Pentesting Active Directory - Part 3 | Recon with AD Module, Bloodhound, PowerView & Adalanche

hacklido.com/blog/864-pentesting-active-directory-part-3-recon-with-ad-module-bloodhound-powerview-adalanche

- Edited

Let’s learn about generic reconnaissance & Enumeration in AD - Using AD Module, Bloodhound, PowerView & Adalanche

The process of finding a Domain Controller involves two main steps. The first step is to use the `nmap` command with the `--script broadcast-dhcp-discover` option. This script allows you to discover DHCP servers on a local subnet by broadcasting a DHCP request.

The second step is to look for machines with open ports 389, 88, and 53. These ports are commonly associated with Domain Controller services. Once you have a list of IP addresses and open ports, you can use the `grep` command to filter out only the IP addresses that have port 53 open. This list of IP addresses will be your target for enumerating file shares.

The process of enumerating file shares is done with the `crackmapexe` command, followed by the `smb` option and the IP range you want to scan (in this case, `xx.xx.xx.0/24`). This command will search for open SMB shares on the specified IP range and report any information it can gather about the shares.

Finding a Domain Controller

Step 1: Discover DHCP servers on a local subnet using the nmap command with the –script broadcast-dhcp-discover option. This will help you identify potential Domain Controllers in the network.

```
nmap --script broadcast-dhcp-discover
```

Step 2: Filter the machines with open ports 389, 88, and 53, which are commonly associated with Domain Controller services.

```
nmap -p 389,88,53 -oG xxx.txt xx.xx.xx.0/24 cat xxx.txt | grep "\\:53\\b"
>domain_controllers.txt
```

Enumerating File Shares

Use the CrackMapExec tool with the smb option to scan the IP range for open SMB shares and gather information about them.

```
crackmapexec smb xx.xx.xx.0/24 --shares
```

This command will output information about any accessible SMB shares within the specified IP range. Review the output to identify file shares that may contain sensitive information or provide further access to the network.

Bloodhound Recon

BloodHound is a web application that uses graph theory to uncover relationships within an Active Directory environment. It helps both attackers and defenders identify complex attack paths and understand privilege relationships in AD environments.

Installing BloodHound

```
sudo apt-get update sudo apt-get install bloodhound
```

Access the Neo4j console: **http://localhost:7474/** (default credentials: neo4j:neo4j)

Enumeration and Data Ingestion

Enumerate the victim domain using SharpHound to generate a JSON file describing relationships and permissions between AD objects. Import this file into BloodHound to visualize potential vulnerabilities.

Utilizing SharpHound:

Download SharpHound from Github: **https://github.com/BloodHoundAD/SharpHound**

PowerShell Execution Policy:

To bypass the default “Restricted” execution policy, use:

```
powershell -ep bypass
```

Running the Tools:

Execute the SharpHound script:

```
powershell -ep bypass . .\sharphound.ps1
```

Run the Invoke-BloodHound command:

```
Invoke-BloodHound -CollectionMethod All -Domain domain.local -ZipFileName file.zip
```

Analysis

Move the resulting zip file to the attacker machine and upload it to BloodHound for analysis.

1. **Move the zip file to the attacker machine**:
    - If both machines are on the same network, you can use a file transfer tool such as `scp` (for Linux) or WinSCP (for Windows) to transfer the zip file to the attacker machine.
    - Alternatively, you can use a shared folder or cloud storage to upload the zip file from the victim machine and download it on the attacker machine.
2. **Upload the zip file to BloodHound**:
    - Open BloodHound on the attacker machine, and log in with the Neo4j credentials (default: neo4j:neo4j).
    - In the BloodHound interface, click on the “Upload Data” button in the upper right corner (it looks like a cloud with an upward arrow).
    - Browse to the location where you saved the transferred zip file, select it, and click “Open” to start the upload process.
3. **Analyze the data in BloodHound:**
    - Once the data is uploaded, BloodHound will visualize the relationships and permissions between Active Directory objects, allowing you to explore the network, identify potential vulnerabilities, and discover attack paths.
    - You can use BloodHound’s built-in queries or create custom queries to investigate specific relationships or vulnerabilities within the AD environment.

PowerView Recon

PowerView Overview

PowerView is a PowerShell tool for Windows domain situational awareness. It replaces various “net *” commands with pure-PowerShell alternatives, leveraging PowerShell AD hooks and the Win32 API functions for domain tasks.

PowerView Metafunctions

PowerView includes custom-written user-hunting functions to track users on a network and determine the machines they are logged into. Additionally, it can check machines on which the current user has local administrator access.

Domain Trust Enumeration and Exploitation:

PowerView has functions for enumerating and exploiting domain trusts. Each function comes with its options and usage descriptions. To get detailed output on the underlying functionality, pass the -Verbose or -Debug flags.

Miscellaneous Functions:

```
Export-PowerViewCSV
Resolve-IPAddress
ConvertTo-SID
Convert-ADName
ConvertFrom-UACValue
Add-RemoteConnection
Remove-RemoteConnection
Invoke-UserImpersonation
Invoke-RevertToSelf
Get-DomainSPNTicket
Invoke-Kerberoast
Get-PathAcl
```

Domain Functions:

```
Get-DomainDNSZone
Get-DomainDNSRecord
Get-Domain
Get-DomainController
Get-Forest
Get-ForestDomain
Get-ForestGlobalCatalog
Find-DomainObjectPropertyOutlier
Get-DomainUser
New-DomainUser
Get-DomainUserEvent
Get-DomainComputer
Get-DomainObject
Set-DomainObject
 Get-DomainObjectAcl
 Add-DomainObjectAcl
 Find-InterestingDomainAcl
 Get-DomainOU
 Get-DomainSite
 Get-DomainSubnet
 Get-DomainSID
 Get-DomainGroup
 New-DomainGroup
 Get-DomainManagedSecurityGroup
 Get-DomainGroupMember
 Add-DomainGroupMember
 Get-DomainFileServer
 Get-DomainDFSShare
```

GPO Functions:

```
Get-DomainGPO
Get-DomainGPOLocalGroup
Get-DomainGPOUserLocalGroupMapping
Get-DomainGPOComputerLocalGroupMapping
Get-DomainPolicy
```

Computer Enumeration Functions:

```
Get-NetLocalGroup
Get-NetLocalGroupMember
Get-NetShare
Get-NetLoggedon
Get-NetSession
Get-RegLoggedOn
Get-NetRDPSession
Test-AdminAccess
Get-NetComputerSiteName
Get-WMIRegProxy
Get-WMIRegLastLoggedOn
Get-WMIRegCachedRDPConnection
Get-WMIRegMountedDrive
Get-WMIProcess
Find-InterestingFile
```

PowerView is an essential tool for Windows domain situational awareness and security, offering a range of powerful functions and metafunctions for security professionals.

Domain Enumeration

Using PowerView

Powerview v.3.0 | Powerview Wiki

- **Get Current Domain:** `Get-Domain`
- **Enumerate Other Domains:** `Get-Domain -Domain <DomainName>`
- **Get Domain SID:** `Get-DomainSID`
- **Get Domain Policy:**

```
Get-DomainPolicy

#Will show us the policy configurations of the Domain about system access
or kerberos
Get-DomainPolicy | Select-Object -ExpandProperty SystemAccess
Get-DomainPolicy | Select-Object -ExpandProperty KerberosPolicy
```

- **Get Domain Controllers:**

```
Get-DomainController
Get-DomainController -Domain <DomainName>
```

- **Enumerate Domain Users:**

```
  #Save all Domain Users to a file
  Get-DomainUser | Out-File -FilePath .\DomainUsers.txt

  #Will return specific properties of a specific user
  Get-DomainUser -Identity [username] -Properties DisplayName, MemberOf |
Format-List

  #Enumerate user logged on a machine
  Get-NetLoggedon -ComputerName <ComputerName>

  #Enumerate Session Information for a machine
  Get-NetSession -ComputerName <ComputerName>

  #Enumerate domain machines of the current/specified domain where specific
users are logged into
  Find-DomainUserLocation -Domain <DomainName> | Select-Object UserName,
SessionFromName
```

- **Enum Domain Computers:**

```
  Get-DomainComputer -Properties OperatingSystem, Name, DnsHostName | Sort-
Object -Property DnsHostName

  #Enumerate Live machines
  Get-DomainComputer -Ping -Properties OperatingSystem, Name, DnsHostName |
Sort-Object -Property DnsHostName
```

- **Enum Groups and Group Members:**

```
  #Save all Domain Groups to a file:
  Get-DomainGroup | Out-File -FilePath .\DomainGroup.txt

  #Return members of Specific Group (eg. Domain Admins & Enterprise Admins)
  Get-DomainGroup -Identity '<GroupName>' | Select-Object -ExpandProperty
Member
  Get-DomainGroupMember -Identity '<GroupName>' | Select-Object
MemberDistinguishedName

  #Enumerate the local groups on the local (or remote) machine. Requires
local admin rights on the remote machine
  Get-NetLocalGroup | Select-Object GroupName

  #Enumerates members of a specific local group on the local (or remote)
machine. Also requires local admin rights on the remote machine
  Get-NetLocalGroupMember -GroupName Administrators | Select-Object
MemberName, IsGroup, IsDomain

  #Return all GPOs in a domain that modify local group memberships through
Restricted Groups or Group Policy Preferences
  Get-DomainGPOLocalGroup | Select-Object GPODisplayName, GroupName
```

- **Enumerate Shares:**

```
  #Enumerate Domain Shares
  Find-DomainShare

  #Enumerate Domain Shares the current user has access
  Find-DomainShare -CheckShareAccess

  #Enumerate "Interesting" Files on accessible shares
  Find-InterestingDomainShareFile -Include *passwords*
```

- **Enum Group Policies:**

```
  Get-DomainGPO -Properties DisplayName | Sort-Object -Property DisplayName

  #Enumerate all GPOs to a specific computer
  Get-DomainGPO -ComputerIdentity <ComputerName> -Properties DisplayName |
Sort-Object -Property DisplayName

  #Get users that are part of a Machine's local Admin group
  Get-DomainGPOComputerLocalGroupMapping -ComputerName <ComputerName>
```

- **Enum OUs:**

```
  Get-DomainOU -Properties Name | Sort-Object -Property Name
```

- **Enum ACLs:**

```
  # Returns the ACLs associated with the specified account
  Get-DomaiObjectAcl -Identity <AccountName> -ResolveGUIDs

  #Search for interesting ACEs
  Find-InterestingDomainAcl -ResolveGUIDs

  #Check the ACLs associated with a specified path (e.g smb share)
  Get-PathAcl -Path "\\Path\Of\A\Share"
```

- **Enum Domain Trust:**

```
  Get-DomainTrust
  Get-DomainTrust -Domain <DomainName>

  #Enumerate all trusts for the current domain and then enumerates all trusts
for each domain it finds
  Get-DomainTrustMapping
```

- **Enum Forest Trust:**

```
Get-ForestDomain
Get-ForestDomain -Forest <ForestName>

#Map the Trust of the Forest
Get-ForestTrust
Get-ForestTrust -Forest <ForestName>
```

- **User Hunting:**

```
#Finds all machines on the current domain where the current user has local admin access
Find-LocalAdminAccess -Verbose

#Find local admins on all machines of the domain
Find-DomainLocalGroupMember -Verbose

#Find computers were a Domain Admin OR a spesified user has a session
Find-DomainUserLocation | Select-Object UserName, SessionFromName

#Confirming admin access
Test-AdminAccess
```

:heavy_exclamation_mark: **Priv Esc to Domain Admin with User Hunting:** \
I have local admin access on a machine -> A Domain Admin has a session on that machine -> I steal his token and impersonate him -> Profit!

Using AD Module

- **Get Current Domain:** `Get-ADDomain`
- **Enum Other Domains:** `Get-ADDomain -Identity <Domain>`
- **Get Domain SID:** `Get-DomainSID`
- **Get Domain Controlers:**

```
Get-ADDomainController
Get-ADDomainController -Identity <DomainName>
```

- **Enumerate Domain Users:**

```
Get-ADUser -Filter * -Identity <user> -Properties *

#Get a spesific "string" on a user's attribute
Get-ADUser -Filter 'Description -like "*wtver*"' -Properties Description | select Name, Description
```

- **Enum Domain Computers:**

```
Get-ADComputer -Filter * -Properties *
Get-ADGroup -Filter *
```

- **Enum Domain Trust:**

```
Get-ADTrust -Filter *
Get-ADTrust -Identity <DomainName>
```

- **Enum Forest Trust:**

```
Get-ADForest
Get-ADForest -Identity <ForestName>

#Domains of Forest Enumeration
(Get-ADForest).Domains
```

- **Enum Local AppLocker Effective Policy:**

```
Get-AppLockerPolicy -Effective | select -ExpandProperty RuleCollections
```

Using BloodHound

Remote BloodHound

Python BloodHound Repository or install it with `pip3 install bloodhound`

```
bloodhound-python -u <UserName> -p <Password> -ns <Domain Controller's Ip> -d
<Domain> -c All
```

On Site BloodHound

```
#Using exe ingestor
.\SharpHound.exe --CollectionMethod All --LdapUsername <UserName> --LdapPassword
<Password> --domain <Domain> --domaincontroller <Domain Controller's Ip> --
OutputDirectory <PathToFile>

#Using PowerShell module ingestor
. .\SharpHound.ps1
Invoke-BloodHound -CollectionMethod All --LdapUsername <UserName> --LdapPassword
<Password> --OutputDirectory <PathToFile>
```

Using Adalanche

Remote Adalanche

```
# kali linux:
./adalanche collect activedirectory --domain <Domain> \
--username <Username@Domain> --password <Password> \
--server <DC>

# Example:
./adalanche collect activedirectory --domain windcorp.local \
--username spoNge369@windcorp.local --password 'password123!' \
--server dc.windcorp.htb
## -> Terminating successfully

## Any error?:

# LDAP Result Code 200 "Network Error": x509: certificate signed by unknown
authority ?

./adalanche collect activedirectory --domain windcorp.local \
--username spoNge369@windcorp.local --password 'password123!' \
--server dc.windcorp.htb --tlsmode NoTLS --port 389

# Invalid Credentials ?
./adalanche collect activedirectory --domain windcorp.local \
--username spoNge369@windcorp.local --password 'password123!' \
--server dc.windcorp.htb --tlsmode NoTLS --port 389 \
--authmode basic

# Analyze data
# go to web browser -> 127.0.0.1:8080
./adalanche analyze
```

Useful Enumeration Tools

Home for infosec writers and readers.

Create your account today and explore more content on this platform. You can also start blogging and be inspiration for others 😎

7 days later

admiralarjun changed the title to **Pentesting Active Directory - Part 3 | Recon with AD Module, Bloodhound, PowerView & Adalanche** 13 days ago.

4 days later

- **ShahisnutaNhemhafuki**
- 9 days ago

I am really having good AD time as the one with zero idea about it. Thank you so much .