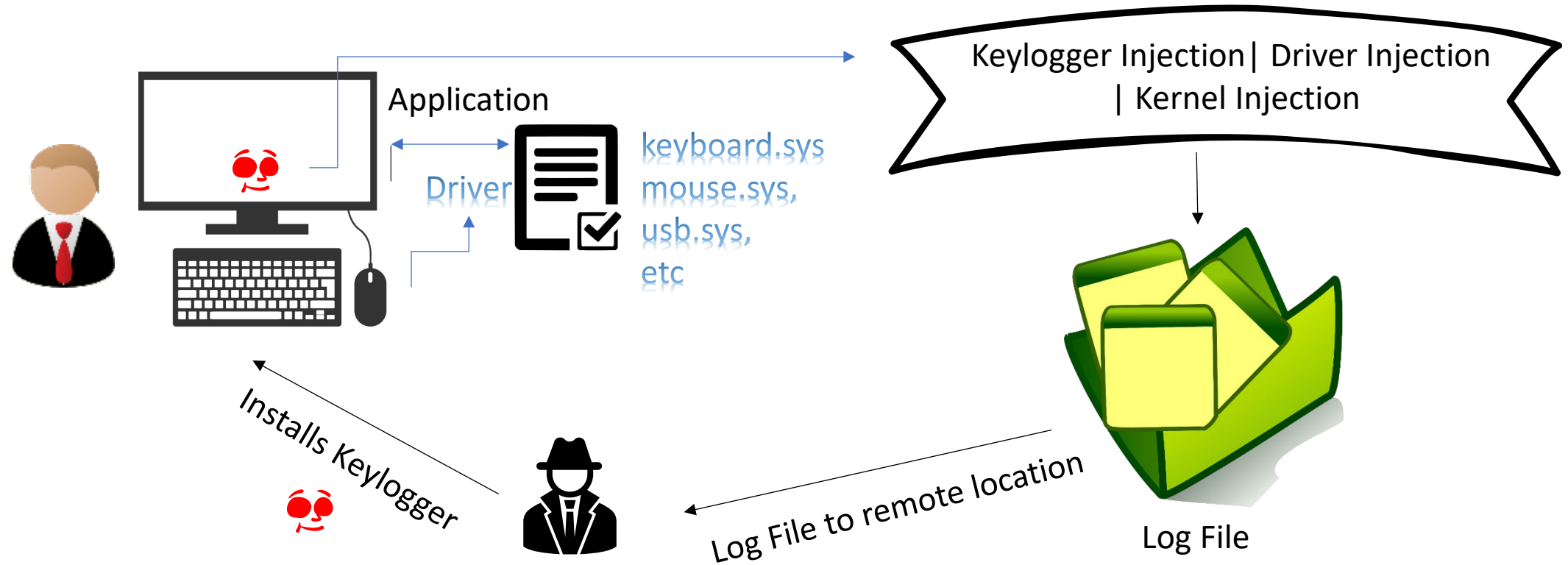
- Gaining Access
  - To bypass the access control systems and gain access into the target system
  - Techniques like Password cracking, Social Engineering helps to gain access
- Privilege Escalation
  - To get rights of a legitimate user or administrator
  - Techniques like vulnerability exploitation helps to escalate privileges
- Execute Application
  - To create and maintain remote access to the target system
  - Trojans, Spywares, Backdoors, Keyloggers help to execute applications
- Hiding Files
  - To hide your presence in the target system
  - To hide the attackers' malicious activities and data theft
  - Rootkits, Steganography helps to hide files
- Covering Tracks
  - To hide the evidences
  - Clearing the logs

# Executing Applications

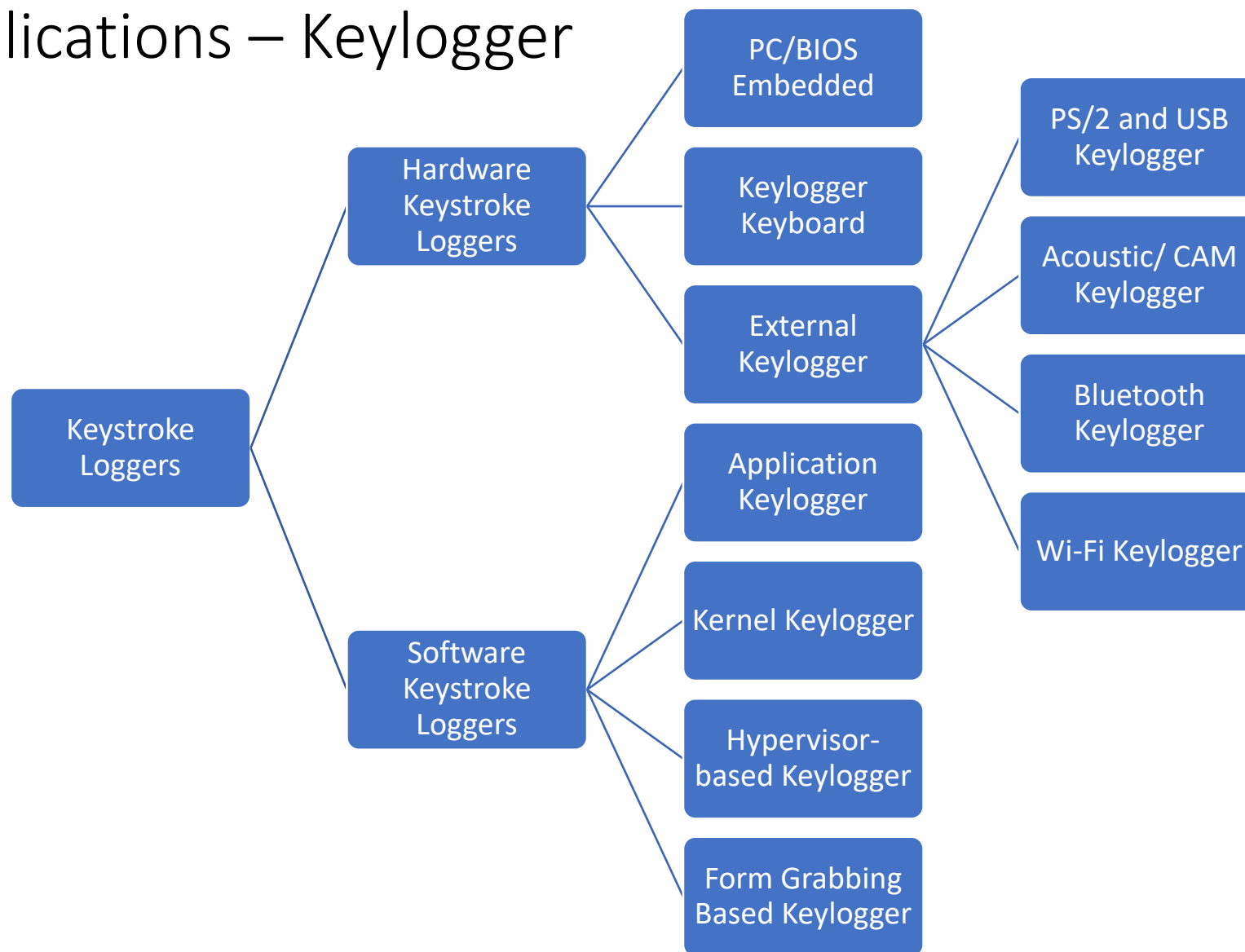
- Once privileges are escalated, attackers are now able to execute malicious applications.
- The **malicious programs** are **remotely executed** by the attackers. This enables the attacker to breakdown the privacy by looking at the information, screen captures etc and to install backdoors to maintain easy access.
- Types
  - Keyloggers
  - Backdoors
  - Spyware
  - Crackers

# Executing Applications – Keylogger

- Keylogger (Keystroke Logger is a software/ hardware that observes the key stroke).
- The purpose of Keylogger is to **monitor computer activities** based on organizational policies/ children's activities etc.
- However, it **allows attackers to spy on the target**.



# Executing Applications – Keylogger



# Hardware Keystroke Loggers

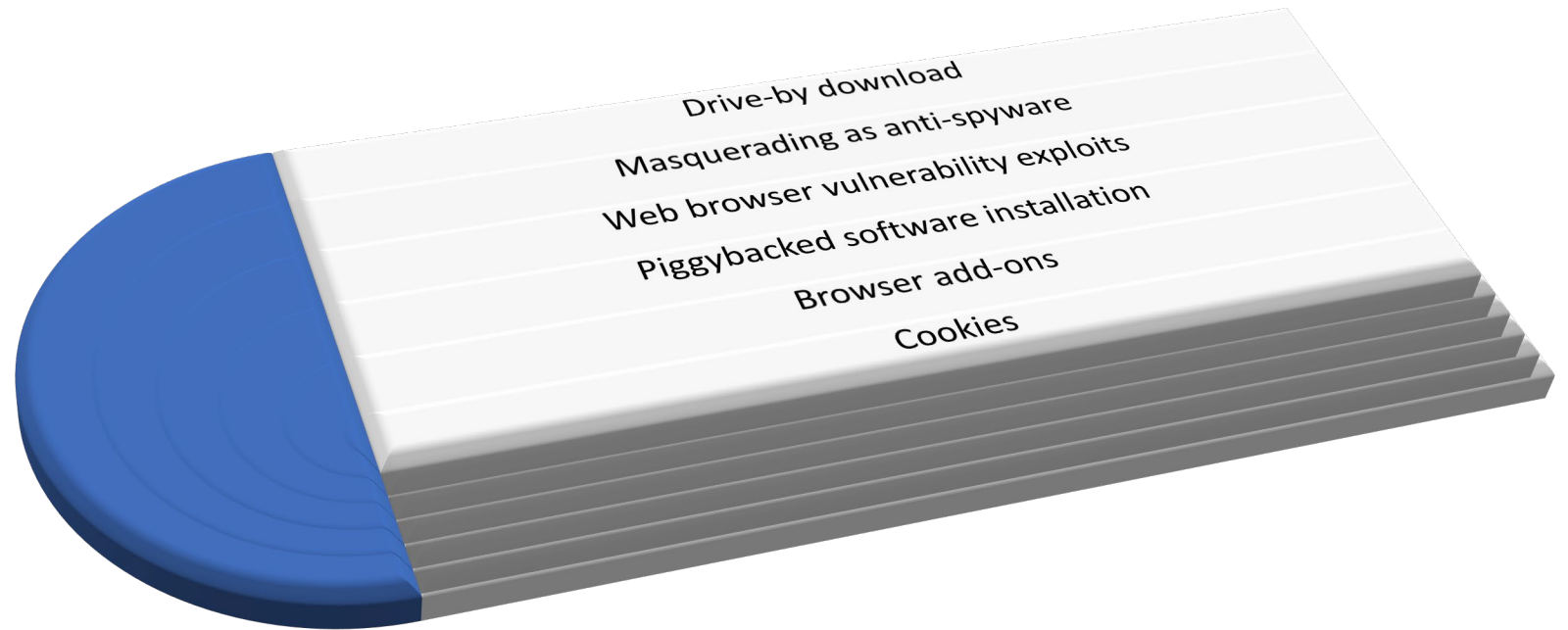
| Type                                | How it works                                                       | Detection Difficulty |
|-------------------------------------|--------------------------------------------------------------------|----------------------|
| <b>PC/BIOS Embedded Keylogger</b>   | Embedded in BIOS/motherboard capturing keystrokes before OS loads. | ★★★★★ Very Hard      |
| <b>Keylogger Keyboard</b>           | Modified keyboard with internal logging hardware.                  | ★★★ Moderate         |
| <b>External Keylogger (Generic)</b> | Device placed between keyboard and PC (USB/PS/2).                  | ★★★ Moderate         |
| <b>PS/2 or USB Keylogger</b>        | Small inline dongle intercepting keystrokes from wired keyboards.  | ★★ Low               |
| <b>Acoustic/CAM Keylogger</b>       | Uses audio or camera analysis to infer keystrokes.                 | ★★★★ High            |
| <b>Bluetooth Keylogger</b>          | Intercepts wireless Bluetooth keyboard communication.              | ★★★ Medium           |
| <b>Wi-Fi Keylogger</b>              | Logs keystrokes and transmits them via Wi-Fi.                      | ★★★★ High            |

# Software Keystroke Loggers

| Type                              | How it works                                                        | Detection Difficulty |
|-----------------------------------|---------------------------------------------------------------------|----------------------|
| <b>Application Keylogger</b>      | Uses OS hooks, APIs, or clipboard monitoring to capture keystrokes. | ★ ★ Low–Medium       |
| <b>Kernel Keylogger</b>           | Installs kernel-level drivers to intercept keystrokes at OS core.   | ★ ★ ★ ★ ★ Very Hard  |
| <b>Hypervisor-Based Keylogger</b> | Compromised or malicious hypervisor monitors keystrokes under OS.   | ★ ★ ★ ★ ★ Very Hard  |
| <b>Form Grabbing Keylogger</b>    | Captures web form inputs before encryption (pre-HTTPS layer).       | ★ ★ ★ Medium         |

# Executing Applications – Spyware

- It is a **stealthy program** that **hides its process**, files and other objects **from being detected**.
- It records **user interaction** with the computer and internet without the user's knowledge and sends them to the remote attackers.



## Spyware Propagation



# Defending

- **Against Keyloggers**

- ✓ **Anti-spyware/antivirus programs, Host-based IDS, Firewalls, Anti-logging software's** with updated signature
- ✓ **Keystroke interference software**
- ✓ **Use registry editor or process explorer** to check for the keystroke loggers
- ✓ **Restrict Physical access**
- ✓ **Use encryption** between keyboard and its driver
- ✓ **Use anti-keylogger**

# Defending

- **Against Spywares**

- ✓ **Anti-spyware/antivirus programs, Host-based IDS, Firewalls, Anti-logging software's** with updated signature
- ✓ Set **Browser security settings**
- ✓ Check **task manager report** and **MS configuration manager report**
- ✓ Create **Awareness** – Be aware when downloading files, Pop-ups etc

# Hiding Files

Once, gaining access (using various methods – we have discussed so far)

- Attackers may **place files to grant access to the target** and **execute malicious tasks**.
- These files may be hidden from being identified.
- These programs are called as **Rootkits**.
- Rootkits mainly **focus on replacing some OS system calls** and utilities with a modified version.
- Rootkits generally are **Backdoor programs, DDoS programs, Packet Sniffers, BOTS** etc.

## Objectives of Rootkit

- ***Gain*** backdoor ***access***.
- ***Store malicious programs***.
- ***Hide tracks*** and ***malicious applications***.
- ***Gather restricted data***.

# Hiding Files

## Types of Rootkit

- **Hypervisor Level Rootkit.**
- **Hardware/ Firmware Rootkit.**
- **Kernel Level Rootkit.**
- **Boot Loader Level Rootkit.**
- **Application-Level Rootkit.**
- **Library Level Rootkit.**

## ✓ Rectangle Table Discussion

- ✓ **Sony's Rootkit Scandal**
- ✓ **New FudModule Rootkit**
- ✓ **What is MosaicRegressor?**
  - ✓ Collaborate as a Team (2-3 Members) – Not more
  - ✓ Do a detailed research on the above rootkits
  - ✓ Document your Team's collected information and initiate a discussion.
    - ✓ Discuss/ Debate with other teams via our Rectangle Table Discussion.

## Process (Original)

- Code section...
- Call FindNextFile
- Import data section
- FindNextFile: 0x87651234
- Kernel32.dll
- 0x87651234: FindNextFile code

## Process (Hooked)

- Code section...
- Call FindNextFile
- Import data section
- FindNextFile: 0x87651234
- Kernel32.dll
- **0x98752**1234: FindNextFile code
- Rootkit Code: **0x98752**1234: FindNextFile

# Types of Rootkits

| Rootkit Type                     | Where it operates                           | How it works                                                            | Detection Difficulty |
|----------------------------------|---------------------------------------------|-------------------------------------------------------------------------|----------------------|
| <b>Hypervisor-Level Rootkit</b>  | Beneath the OS, in the virtualization layer | Replaces or compromises the hypervisor to intercept OS-level operations | ★★★★★ Very Hard      |
| <b>Hardware/Firmware Rootkit</b> | BIOS, firmware, hardware components         | Injects malicious code into firmware (BIOS, network cards, hard drives) | ★★★★★ Very Hard      |
| <b>Kernel-Level Rootkit</b>      | Operating system kernel                     | Modifies system calls, kernel modules, and core OS processes            | ★★★★★ Very Hard      |
| <b>Boot Loader Level Rootkit</b> | Boot loader / pre-OS environment            | Loads before the OS and manipulates the OS as it boots                  | ★★★★ High            |
| <b>Application-Level Rootkit</b> | User-space applications                     | Replaces or modifies legitimate application files                       | ★★ Moderate          |
| <b>Library-Level Rootkit</b>     | Shared libraries (DLLs, .so files)          | Injects or replaces libraries used by many programs                     | ★★★ Medium–High      |

# Detecting Rootkits

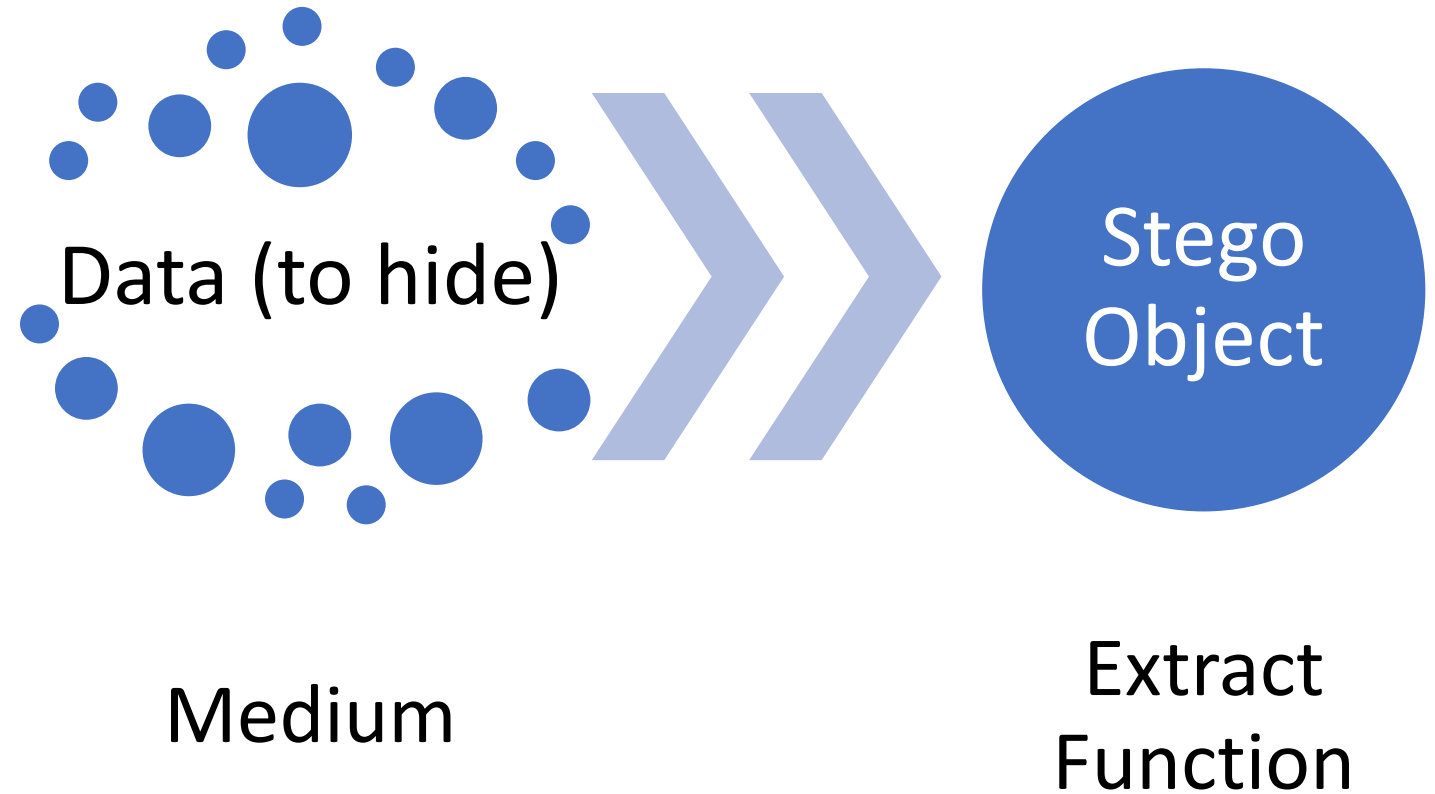
| Detection method                             | How it works                                                                                                                                                         | Strengths                                                                                                              |
|----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| <b>Integrity-Based Detection</b>             | Compares current system files, binaries, or configurations with known-good baselines to detect unauthorized changes.                                                 | Very effective for detecting modified system files or kernel manipulation; ideal for rootkits that alter system calls. |
| <b>Signature-Based Detection</b>             | Uses known malware/rootkit signatures to identify malicious components through pattern matching.                                                                     | Fast and efficient; effective against known threats with established signatures.                                       |
| <b>Heuristic / Behaviour-Based Detection</b> | Monitors system behaviour, unusual activity, or deviations from normal patterns to detect suspicious operations.                                                     | Can detect new or unknown rootkits; useful for catching zero-day threats.                                              |
| <b>Runtime Execution Path Profiling</b>      | Analyses the normal execution path of programs and flags deviations caused by malicious code injection or altered execution flows.                                   | Excellent against rootkits that alter program flow or inject code; detects deep system manipulation.                   |
| <b>Cross-View-Based Detection</b>            | Compares multiple system “views” (e.g., OS API view vs. raw disk view) to identify inconsistencies created by rootkits hiding files, processes, or registry entries. | Very effective for detecting hidden processes, files, and kernel objects that rootkits attempt to conceal.             |

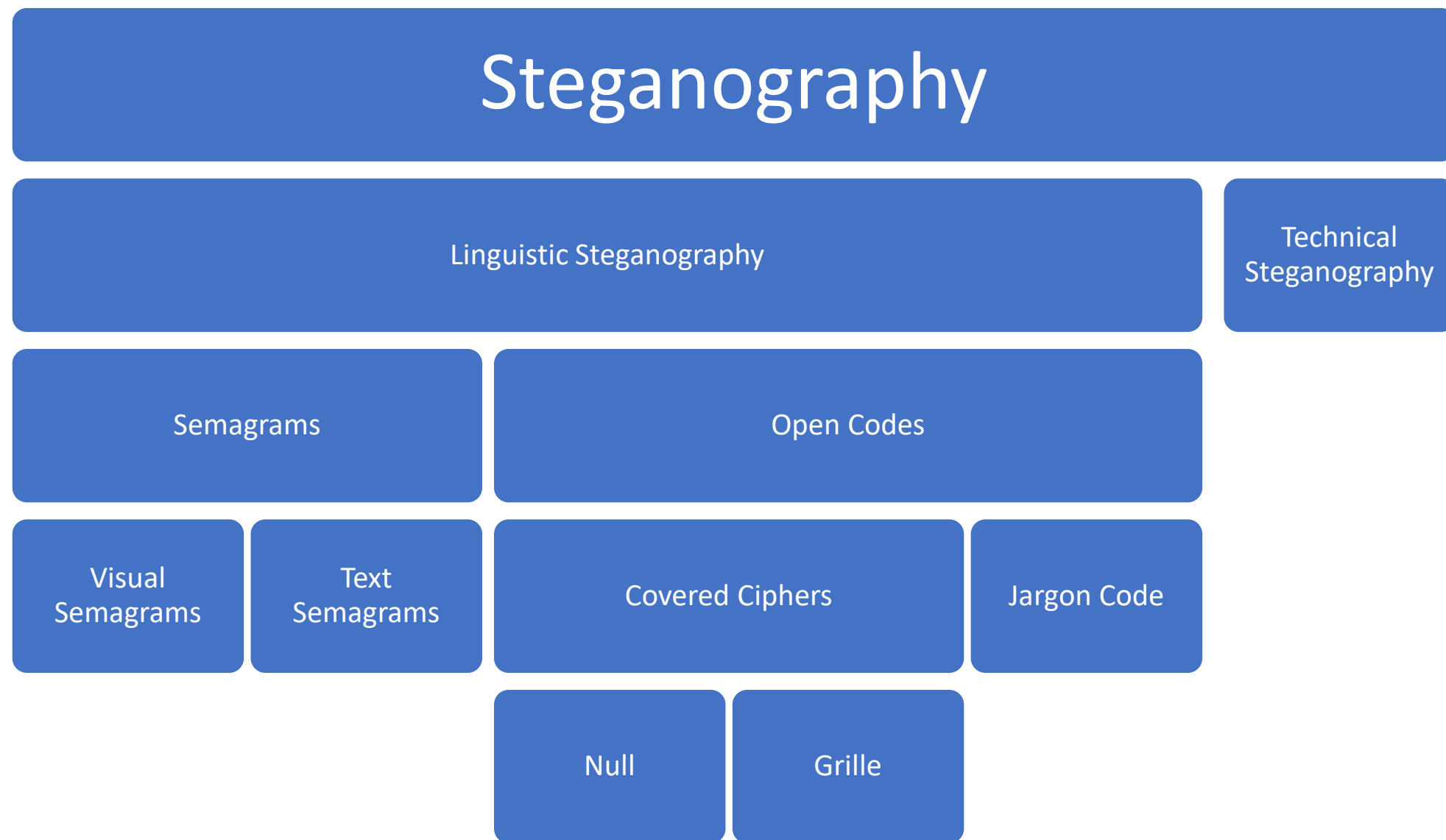
# Defending Rootkits

| Defence measure                                         | What it does                                                                         |
|---------------------------------------------------------|--------------------------------------------------------------------------------------|
| <b>Reinstall OS / Applications</b>                      | Completely wipes compromised components and replaces them with clean versions.       |
| <b>Update and Patch OS / Applications</b>               | Applies security updates to close vulnerabilities exploited by rootkits.             |
| <b>Verify the Integrity of System Files Regularly</b>   | Compares system files against known-good baselines using integrity-checking tools.   |
| <b>Update Anti-virus / Anti-spyware Software</b>        | Ensures the latest signatures and heuristic models are available.                    |
| <b>Prevent Installation of Unnecessary Applications</b> | Restricts what software can be installed on the system, reducing the attack surface. |

# Steganography

- Steganography is a **technique of hiding a secret message within an ordinary message** and extracting it at the destination to maintain confidentiality of data.
- Attackers use steganography to hide messages.





# Types of Steganography

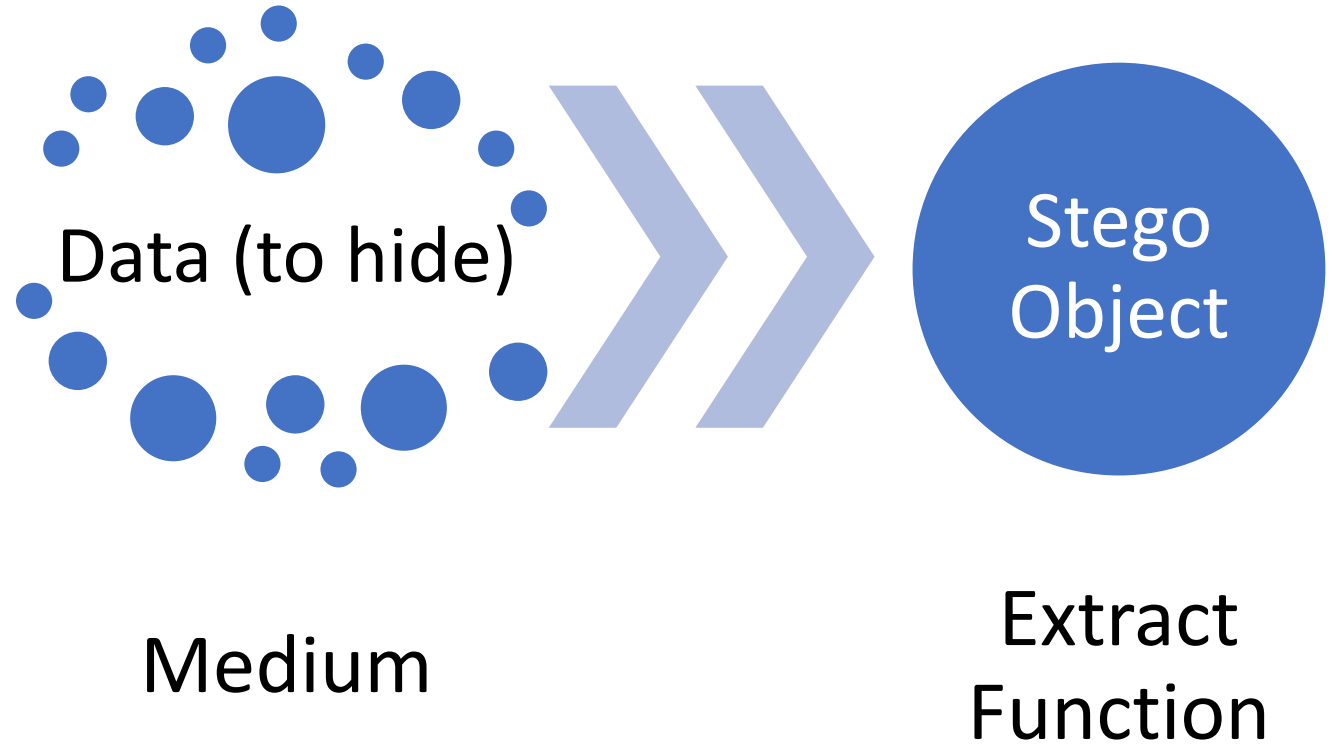
| Category                 | Sub-Type         | How it works                                                                                                  | Examples / Notes                                                                                                                      |
|--------------------------|------------------|---------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Linguistic Steganography | Semagrams        | Hides messages using symbols, shapes, spacing, or patterns rather than altering text meaning.                 | <b>Visual Semagrams:</b> drawings, marks, symbols hidden in images. <b>Text Semagrams:</b> spacing, font changes, small textual cues. |
|                          | Open Codes       | The message is present in readable text but disguised so only the intended audience understands.              | Includes <b>Jargon Codes</b> , <b>Covered Ciphers</b> , <b>Null</b> and <b>Grille</b> types.                                          |
|                          | Visual Semagrams | Uses images or symbols to embed hidden meaning.                                                               | A doodle, mark, or shape that carries hidden information.                                                                             |
|                          | Text Semagrams   | Uses text formatting, spacing, capitalization, or special characters to hide data.                            | Hidden letters in spacing or alignment.                                                                                               |
|                          | Covered Ciphers  | Encrypted or hidden messages embedded within normal-looking text.                                             | Rearranged letters, concealed cipher text.                                                                                            |
|                          | Jargon Code      | Uses specialized or coded language to hide meaning within plain text.                                         | Slang or specialized terminology known only to certain groups.                                                                        |
|                          | Null             | Uses irrelevant or filler words; the hidden message is found by removing or selecting specific letters/words. | Every nth word forms the secret message.                                                                                              |
|                          | Grille           | A stencil or mask (grille) reveals the hidden message when placed over text.                                  | Derived from “Cardan Grille” method.                                                                                                  |
| Technical Steganography  | —                | Hides data using scientific or technical methods rather than linguistic tricks.                               | Can use images, audio, video, network packets, or file headers to store hidden information.                                           |

# Steganalysis

- Reverse process of Steganography.
- Art of discovering and rendering covert messages using Steganography.
- Detects the hidden messages embedded in images, text, audio, video etc.

## Methods

- *Stego-Only*
- *Known-stego*
- *Known-message*
- *Known-cover*
- *Chosen-message*
- *Chosen-stego*



# Steganalysis attack types

| Attack Type           | What the Analyst has                                                                       | How the attack works                                                                                                         | Strengths                                                                            |
|-----------------------|--------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| <b>Stego-Only</b>     | Only the <i>stego object</i> (the file suspected of hiding data).                          | Analyst examines the stego file for statistical anomalies, noise patterns, or distortions to identify hidden data.           | Useful when nothing else is available; applies to real-world forensic situations.    |
| <b>Known-Steg</b>     | Both the <i>original cover</i> and the <i>stego object</i> .                               | Analyst compares the two versions to detect differences caused by embedding.                                                 | Highly effective because differences reveal the hidden content clearly.              |
| <b>Known-Message</b>  | The hidden <i>message</i> itself is known.                                                 | Analyst uses the known message to reverse-engineer or detect the embedding method.                                           | Helps derive patterns and detect similar future stego operations.                    |
| <b>Known-Cover</b>    | The <i>original clean cover</i> is available but the stego file may not be.                | Analyst compares expected file structure or features to suspicious files to identify changes.                                | Helpful when verifying if a specific cover was used in a stego operation.            |
| <b>Chosen-Message</b> | Analyst can choose a message and generate a stego object using the suspected stego method. | By embedding chosen content, analyst studies how the method modifies files, then uses that knowledge to detect real samples. | Very effective for algorithm fingerprinting.                                         |
| <b>Chosen-Steg</b>    | Analyst can feed chosen data into the suspected stego system/tool and examine the output.  | Allows the analyst to study how the tool transforms data and creates patterns in stego files.                                | Highly effective for understanding tool behaviour and creating detection signatures. |

# Detecting Steganography

- **Text File**

*Modifications are done based on Character Positions*

*Observe the text patterns, disturbances, language used, blank spaces etc to detect*

- **Image File**

*Observe the Size, file format, timestamp modification, colour palette.*

*Statistical Analysis method could be used for scanning images*

- **Audio File**

*Modifications are done based on LSB. Detect data existence based on Odd distortions/ Patterns*

*Statistical Analysis method could be used for audio files. Scan inaudible frequencies.*

- **Video File**

✓ Combination of Image and Audio file detection shall be used to detect data in video files

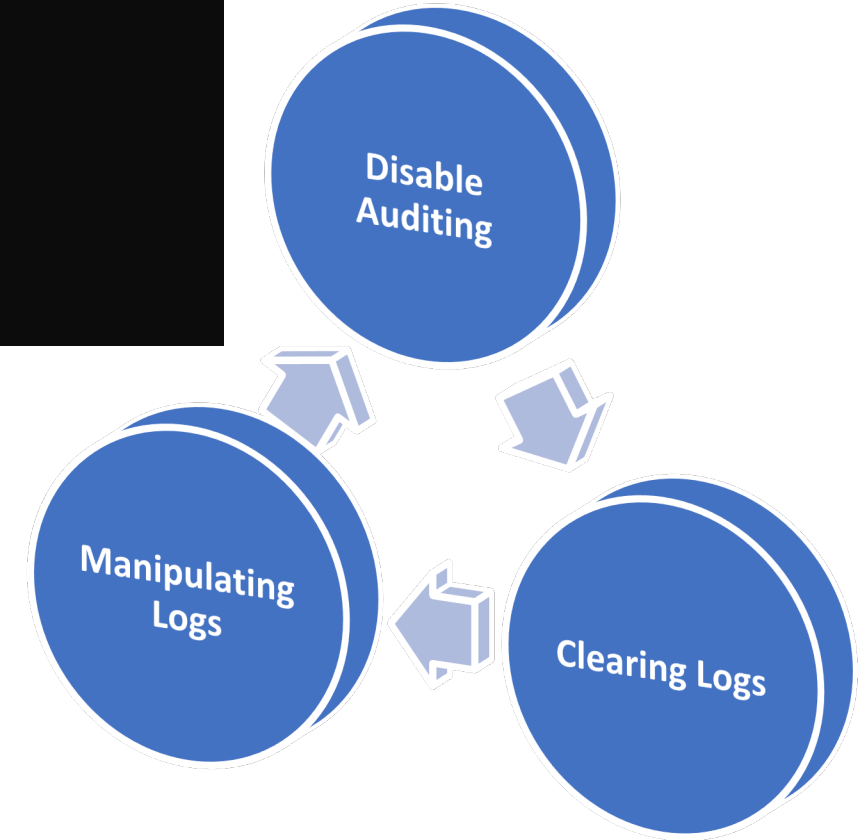
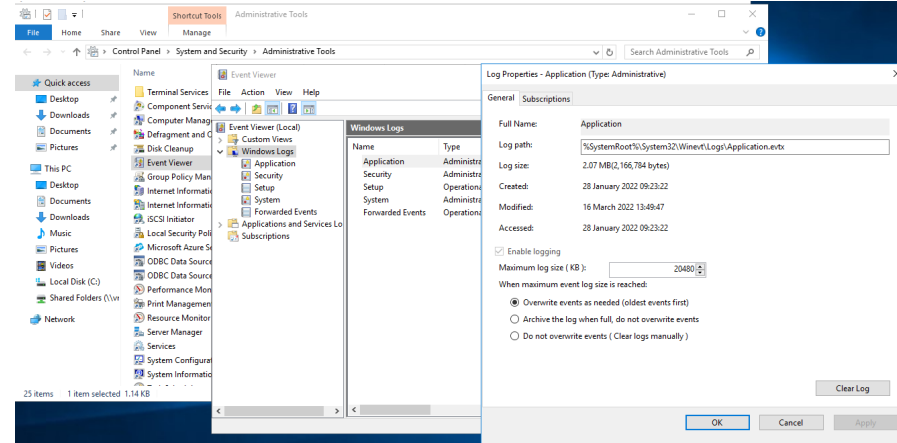
# Clearing Logs

- Disables Auditing features
- Clear or delete the system log entries
- Clear Event Logs Manually
- Delete the MRU, cookies, cache etc
- Turnoff Autocomplete
- Clear Toolbar data from browsers
- Clear the Bash History
- Use Reverse HTTP Shells
- Use Reverse ICMP Tunnels
- Use DNS Tunneling
- Use TCP Parameters

```
Use AuditPol <command> /? for details on each command

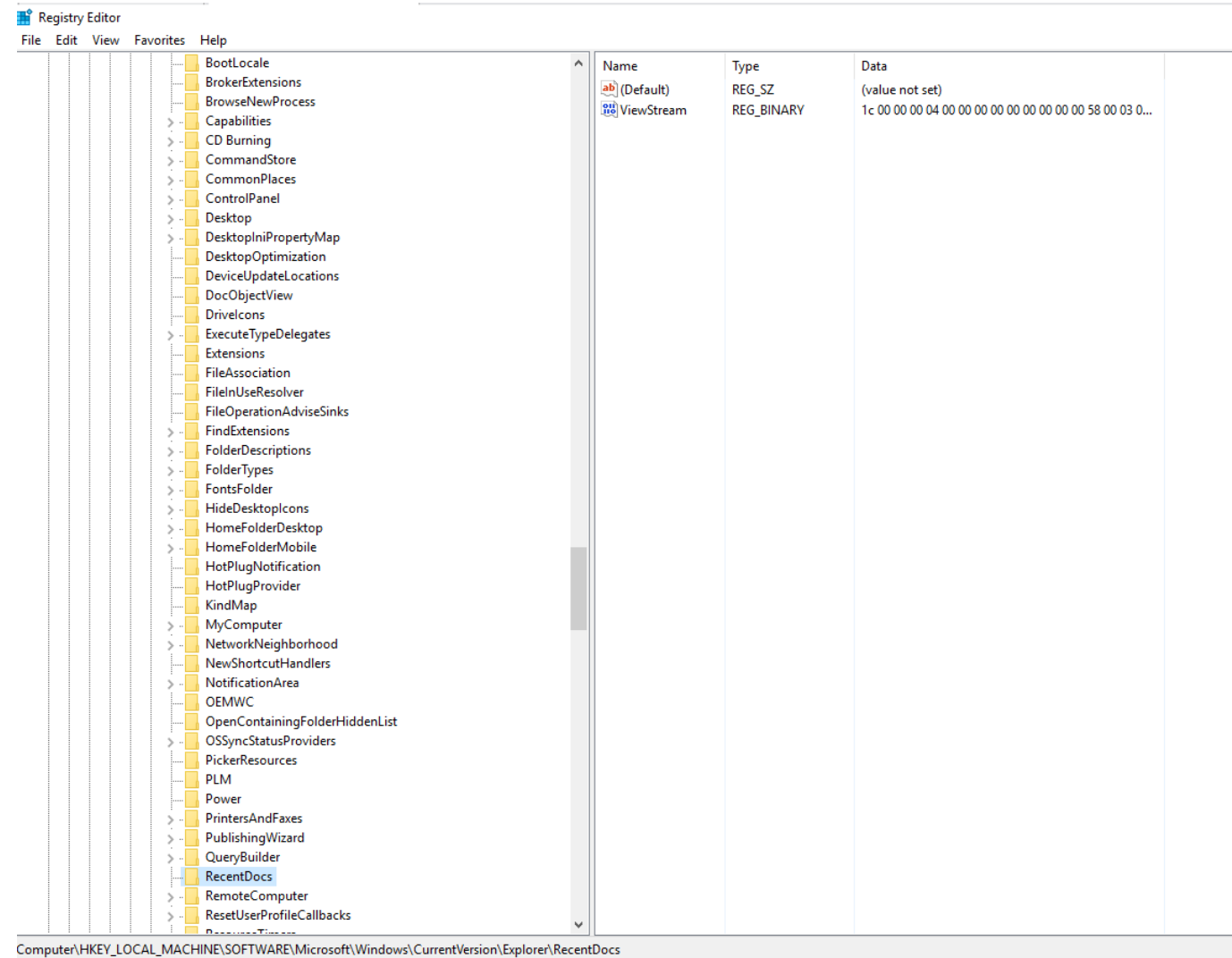
C:\windows\system32>Auditpol.exe /get /category:"Logon/Logoff"
System audit policy
Category/Subcategory          Setting
Logon/Logoff
  Logon                        Success and Failure
  Logoff                       Success
  Account Lockout              Success
  IPsec Main Mode               No Auditing
  IPsec Quick Mode              No Auditing
  IPsec Extended Mode           No Auditing
  Special Logon                 Success
  Other Logon/Logoff Events     No Auditing
  Network Policy Server         Success and Failure
  User / Device Claims          No Auditing
  Group Membership              No Auditing

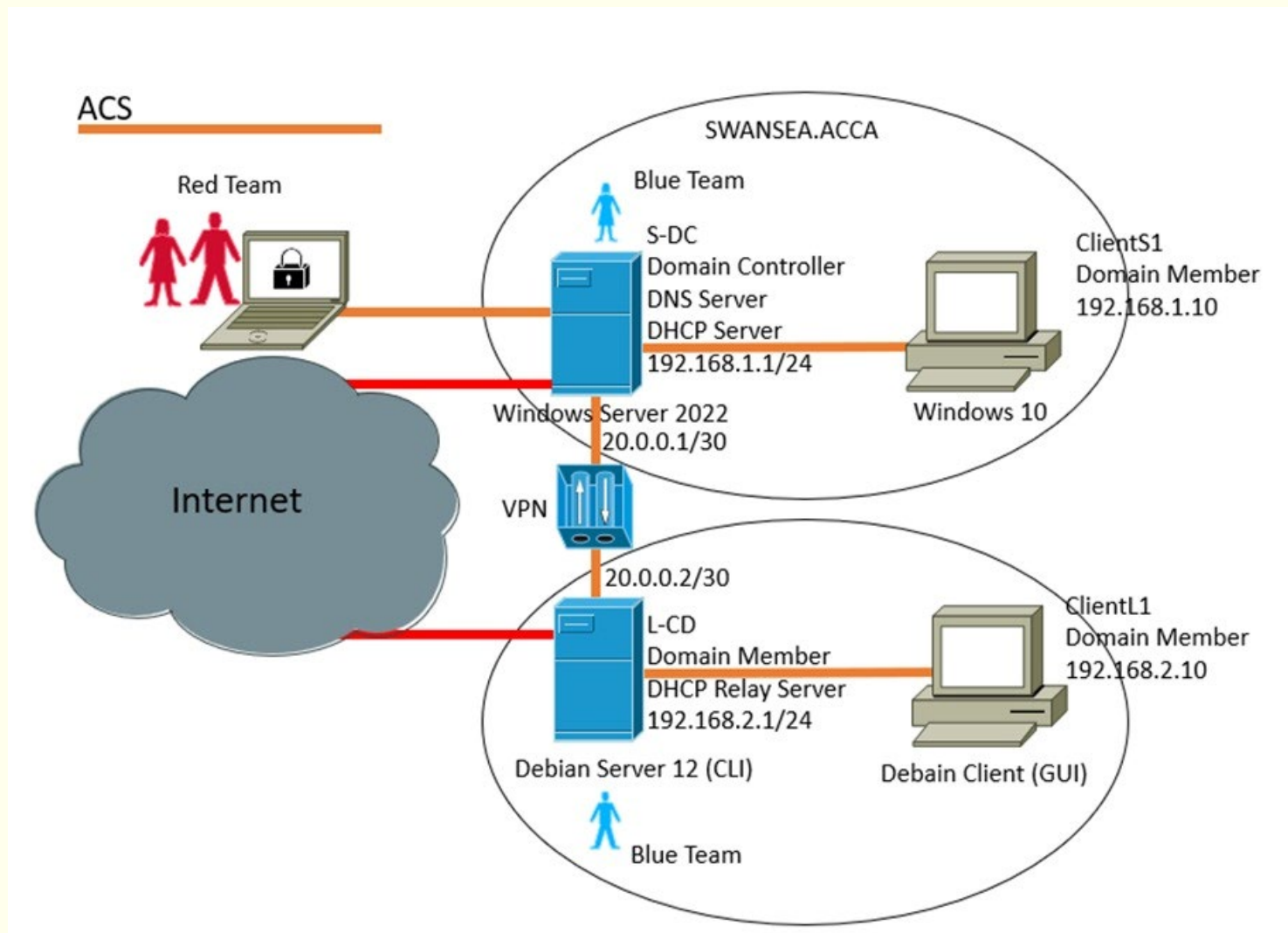
C:\windows\system32>
```



# Clearing Logs

- **Disables Auditing features**
- **Clear or delete the system log entries**
- **Clear Event Logs Manually**
- **Delete the MRU, cookies, cache etc**
- **Turnoff Autocomplete**
- **Clear Toolbar data from browsers**
- **Clear the Bash History**
- **Use Reverse HTTP Shells**
- **Use Reverse ICMP Tunnels**
- **Use DNS Tunneling**
- **Use TCP Parameters**





# Any Questions?

## References

- **EC-Council CEH**
- **Cisco Networking Academy**
- **Tevel Cyber Corps**
- **Certified Ethical Hacker (CEH) Version 10 Cert Guide, 3rd edition**
- **Wikipedia**
- **Other website sources given in this presentation**
- **\* The intension of usage is for Educational Purpose Only**