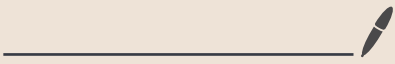
chapter 6

System Hacking



Concepts:

- Footprinting
- Scanning
- Enumeration
- Vulnerability Analysis
- Gaining Access
  - exploitation
  - Password Cracking
- Escalating privilege
- Maintaining Access
  - Executing Application
  - Hiding Files.
- Clearing Logs
  - Covering Tracks.

Hacking Methodology

Gaining Access

Microsoft Authentication

- SAM (Security Accounts Manager)
  - Stores User passwords in SAM / AD
- NTLM Authentication
  - NTLM and LM Protocol
  - These protocol store in SAM
- Kerberos Authentication
  - default authentication for windows
  - stronger authentication (client/server)

Windows SAM - hashes.

Password Hash → LM / NTLM

Shiela : 1005 : XXX : YYY :::

- Shiela → Username
- 1005 → User ID
- XXX → LM Hash
- YYY → NTLM Hash

C:\windows\System32\config\SAM

(windows Vista & later → No SAM)

NTLM Process

- Password run through hash algorithm
- computer sends request to DC (Domain Controller)
- DC sends logon challenge.
- Computer responds challenge
- DC compares & provides Authentication

Kerberos Authentication

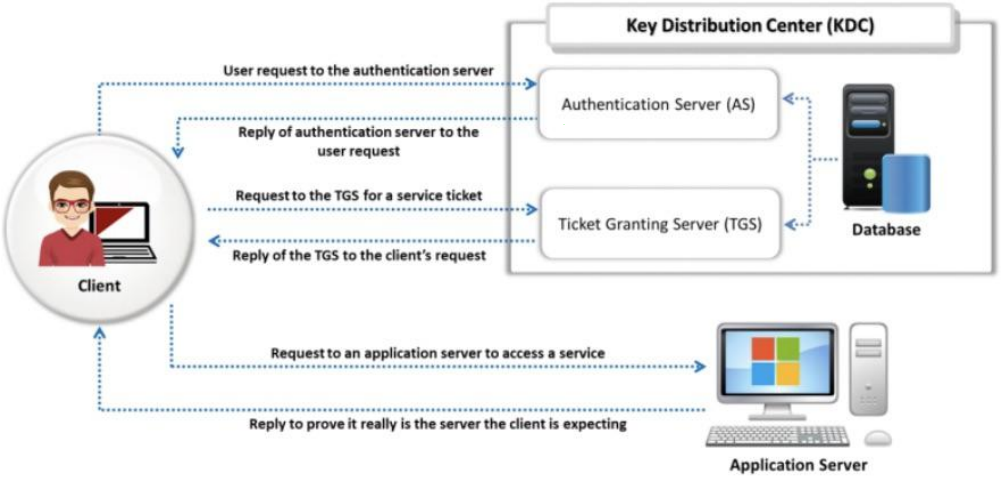


Figure 6.7: Kerberos authentication process

Password Cracking

- Non electronic
  - shoulder surfing
  - social Engineering
  - Dumpster diving
- Active online attack
  - Dictionary, BruteForce, Rule based
  - Hash Injection (Pass-the-hash)

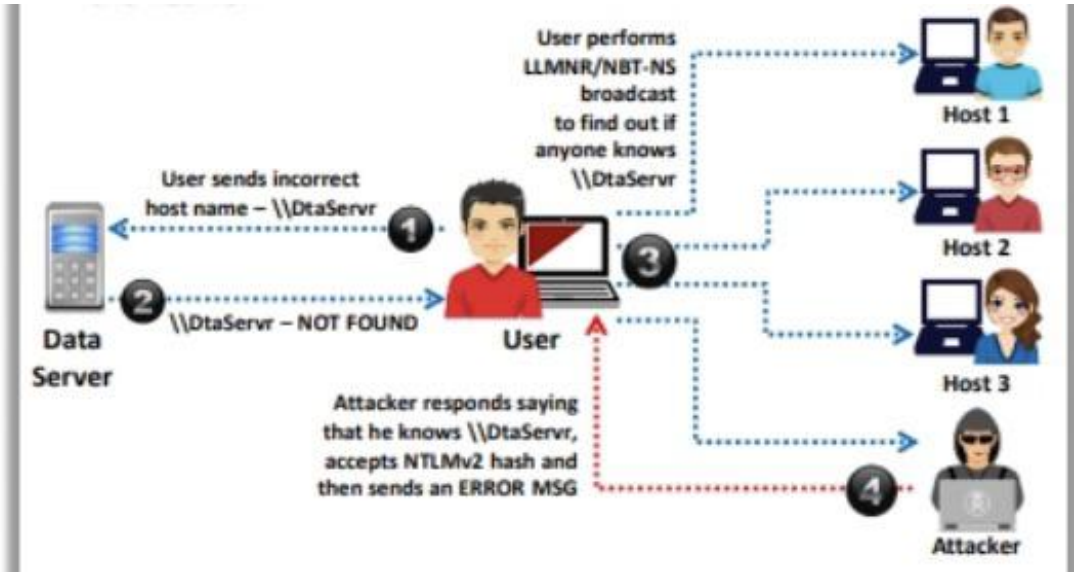
- Trojan / keyloggers / spywares
- Password Guessing
- Cracking Kerberos passwords
- LLMNR / NBT-NS Poisoning
- Internal Monologue Attack.

- Passive online attack
  - Wire Sniffing
  - MITM attack
  - replay attack.
- offline attack
  - rainbow table
  - distributed Network.

LLMNR / NBT-NS Poisoning

- Perform name resolution [windows] on same link

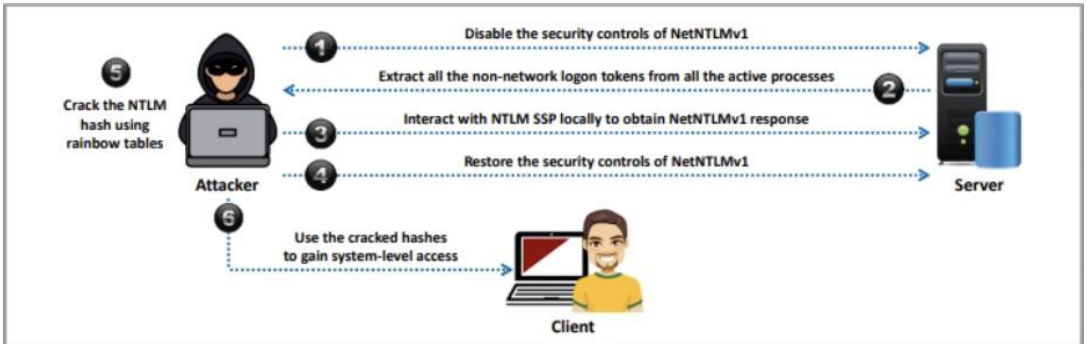
- Cracks NTLMv2 hash



Tools : Responder

Internal Monologue

- Performs using SSPI (Security Support Provider Interface)



- disable security controls by modifying values of LMCompatibilityLevel, NTLMMinClientSec, RestrictSendingNTLMTraffic

Cracking kerberos Password

- Cracking TGS (Kerberoasting)
- Cracking TGT (AS-REP)

AS-REP:

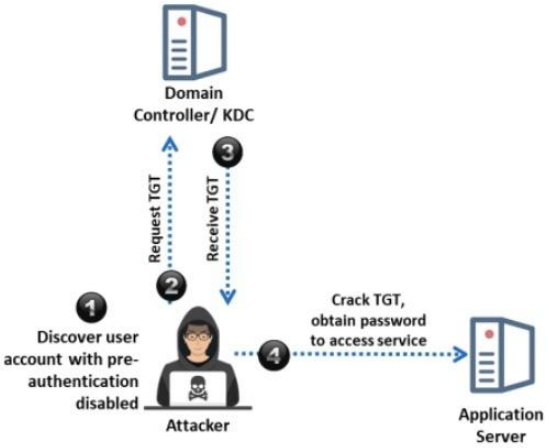


Figure 6.15: AS-REP Roasting

Kerberoasting

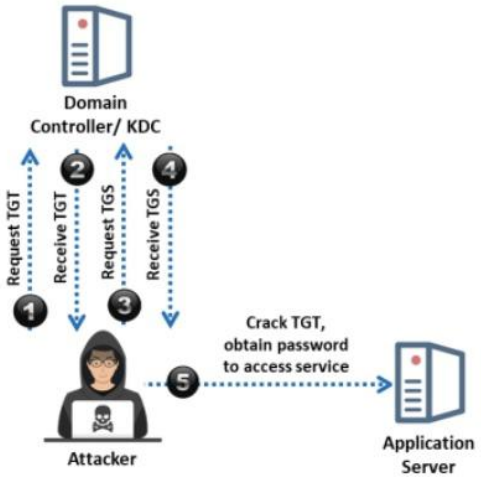


Figure 6.16: Kerberoasting

Pass the ticket

- authenticating user using kerberos ticket without providing password.
- TGT/ST can be captured based on level of access.
- Silver tickets → create tickets to call specific service

- Golden tickets → creation of TGT in Active directory.

Tools: Mimikatz.

Other online active attacks

- Combinator attacks
  (first dictionary + second dictionary)
- Fingerprint attacks
  PassPhrase → fingerprints (single/multi character)
- PRINCE attacks
  - Single input dictionary
- Toggle-Case attacks
  - Upper, lower case (dictionary)
- Markov chain attacks
  - Split each word (2-3 character)

Passive Online attack

- Wire Sniffing
  - Capture packets & find passwords
  - FTP, rlogin session, etc.

offline attack

- Distributed Network attack

  recovering passwords from hashes / files

  using unused system across network.

  DNA client – DNA Manager

Tools:

- Elcomsoft Distributed (Password recovery)

- Pwdump7 (Password hashes)
- LOphtCrack (audit password; recovery)
- OphCrack (rainbow table)
- Rainbow Crack (rainbow table; time-memory)

Defend LLMNR / NBT-NS Poisoning

LLMNR:

- open Local Group Policy Editor.

Local computer policy → Computer Configu → Administrative templates → Network → DNS client

- turn off multicast name resolution.

NBT-NS:

- open Control Panel

Network & Internet → Network & sharing center → change adapter setting

- TCP/IPv4 → Properties → advanced → WINS
- disable NetBIOS over TCP/IP

Tools:

- Vindicate (detect nameservice spoofing)
- got-responded (LLMNR/NBT-NS spoof)
- responder (detect rogued hosts)

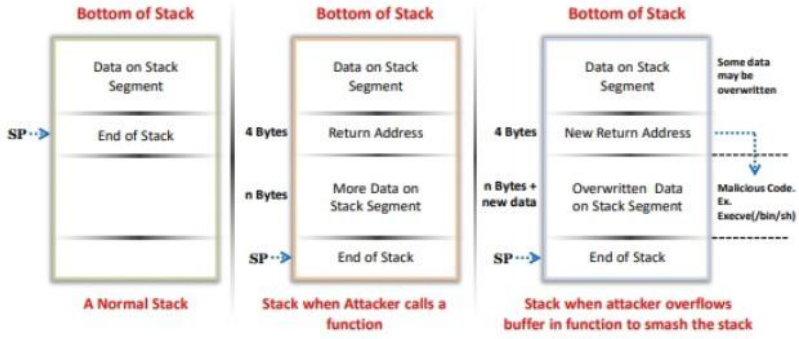
Buffer Overflow

- Stack based
- Heap based.

Stack based

- Static memory allocation
- Stores variable in LIFO order
- Operations: PUSH, POP (EIP register)

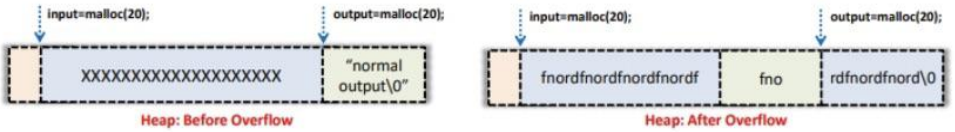
attacker replace return address in EIP register with malicious code



Heap based:

- dynamically allocated at runtime
- when allocated to heap, data written without bounds checking.

- This leads to overwriting dynamic object pointers



Window Buffer Overflow Exploitation

- Perform Spiking
- Perform Fuzzing
- Identify offset
- Overwrite EIP register
- Identify bad characters
- Identify right module
- Generate shellcode
- Gain root access

Immunity Debugger.

Spiking

- Send crafted TCP, UDP packets
- identify buffer overflow vulnerability.
- establish connection using Netcat.
- Generate spike template & perform it.
- overwrites stack register (EAX, ESP, EIP)

Fuzzing

- helps to identify no. of bytes to crash
- helps to determine exact location of EIP register
- send large amount of data to target

Identify offsets

- Use pattern_create to generate random bytes of data.
- overwrite EIP register with random bytes
- find offset of random byte in EIP

Overwrite EIP register

- run Python script to check whether EIP register can be controlled
- control & overwrite malicious shell code

Identify Bad Characters

- before code Injection, find bad character that cause issues.

Identify right module

- identify right module of server that lacks memory Protection
- Use mona.py → find the same
- Inject code & take full control of EIP
- Use nasm_shell.rb to convert assembly lang → Hex Code.

Generate shell code

- Generate shellcode using msfvenom.
- run the script by listening to a port
- Gain Access to shell of window.

Tools:

- OllyDbg (detects bufferoverflow)

Defend Buffer Overflow

- Secure Coding Practice.
- ASLR technique.
- allow compiler to add bounds
- Implement automatic bound checking
- Implement code pointer integrity.

Privilege Escalation

- gain administrative privilege by accessing non-admin user account

Types:

- Horizontal Privilege (user - user)
- Vertical Privilege (user - Admin)

DLL Hijacking

- Window don't use fully qualified path during external dll library load

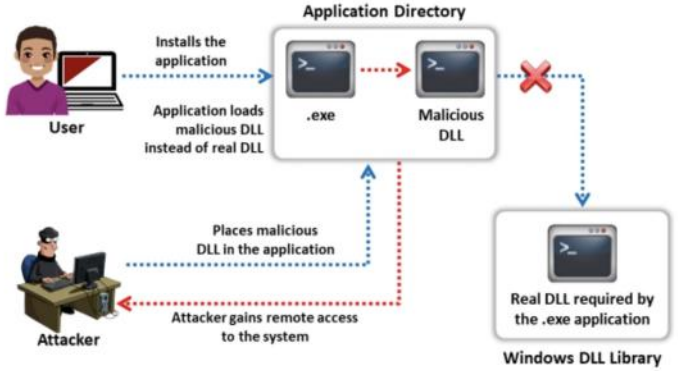


Figure 6.76: Example of privilege escalation using DLL hijacking

Tools: Robber (find vulnerable dll files)

By exploiting Vulnerability

- exploit database
- SecurityFocus.

Dylib Hijacking

In Mac, when application load external dynamic library, it search in diff. directories

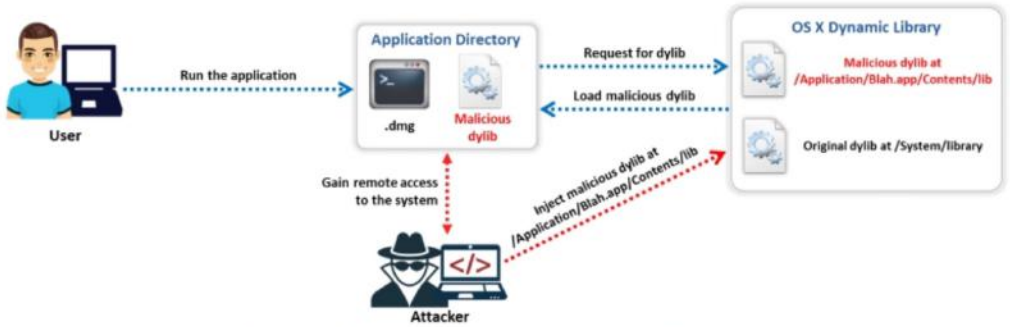


Figure 6.79: Example of privilege escalation using Dylib hijacking

Tools: Dylib Hijack Scanner (detect dylibs)

Spectre & Meltdown

- Vulnerability in AMD, ARM, Intel.
- branch prediction, out of order execution, Caching, speculative execution.
- reading adjacent memory location and Kernel memory.

Named Pipe Impersonation

- (windows) named Pipes → legitimate communication between running processes

Misconfigured Services

- Unquoted Service Paths. (quotation marks)
- Service Object Permission. (reconfigure attributes)

- unattended Installs.

Pivoting & Relaying to Hack External Machine

- discover live hosts in network
- setup routing rules
- scan ports of live system
- exploit vulnerable services.

} Pivoting

- setup port forwarding rules
- access system resources

} Relaying

Other techniques

- Access token Manipulation.
  (spoofed token – security context)
- Application shimming.

Shim- compatibility old ↔ new

- used to install backdoors.

- Filesystem Permission Weakness
- Path Interception.
- Scheduled Task
- Launch Daemon
- Plist Modification
- Setuid ay Setgid.
- web shell.
- Abusing Sudo rights
- Abusing SUID, SGID permissions
- Kernel exploits.

Tools:

- Be root (check common misconfigs)
- linpostexp (get info about kernel)

Maintaining Access

execute applications

- Keyloggers
- Backdoors
- Spyware
- Crackers

RCE Techniques

- exploiting client execution
- scheduled task
- service execution
- Windows Management Instrumentation
- Windows Remote Management.

Keyloggers

- Hardware
- Software

Hardware:

- PS/2 Keylogger.
- USB Keylog
- Bluetooth
- Hardware
- WiFi
- inside keyboard

Software:

- Application
- Kernal
- Hypervisor
- Form Grabbing
- Javascript
- Memory Injection

Spyware

- drive by download
- Masquerade - anti spyware
- web browser vulnerability.
- Cookies
- Browser add ons
- Piggybacked software Installation.

Rootkits

- hide their presence, granting full access
- comprises backdoor, DDoS, Packet sniffer, log wiping, IRC bots.

Types

- Hypervisor (modifies boot sequence)
- Hardware / Firmware (hide in devices)
- Kernel level Rootkit (OS, device drivers)
- Bootloader level (replace boot loader)
- Application level / User mode (replace binary files)
- Library level (replace original system calls)

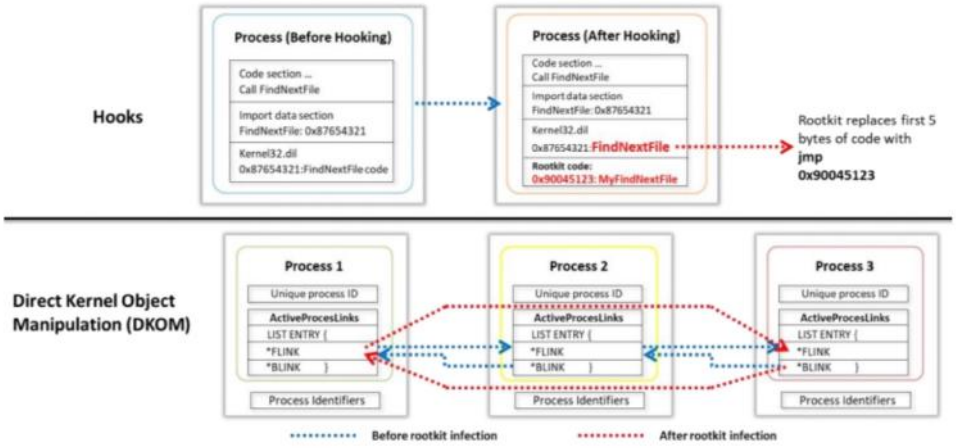


Figure 6.117: Working of a rootkit

Tools:

- Lojax (UEFI rootkit; automatically)
- Scranos (windows kernel rootkit)
- Horse Pill (reside in initrd)
- Necurs (allow remote access)

Detect Rootkits

- Integrity based
- Signature based.

- behaviour / heuristic
- Runtime execution Path Profiling.
- Cross view-based.
- Alternative trusted medium.
- Analyse Memory dump.

Tools:

- GMER (Anti-Rootkit)

NTFS Datastream

NTFS (Alternate Data Stream)

- windows hidden stream
- contains metadata of file.
- allows attacker to inject malicious code.

Detect: StreamArmor.

Steganography

- Linguistic
  - Semagrams (Visual / text)
  - Opencodes (covered / jargon)
- technical
  - invisible ink
  - Microdots.
  - Computer based
    - Substitution
    - Transform Domain
    - Spread Spectrum
    - Statistical.
    - distortion.
    - Cover Generation.

Whitespace Stegno:

- hides message in ASCII text.
- built in encryption.
- use snow tool to hide message

Image Stegno:

- hide message in image.
- replace redundant bits of image.

Techniques used:

- Least Significance bit Insert
- Masking & filtering
- Algorithms & Transformation

Document stegno:

- addition of white space, tabs (EOL)
- stegostick tool

Video Stegno

- hide secret in coorier video
- Discrete Cosine Transform (DCT) manipulation.
- Omnihide Pro

Audio Stegno

- Using LSB / Frequencies (20000 Hz)
- echo data hiding, LSB coding
- tone insertion, phase encoding
- Tool: DeepSound.

Folder stegno

- files hidden, encrypted within folder.
- Tools: Gilisoft File Lock Pro

Steganalysis

- reverse process of steganography
- detects hidden message.

Stego only (only stego)
Known stego (both cover + stego)
Known message (access hidden message)
Known cover (compares both stego + cover)
Chosen message (generate stego, identify message)
Chosen stego (access stego, algorithm)
Chi-square (probability analysis)
Blind classifier (learn resemblance with original)
Distinguishing Statistical. (learn statistical changes)

Tools detection: zsteg

Covering Tracks

- Disable auditing
- clearing logs
- Manipulating logs
- Covering tracks, (Network/os)
- deleting files
- disable windows functionality.

- **Disable auditing**
  - disable auditing immediately after access
  - use auditpol to disable system audit
- **clearing logs**
  - use meterpreter shell to wipe out logs or .bat utility.

- Use clear-Eventlog command.
- wevutil utility to clear logs.
- or manually delete logs in
  * Windows: Event Viewer
  * Linux: /var/log/messages.
- To remove online tracks:
  * remove Most recently used, Cookies, caches, autocomplete, toolbar data
  * Windows 10 → Show recent app clear
  * registry → Explorer recent Docs.
- Cover Bash shell (~/.bash_history)
  * export HISTSIZE=0.
  * history -c

* cat /dev/null > ~.bash_history
* shred bash_history

- Cover Networks
  * using reverse HTTP shells
  * Using reverse ICMP tunnels
  * Using DNS tunneling
  * Using TCP Parameters
- Cover OS (Windows (Unix)
  * Windows → ADS hide files.
  * Unix → append dot in files.
- Delete files using cipher.exe.

- Disable windows functionality
  * last access timestamp (fsutil)
  * windows Hibernation
  * windows virtual memory (Paging)
  * System restore point
  * windows thumbnail cache
  * Windows prefetch feature.

Tools:

- ccleaner