Internal Reconnaissance Protection using NetCease and SAMRi10

blog.netwrix.com/2022/11/18/making-internal-reconnaissance-harder-using-netcease-and-samri1o

Joe Dibley

What is Internal Reconnaissance?

Internal reconnaissance is one of the first steps an attacker will take once they have compromised a user or computer account in your network. Using various tools or scripts, they enumerate and collect information that will help them identify what assets they should try to compromise next to get what they want. For example, BloodHound will map out attack paths that can enable an adversary to escalate their privileges from ordinary user to admin.

Handpicked related content:

[Free Guide] Active Directory Security Best Practices

Almost all common enumeration methods can be executed by an unprivileged user, which makes them extremely hard to detect and block. In this post, I will explain how to use to defend against two types of internal reconnaissance:

- Session enumeration by unprivileged users
- Queries to the MS-SAMR protocol

Types of Reconnaissance

To find information about the network they have penetrated, attackers often enumerate the following types of data:

- **Sessions** — To find out who is logged on where
- **Users** — To understand all users in the domain, ideally with their group membership
- **Groups** — To see all groups in the domain, ideally with their membership
- **Active Directory access control lists (ACLs)** — To understand which security principals (such as users and groups) can access which resources
- **Local group membership** — To see who has access rights on the machine (especially who has administrative rights)

Blocking Session Enumeration with NetCease

This blog post focuses on session enumeration, which enables an adversary to determine where user and service accounts are logged in. That information helps them prioritize which hosts to attempt to compromise first — such as hosts that have an administrator logged in.

Note: The default permissions in Windows 10 have been changed to stop attackers from performing session enumeration; however, it is still worth checking.

NetCease is a short PowerShell script that helps prevent session enumeration by changing the Registry key that controls the NetSessionEnum method permissions. (The reason why this is completed using a script rather than manually is because the key, SrvsvcSessionInfo, is editable only in a reg binary value.) Here is the path to the SrvsvcSessionInfo key:

Path: HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/LanmanServer/DefaultSecurity

The default value of the SrvsvcSessionInfo registry key is the ACL that allows use of the NetSessionEnum method. This is assigned to the following:

- Member of Administrators
- Member of Server Operators
- Member of Power Users
- Authenticated Users

The Authenticated Users permission enables attackers to perform session reconnaissance using any compromised user account.

What the NetCease script does is back up the current registry value and then amend the permissions, so the following ACEs are in the ACL:

- InteractiveSid
- ServiceSid
- BatchSid
- Administrators
- Server Operators
- PowerUsers

Checking the ACL

If you want to view the security descriptor, you can use the following PowerShell snippet, which will show you the ACL:

```
#Registry Key Information
$key = "HKLM:SYSTEMCurrentControlSetServicesLanmanServerDefaultSecurity"
$name = "SrvsvcSessionInfo"

#Get the Registry Key and Value
$Reg_Key = Get-Item -Path $key
$BtyeValue = $reg_Key.GetValue($name, $null)

#Create a CommonSecurityDescriptor Object using the Byte Value
$Security_Descriptor = New-Object -TypeName
System.Security.AccessControl.CommonSecurityDescriptor -ArgumentList $true, $false,
$ByteValue, 0

#Output of the ACL to make it simple to see for document. Use only
$Security_Descriptor.DiscretionaryAcl if you want to see the full ACL!
$Security_Descriptor.DiscretionaryAcl | Select-Object SecurityIdentifier, ACEType | Format-
Table -AutoSize
```

Output before running NetCease:

Output after running NetCease:

Note: Information on well-known SIDs can be found here.

Testing Session Enumeration

An easy way to test whether you have blocked session enumeration by ordinary user accounts is to use the NetSess tool from Joeware.net, but there are plenty of other options, including SharpHound. When doing this, make sure that you are using a user account that is not a member of Administrators, Server Operators or Power Users.

Output before using NetCease

```
Netsess.exe [Computer]
```

Output after using NetCease using an unprivileged account

```
Netsess.exe [Computer]
```

Output after using NetCease with a privileged user account

```
Netsess.exe [Computer]
```

Blocking Reconnaissance using Remote SAM (SAMR)

Attackers can perform reconnaissance using the SAMR protocol, which can remotely query devices but can also query Active Directory. Using SAMR, an attacker without any administrative privileges can find highly privileged groups and users, as well as local users and groups for every system on the network. Tools such as BloodHound can then automatically map this information into attack paths to compromise Active Directory.

Microsoft introduced protections for querying SAMR with Windows 10, and in 2017 added updates for previous operating systems down to Windows 7 and Server 2008 R2 using the RestrictRemoteSAM registry key. This key is a string (REG_SZ) that will contain the SDDL of the security descriptor that protects Remote SAM calls.

In the anniversary edition of Windows 10 (1607) and Windows Server 2016 and later, the default SDDL was changed to allow only local administrators to query Remote SAM.

Below is a table breaking down the requirements, default behavior and protection options for all operating systems:

OS	KB required	Who can query by default	SAMR protection options

Prior to Windows 7 and Server 2008 R2	N/A	Any domain user	None
Windows 7	KB 4012218	Any domain user	Registry Key or Group Policy
Windows Server 2008 R2	KB 4012218	Any domain user	Registry Key or Group Policy
Windows 8.1	KB 4102219	Any domain user	Registry Key or Group Policy
Windows Server 2012	KB 4012220	Any domain user	Registry Key or Group Policy
Windows Server 2012 R2	KB 4012219	Any domain user	Registry Key or Group Policy
Windows 10 1507	KB 4012606	Any domain user	Registry Key or Group Policy
Windows 10 1511	KB 4103198	Any domain user	Registry Key or Group Policy
Windows 10 1607 and later	N/A	Local Administrators	Registry Key or Group Policy
Windows Server 2016 and later	N/A	Local Administrators	Registry Key or Group Policy

There are three ways to set the RestrictRemoteSAM registry key:

- Registry
- Group Policy
- SAMRi10 (Samaritan)

Let's review each of them.

Registry Option

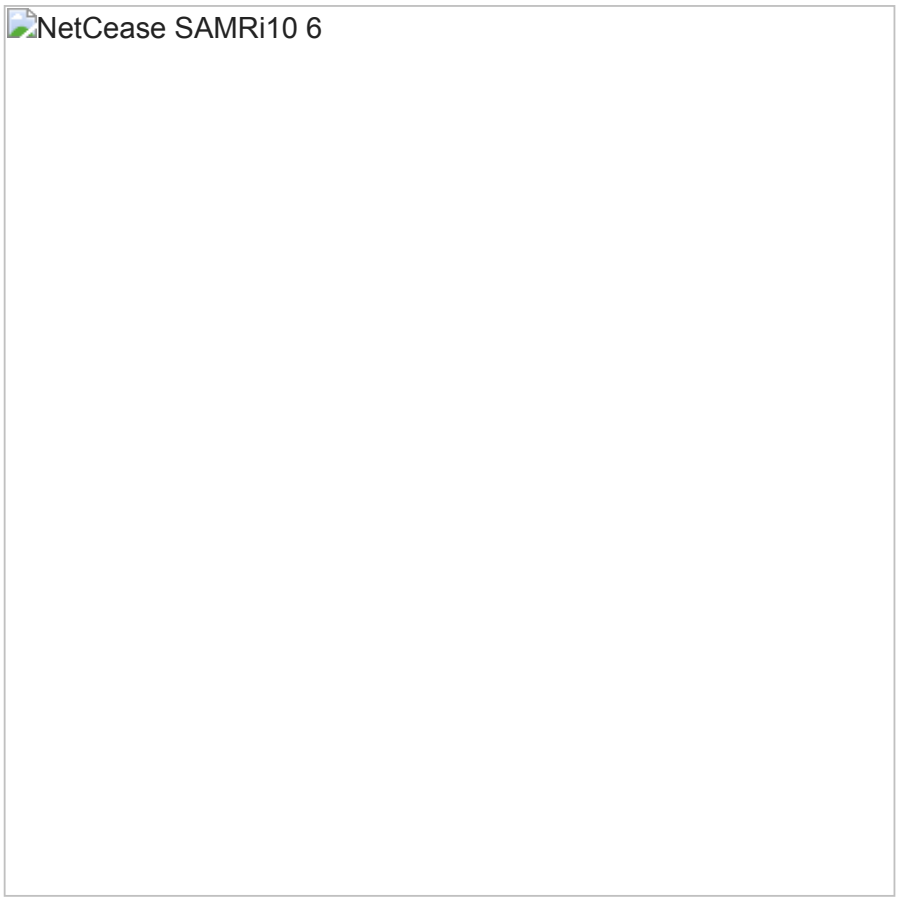
The RestrictRemoteSAM registry key is available for administrators to update as they wish:

Path: HKLM/System/CurrentControlSet/Control/Lsa

Name: RestrictRemoteSAM

Default value (SDDL) in Windows 10: O:SYG:SYD:(A;;RC;;;BA)

Components of the SDDL



As you can see, the default value that Windows 10 sets is SYSTEM for Owner and Primary Group and Read Control for Built-in Administrators.

Checking the SDDL before applying it

To check that the SDDL is correct before applying the change, you can use the ConvertFrom-SDDLString command in PowerShell to convert it to a security descriptor that is easier to read.

Group Policy or Local Security Policy Option

The Group Policy and Local Security Policy settings enable administrators to set this key easily. This can work well for administrators who want to set the same value across every system or multiple groups of systems (e.g., allowing Remote SAM connections for all servers in a specific OU or a certain set of application servers).

The details about the setting are as follows:

Policy name	Network access: Restrict clients allowed to make remote calls to SAM
Location	Computer Configuration|Windows Settings|Security Settings|Local Policies|Security Options
Possible values	· Not defined · Defined, along with the security descriptor for users and groups who are allowed or denied to use SAMRPC to remotely access either the local SAM or Active Directory

SAMRi10 (Samaritan) Option

SAMRi10 is a PowerShell script that offers a key benefit over the previous options: It creates a new local group and delegates access for the group to be able to perform Remote SAM calls. This makes it possible for administrators to control this functionality fully in Group Policy Preferences, or to manually grant it to accounts when required.

The SAMRi10 script does the following:

1. Creates a local group called “Remote SAM Users”
2. Amends the SDDL to include the newly created group:
    - If there is no default SDDL, then grant access to Built-in Administrators.
    - If an SDDL exists, then amend it to include the new ACE for the Remote SAM Users group.

Benefits of using SAMRi10

- Easy to grant granular access for Remote SAM access
- Helps organizations that want to enforce least-privileged access
- Can be used in conjunction with a local group membership Group Policy to grant users access centrally using item-level targeting
- Can be used by a privileged access management (PAM) system to easily grant dynamic (just-in-time) access if an account or process requires this specific permission

Defending against Reconnaissance with Netwrix Solutions

Netwrix StealthAUDIT includes an attack path analyzer that provides admins with insight into their Active Directory ACLs so they can plug any gaps before adversaries exploit them. Netwrix StealthINTERCEPT can monitor LDAP queries and then pass them to Netwrix StealthDEFEND, which can detect multiple reconnaissance scenarios and queries out of the box, including BloodHound, queries for all SPNs, and queries for all accounts with password never expires.

Joe Dibley
Security Researcher at Netwrix and member of the Netwrix Security Research Team. Joe is an expert in Active Directory, Windows, and a wide variety of enterprise software platforms and technologies, Joe researches new security risks, complex attack techniques, and associated mitigations and detections.