**Module 06: System Hacking**

**CHM (CEH Hacking Methodology)**

* Footprinting
* Scanning
* Enumeration
* Vulnerability Analysis
* System Hacking
  + Gaining Access
    - Cracking Passwords
    - Vulnerability Exploitation
  + Escalating Priviledges
  + Maintaining Access
    - Executing Applications
    - Hiding Files
  + Clearing Logs
    - Covering Tacks

**Microsoft Authentication**

* **SAM (Security Accounts Manager) Database**: Store user passwords, or in the AD database in domains. Passwords are **hashed**.
  + It is located at **C: \windows\system32\config\SAM.**
  + **Form:** Username: User ID: LM Hash: NTLM Hash
  + **Example:** Shiela: 1005: NO PASSWORD \*\*\*\*\*\*\*\*\*\*\*:0CB6948805F797BF2A82807973B89537: : :
  + LM hashes have been **disabled** in newer Windows OS, should be blank in those system
* **NTLM Authentication (NT LAN Manager)**: Using a **challenge/response strategy**.
  + The NTLM authentication protocol types are as follows: **NTLM authentication protocol** and **LM authentication protocol**.
  + There protocols store password in the SAM database using different hashing method.
* **Kerberos Authentication**: Microsoft has upgraded its **default authentication protocol** to Kerberos, providing a stronger authentication for C/S apps than NTLM.
  + It employs the **KDC (Key Distribution Center)**, which is a trusted third party.
  + This consists of **AS (Authentication Server)** and a **TGS (Ticket Granting Server).**
  + It use **tickets** to prove a user’s id.

**NTLM Authentication Process:**

* Include three methods of challenge-response authentication: LM, NTLMv1, and NTLMv2
* The client and server negotiate an authentication protocol, accomplished through **SSP (Security Support Provider)**

**Kerberos Authentication**

* A network authentication protocol provides strong authentication for C/S applications through **secret-key** cryptography
* Both the server and the user verify each other’s id.
* Messages sent through this protocol are protected against **replay** attacks and **evaesdropping.**

**Types of Password Attacks**

* **Non-Electronic Attacks:** Do not need technical knowledge to crack the password
  + Shoulder Surfing
  + Social Engineering
  + Dumpster Diving
* **Active Online Attacks:** Perform password cracking by directly communicating with the victim’s machine
  + Trojan/Spyware/Keyloggers
  + Dictionary Attack, Brute-Force Attack, Rule-based Attack
  + **LLMNR (Link-Local Multicast Name Resolution)/NBT-NS (NetBIOS Name Service)** poisoning
  + Password Guessing
  + Internal Monologue Attack
  + Cracking Kerberos Passwords
* **Passive Online Attacks:** Without communicating with the authorizing party
  + Wire sniffing
  + MITM attack
  + Replay attack
* **Offline Attacks:** Copy password file and try to crack password on ones own system at a different location
  + Rainbow Table Attack: **Time-memory tradeoff**
  + DNA (Distributed Network Attack)

**Password Recovery Tools**

* Elcomsoft Distributed Password Recovery:
* Hashcat:

**Tools to extract password hashes**

* **pwdump7:** Extract LM and NTLM password hashes from SAM database
* **Mimikatz**:

**Password-Cracking Tools:**

* L0phtCrack: Audit passwords and recover applications
* ophcrack: Windows password cracker based on rainbow tables.
* RainbowCrack: Crack hashes with rainbow tables. Use time-memory tradeoff algorithm to crack hashes,
* John the Ripper

**Defend against LLMNR/NBT-NS Poisoning**

* Disable LMBNR
* Disable NBT-NS

**Tools to detect LLMNR/NBT-NS poisoning**

* Vincidate
* got-responded

**Vulnerability Exploitation:** Involve the execution of multiple complex, interrelated steps to gain access to a remote system.

* identify the vulnerability
* Determine the risk associated with the vulnerability
* Determine the capability of the vulnerability
* Develop the exploit
* Select the method for delivering -local or remote
* Generate and deliver the payload
* Gain remote access

**Buffer Overflow**

* **Stack-Based Buffer Overflow:** Used for static memory allocation and stores the variables in LIFO (last-in first-out) order.
  + **Five types of registers:**
  + **EBP:** Extended Base Pointer, also known as StackBase, stores the address of the first data element stored onto the stack.
  + **ESP:** Extended Stack Pointer stores the address of the next data element to be stored onto the stack
  + **EIP:** Extended Instruction Pointer stores the address of the next instruction ot be excuted.
  + **ESI:** Extended Source Index maintains the source index for various string operations.
  + **EDI:** Extended Destination Index maintains the destination index for various string operations
* **Heap-Based Buffer Overflow:** Heap memory is dynamically allocated at runtime during the execution of the program and it stores program data. It occurs when a block of memory is allocated to a heap, and data is written without any bounds checking. It leads to overwriting dynamic object pointers, heap headers, heap-based data, virtual function table, etc.

**Windows Buffer Overflow Expoitation**

* **Perform spiking:** It allows attackers to send crafted TCP or UDP packets to the vulnerable server in order to make it crash. Help attackers to identify buffer overlflow vulnerabilities in the target applications
* **Perform fuzzing:** Use fuzzing to send **large amount of data** so that it experiences buffer overflow and overwrites the EIP. Help in identifying the number of bytes required to crash the target server. Help in determining the exact **location of the EIP**, which further helps in injecting malicious shellcode
* **Identify the offset:** Use the Metasploit framework **pattern\_create** and **pattern\_offset** ruby tools to identify the offset and exact location where EIP is being overwritten.
* **Overwrite the EIP register:** Allow attackers to identify whether the EIP can be controlled and can be overwritten with malicious shellcode
* **Identify bad characters:** Identify bad characters that may cause issues
* **Identify the right module:** Identify the right module of the vulnerable server that lacks memory protection.
* **Generate shellcode:** Use msfvenom command to generate the shellcode and inject it into the EIP to gain all access to…
* **Gain root access**

**Buffer Overflow Detection Tools**

* **OllyDbg:** Traces stack frames and program execution, and it logs arguments of known functions.

**Defend against Buffer Overflow**

* Develop program by following secure coding practices and guidelines
* Use **ASLR (address space layout randomization)** technique
* Validate arguments and minimize code that require root priviledges.
* Employ DEP (Data Execution Prevention) to mark memory regions as non-executable
* …

**Escalating Privileges:** The second stage of system hacking

* An attacker performs this that take advantage of design flaws, programming errors, bugs, and configuration oversights in the OS and software application to gain administrative access…
* **Horizontal Priviledge Escalation:** Acquire the same priviledges that have already been granted, by assuming the id of another user with the same priviledges
* **Vertical Priviledge Escalation:** Gain higher priviledge that those existing

**Privilege Escalation Using DLL Hijacking**

* Most windows app don’t use the **fully qualified path** when locading an external DLL lib. Instead, they search the dir, from which they have been loaded
* If attackers can place **a milicious DLL** **in the app dir**, it will be executed in place of the real DLL
* Attackers use tools such as **Robber** and **PowerSploit** to detect hijackable DLLs and perform DLL hijacking on the target system
* Robber: An open-source tool that helps attackers to **find executable prone to DLL hijacking**

**Privilege Escalation by Exploiting Vulnerabilities**

* Exploit software vulnerabilities by taking advantage of programming flaws in a program, service, or within the OS software or kernel, to execute malicious code
* Attackers search for an exploit based on the OS and software app on exploit sites such as **SecurityFocus** and **Exploit Database**

**Privilege Escalation Using Dylib Hijacking**

* In OS X, when applications **load an external dylib**, the loader searches for the dylib in multiple dirs
* **Dylib Hijack Scanner** helps attackers to detect dylibs that are vulnerable to hijacking attack
* Attackers use tools such as **DylibHijack** to perform dylib hjjacking on the target system

**Using Spectre and Meltdown Vulnerabilities**

* Spectre and Meltdown are vulnerabilities found in the **design of modern processor chips** from AMD, ARM, and Intel
* **The performance and CPU optimizations** in the processor, such as branch prediction, out of order execution, caching, and speculative execution, lead to these vulnerabilities
* Attacker exploit these vulnerabilities to gain unauthorized access and **steal critical system info such as credentials** and **secret keys** stored in the application’s memory, to escalate privileges.
* **Spectre Vulnerability:** Read adjacent memory locations of a process and access info for which he is not authorized. An attacker can even read the kernel memory or perform a web-based attack using JS
* **Meltdown Vulnerability:** Escalate privileges by forcing an unpriviledge process to read other adjacent memory locations such as kernel memory and physical memory. This lead to revealing critical system info such as credentials, private keys, etc.

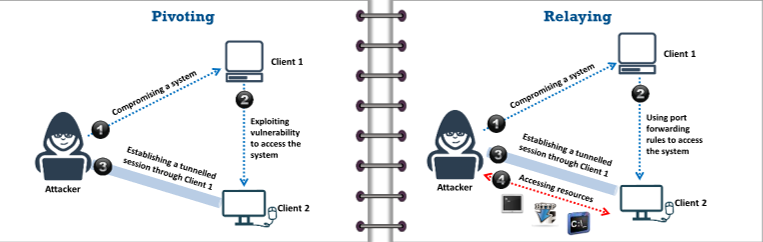
**Privilege Escalation Using Named Pipe Impersonation**

* In the Windows OS, named pipe provide **legitimate communication** between running processes. Attackers often exploit this technique to escalate privileges on the victim’s system to those of a user account **having higher access privilege**
* Attackers use tools such as Metaslpoit to perform named pipe impersonation
* Attackers use Metasploit commands such as getsystem to gain administrative-level privileges and extract password hashes of the admin/user accounts

**Privilege Escation by Exploiting Misconfigured Services**

* **Unquoted Service Paths:** In windows, the system attempts to find the location of the **executable file** to launch the service when starting a service. The executable path is **enclosed in quotation marks “”**. so that the system can easily locate the application binary. Attackers exploit services with unquoted paths running under **SYSTEM privileges** to elevate their privileges.
* **Service Object Permissions:** Misconfigured service permissions may allow an attacker to modify or **reconfigure the attributes** associated with that service. By exploiting such services, attackers can even **add new users** to the local admin group and then hijack trhe new account to elevate their privilege.
* **Unattended Installs:** Unattended install details such as **configuration settings** used during the installation process are stored in Unattend.xml file. Attackers exploit info stored in Unattend.xml to escalate privileges. It is stored in one of the following locations: **C:\Windows\panther\, C:\Windows\Panther\Unattend\, C:\Windowd\System32\, C:\Windows\System32\sysprep\**

**Pivoting and Relaying to Hack External Machines**

* Use pivoting technique to compromise a system, gain remote shell access on it, and futher **bypass the firewall to pivot via the compromised system** to **access other vulnerable systems** in the network
* Use relaying technique to access resources present on other system via the compromised system such a way that the requests to access the resources are coming from the initially compromised system.
* 

**Other Privilege Escalation Techniques**

* Access Token Manipulation
* Application Shimming
* Filesystem Permission Weakness
* Path Interception
* Scheduled Task
* Launch Daemon
* Plist Modification
* Setuid and Setgid
* Web Shell
* Abusing Sudo Rights
* Abusing SUID and SGID Permissions
* Kernel Exploits

**Privilege Escalation Tools**

* **BeRoot:** A post-exploitation tool to check common misconfiguration to find a way to sacalate privileges.
* **Linpostexp:** Obtain detailed info on the kernel, which can be used to escalate privileges on the target system

**How to Defend Against Privilege Escalation**

* Restrict the interactive logon privileges
* Run users and apps with lowest privileges
* Implement MFA and authorization
* Run services as unprivileges accounts
* Use encryption to protect sensitive data
* …..

**Defend against DLL and Dylib Hijacking**

* **Dependency Walker:** Detect many common app problems such as missing modules…
* **Dylib Hijack Scanner:** A simple utlity that will scan your computer for app that are…

**Defeng Spectre and Meltdown Vulnerabilities**

* **InSpect:** Examines and disclose any windows system’s hardware and software vulnerabilities to…
* **Spectre & Meltdown Checker:** A shell script to tell if your system is vulnerable again …

**Executing Application**

* When attackers execute malicious apps it is called **owning** the system
* The attacker executes malicious programs remotely in the victim’s machine to gather info.., gain unauthorized access…, crack the password,…
* Malicious Program that attackers execute on target: Keyloggers, spyware, backdoors, crackers

**Remote Code Execution Techniques**

* Exploitation for Client Execution
* Scheduled Task
* Service Execution
* WMI (Windows Management Instrumentation)
* WinRM (Windows Remote Management)
* Tools for Executing Applications**: Remote Exec** remotely installs applications, executes programs/scripts, and updates files and folders on Windows system throughout the network

**Keyloggers**

* **Spyrix Keylogger FREE:** Used for remote monitoring on your pc that includes recording of keystrokes, passwords, and screenshots.
* Anti-Keyloggers: **Zemana AntiLogger**

**Spyware**

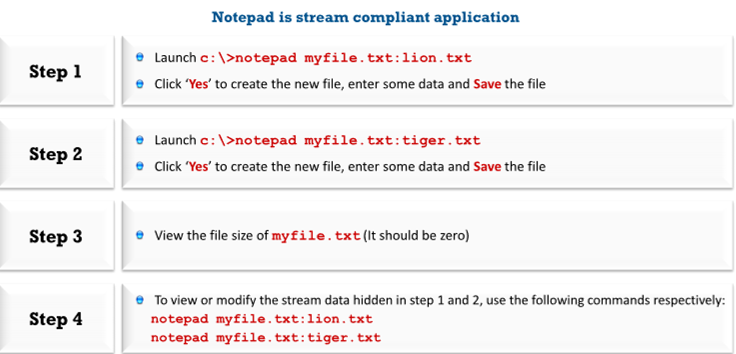
* A stealthy program that records the user’s interaction with the computer and…
* Hide its process, file and other…
* It is like a Trojan horse, which is usually bundled as a hidden component of a…
* **Spytech SpyAgent, Power Spy**
* **Anti-Spyware:** SUPERAnti Spyware

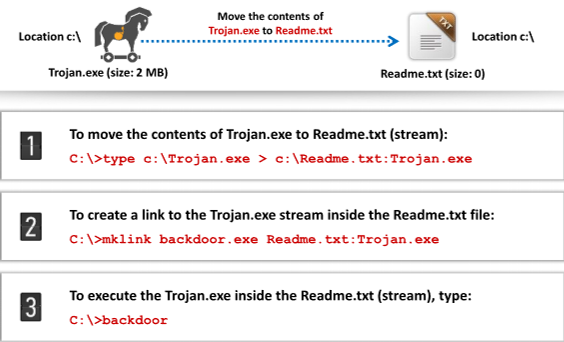
**RootKits**

* Programs that hide their presence as well as attacker’s malicious activities, granting them full access to…
* Replace certain OS calls and utilities with their own modified versions of those routines…
* Comprises of backdoor programs, DDoS programs, packet sniffers, log-wiping utilities, IRC bots, etc.
* **Types:**
  + **Hypervisor Level Rootkit:** Act as a hypervisor and modifies the boot sequence of the computer system to load the host OS as a virtual machine
  + **Hardware/Firmware Rootkit:** Hide in hardware devices or platform fireware that we are not inspected for **code integrity**
  + **Kernel Level Rootkit:** Adds malicious code or replaces the **original OS** kernel and **device driver codes**
  + **Boot Loader Level Rootkit:** Replace the original **boot loader** with the one controlled by a remote attacker
  + **Application Level/User Mode Rootkit:** Replace regular **application binaries** with a fake Trojan or modifies the behavior of exeisting applications by injecting mailicous code
  + **Library Level Rootkit:** Repleace the original system calls with fake ones to **hide info** about the attacker
* **System hooking** is the process of changing and replacing the original function pointer with a pointer provided by the rootkit in stealth mode.
* **DKOM (Direct Kernal object manipulation):** Locate and manipulate the system process in kernel memory structures and patch it.
* **Popular Rootkits:** Lojax and Scranos
  + **Lojax:** A type of **UEFI (Unified Extensible Firmware Interface) rootkit** that injects malware into the system and is automatically executed whenever the system starts up. Exploit UEFI that **acts as an interface** between the OS and the fireware
  + **Scranos:** A windows kernel rootkit that runs inside the windows OS and provide an effective mechanism, **hidden storage**, and mailicious command execution while remaining invisible. It injects its malicous code into the boot record which handles the launching of Windows at each step
  + **Horse Pill:** A linux kernel reootkil that resides inside the **initrd,** which it uses to infect the system and deceives the system owner with the use of container primitives.
  + **Necurs:** Contain backdoor functionality allowing remote ccess and control of the infected computer. Monitor and filter network activity and have been observed so send spam and install rogue security software.
* **Steps for detecting Rootkits by examining the filesystem**
  + Run “dir /s /b /ah” and “dir /s /b /a-h” inside the potentially infected OS and save the results
  + Boot into a clean CD, run “dir /s /b /ah” and “dir /s /b /a-h” on the same drive and save the results
  + Run a latest version of **WinMerge** on the two sets of results to detect file-hidhing ghostware
* Anti-Rootkits: **GMER** is an application that detects and removes rootkits by scanning processes, threads, modules, services…

**NTFS Data Steam**

* **NTFS ADS (Alternate Data Stream)** is a windows hidden stream, which contains metadata for the file, such as attributes, word count, author name and access…
* ADS can **fork data into exisiting files** without changing or altering their functionality, size, or display to file browsing ultities
* ADS allow an attacker to **inject malicious code** to files on accessible system and execute them without being detected by the user
* **Steps:**



* **NTFS Steam Manipulation**
* **Defend against NTFS Streams:**
  + Move the suspected files to FAT partition
  + Use a third-party integrity checker such as Tripewire File Integrity Manager to maintain the integrity…
  + Use programs such as Stream Detector, LADS, or ADS Detector to detect streams
  + Enable real-time antivirus scanning to…
  + Use up-to-data antivirus software…
* **Detectors: Stream Armor** discovers hidden ADS and cleans them completely

**Steganography**

* **Whitespace Steganography:** Use the **SNOW** tool to hide the message
* **Image Steg:** The info is hidden in image files of different formats such as PNG, JPG, BMP. Image steg tools replace redundant bits of imaga data with the message…Techniques includes **Least Significant Bit Insertion, Masking and Filtering, Algorithms and Transformation**.
* **Image Steg Tools:** OpenStego has function of data hiding and watermarking.
* **Document Steg:** Include the addition of white space and tabs at end of the lines.
* **Document Steg Tool:** StegoStick
* **Video Steg:** DCT (Discrete Cosine Tranorm) manipulation is used to add secret data at the time of the transformation process of the video.
* **Video Steg Tool:** OmniHide Pro
* **Audio Steg:** Omfp can be hidden in an audio file using LSB or using frequencies that are inaudible to the human ear. Some of the audio steg methods are echo data hiding, spread spectrum method, LSB coding ,tone insertion, phase encoding, etc.
* **Audio Steg Tool:** DeepSound
* **Folder Steg:** Files are hidden and encrypted within a folder and do not appear to normal windows applications, including windows explorer.
* **Folder Steg Tool:** GiliSoft File Lock Pro
* **Spam/Email Steg:** Spam emails help to communicate secretly by embedding the secret messages in some way and hiding the embedded ddata in the spam emails.
* **Spam Steg Tool:** Spam Mimic
* **Mobile Steg Tool:** Steg Master, Stegais

**Steganalysis**

* The art of discovering and redering covert messages using steg.
* Detect hidden messages embedded in images,txt…
* Challenges
  + Suspect info stream may or may not have encoded hidden data
  + Efficient and accurate detection of hidden content within digital images is difficult
  + The message could be encrypted before being inserted..
  + May have irrelevant data or noise encoded into them
* Steganalysis Methods/Attacks on Steg
  + **Steg-only:** Only the stego object is available for analysis
  + **Known-Stego:** Have access to the stego algorithm and both the cover medium and steg-object
  + **Known-Message:** Have access to the hidden message and the stego object
  + **Known-Cover:** Compare the stego-object and the cover medium to identify the hidden message
  + **Chosen-message:** Generate stego objects from a known message using specific stego tools in order to identify the stego algorithm
  + **Chosen-stego:** Have access to the stego-object and stego algorithm
  + **Chi-square:** Perform probability analysis to test whether the setgo object and original data are the same or not
  + **Distinguishing Statistical:** The attacker analyzes the embedded algorithm used to detect distinguishing statistical changes along with the length of the embedded data
  + **Blind Classifier:** A blind detector is fed with the original or unmodified data to learn the resemblance of original data from multiple perspectives
* Detect Stego
  + **Text file:** The alteration are made to the character positions to hide the data. The alterations are detected by looking for the text patterns or disburbances, language used, and an unusual amount of blank spaces
  + **Image file:** Determine changes in size, file format, the last modified timestamp, and the color palette pointing to the existence of the hidden data. The statistical analysis method is used
  + **Audio file:** Statistical analysis method is used as it involves LSB modification. The inaudible frequencies can be scanned for hidden info. Any odd distortions and patterns show the existence.
  + **Video file:** A combination of methods in image and autio files
* Detection tools: **zsteg**

**Covering Tracks:**

* Disable Auditing
  + Tools: Audipol
* Clearing Logs
  + Tools: Clear\_Event\_Viewer\_Logs.bat
* Manipulateing Logs
  + For windows: Start->Control Panel->System and security->Administrative Tools->Event viewer
  + For Linux: /var/log/messages
* Covering BASH Shell Tracks:
  + more ~/.bash\_history
* Covering Tracks on the Network/OS
  + Using Reverse HTTP Shells
  + Using Reverse ICMP Tunnels
  + Using DNS Tunneling
  + Using TCP Parameters
  + For Windows: ADS
  + For UNIX: Append a dot in front of a file name.
* Deleting Files
  + **Cipher.exe** is an in-built windows command-line tool that can be used to securely detele data by overwriting it to avoid their recovery in the future
* Disabling Windows Functionality
  + **Disable the Last Access Timestamp: fsuti**l is a ultility in windows used to set the **NTFS** volume behabior parameter, DisableLastAccess which controls…
  + **Disable Windows Hibernation:** Use the registry editor or powercfg command
  + **Disable Windows Thunmnail Cache**
  + **Disable Windows Prefetch Feature**
* Tools: CCleaner

**Defend against Covering Tracks**

* Activate logging functionality on all critical systems
* Conduct a periodic audit on IT system to ensure…
* Ensure new event do not overwrite old entries
* Configure appropriate and minimal permissions necessary…
* Maintain a separate logging server on the DMZ to store…
* Update and patch regularly
* …