[Backup] Domain Enumeration Methodology

Hey folks, today I start a new series of articles to discuss Active Directory Exploitation. This is the first article, we focus on domain enumeration. We assume you have already had an initial shell on a domain computer, no matter it is Windows domain computer or Linux domain computer, because we will discuss both of them : D

Be aware that it is not an article which focuses on the detailed usage of tool and command, we focus on methodology.

Enumeration on Windows

After exploiting the public-facing service, we could get an shell as a local service account, at this moment, we are not under a domain context. It is helpful to find a way to escalate privilege. Finally, we get **SYSTEM** privilege, it means we are under the domain computer account’s context, so we can start to enuemrate the domain. Here is a checklist for myself as I initially get access to domain context.

0: One-Click Vulnerabilities

In recent years, there are few zero-day vulnerabilities which can help us compromise the whole domain immediately. Though they could have been fixed in the environment, but it does not hurt to have a try!

- **CVE-2021–42278:** No PAC Vulnerability
- **CVE-2022–26809:** RPC RCE
- **CVE-2022–26923:** ADCS Vulnerability
- **CVE-2020–1472:** Zerologin Vulnerability
- **MS14–068:** Kerberos Vulnerability

1: Domain User

- **User Description**: Though at many time, the description may be blank, but if it is not blank, user description may reveal the role of the domain user. Such as server admin, developer, etc.
- **Kerberos Pre-Authentication**: If some domain users do not have pre-auth enabled, we can ASREPRoast them and get krb5asrep hashes. If we are lucky, we have a chance to crack those hashes offline and get plaintext credential.
- **SPN**: If a domain user has SPN, it is a service account. We can Kerberoast them and get krb5tgs hashes. If we are lucky, we have a chance to crack those hashes offline and get plaintext credential.
- **Group Membership**: Each domain user at least belongs to “Domain User” group, but if any domain user belongs to more than this group, we must check which groups they belong to.

2: Domain Group

- **Group Description**: Just as User Description.
- **Group Type**: If a group is custom one, we need to pay more attention to it, what rights does the group have?

3: Foreign Members

If a foreign member is compromised, we have a chance to pivot to the other domain/forest.

4: Domain Computer

Take note of all domain computers’ **FQDN** and **IP addresses**.

- **Windows Computer**
- **Linux Computer**: Typically linux domain computers allow **SSH** access for domain users by default.

5: Existing Sessions and Processes

- Processes owned by other domain users
- Available Token

After getting SYSTEM privilege, we can impersonate any logged domain users. If impersonated user has specific rights, we could move to other machines even domains.

6: Owned Users' Permission

- **RDP** Access to Other Computers:
- **Local Admin** Privilege to Other Computers:
- **WinRM** Access to Other Computers:
- **DACL**: Such as ForceChangePassword, GenericWrite, etc.

7: Service Access

- **SMB**: If compromised users have access to **C$/ADMIN$** on other domain computer(s), it means the user has local admin privilege over the computer. Apart from C$/ADMIN$, also pay attention to any readable/writable custom shares, such as "dev", which could store source code of an app.
- **FTP**: If we have access, check any juicy file inside it.
- **Web**: If we can exploit an internal web app, we could move to the other machine.
- **SQL**: Abuse xp_cmdshell and SQL Link to execute command on other machines.

8: GPO

By enumerating GPO, we can take a look at current domain's special settings for specific OUs. We may not know detailed settings for a GPO, but we can infer them according to GPO name or description. GPO may also be helpful for us to move to other machines. For example, a GPO can grant some users **RDP** or **WinRM** access to specific machines.

9: Delegation

Typically, delegation is helpful for us to get command execution on other host(s). But we also need to be aware that some users and computers are disallowed to be delegated, such as domain admin, because they have high privilege.

- **Unconstrained Delegation**: It is the most powerful for us, we could compromise multiple users and computers.
- **Constrained Delegation**: We could be able to move to the computer by abusing S4U.
- **Resource Based Constrained Delegation**: If compromised computers or users have GenericWrite permission over a computer, we could finally move to the computer by abusing S4U.

10: ADCS

- Vulnerable template
- **CVE-2022–26923**

11: Trust

Domain Trust will be very helpful to us especially when we compromised domain admin

- **Within Forest**: The trust is always bi-directional, we can abuse golden ticket or trust key
- Between Forest
- **Bi-directional**: Abuse trust key or golden ticket, but be aware of SID filter.
- **Inbound**: Check if any domain user is a foreign member in target domain
- **Outbound**: Can be abused via SQL Link, logged foreign member, etc.

Enumeration on Linux

Sometimes, the public facing server is Linux OS, such as a web server. After exploiting the web app, we successfully get access to the Linux server as a normal user or privileged user.

As A Normal User

Since we are logged as a normal user, we cannot get access to all files. But sometimes, some files' permission could be misconfigured, as a result, a normal user can access them. Otherwise, we'd better find a way to escalate ourself to root.

As A Privileged User

As a privileged user, where we are going to gather domain information?

- **1: ccache file**

ccache files hold the Kerberos credentials for a user authenticated to a linux domain computer. If there is any active domain user session, we can see ccache files in **/tmp**, the file is in the form of **krb5cc_xxxxx**. We can pass ccache file directly on Linux machine, or use impacket to convert it to .kirbi form and pass it to current session on a Windows machine.

- **2: keytab file**

keytab file contains mappings between Kerberos Principal names and DES-encrypted keys that are derived from the password used to log into the Kerberos Key Distribution Center (KDC). We can use a script (https://github.com/sosdave/KeyTabExtract)to retrieve credentials from it. Each linux domain computer has its keytab file at /etc/krb5.keytab, it is accessible for root by default.

Revision #1
Created 28 February 2024 18:22:59 by winslow
Updated 28 February 2024 19:38:59 by winslow