Ex090

Joshua Main-Smith

2020-10-21

Contents

Technical Report

Finding: Users can Create Local Admin Accounts

Risk Assessment

Users without admin privileges are able to create local admin account, thereby bypassing the security impediments put into place for the said user by signing in as a user with admin privileges.

Vulnerability Description

The vulnerability is due to a misconfiguration of the system that allows users without admin privileges to create local admin accounts.

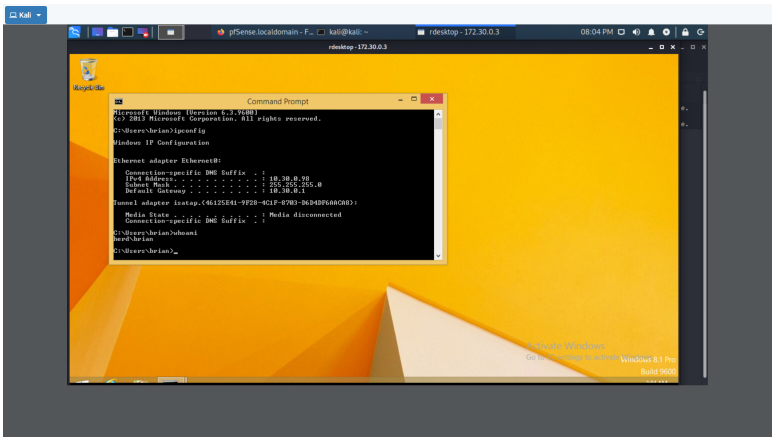
Mitigation or Resolution Strategy

Turn off functionality that allows non-admin users to create local admin accounts.

Attack Narrative

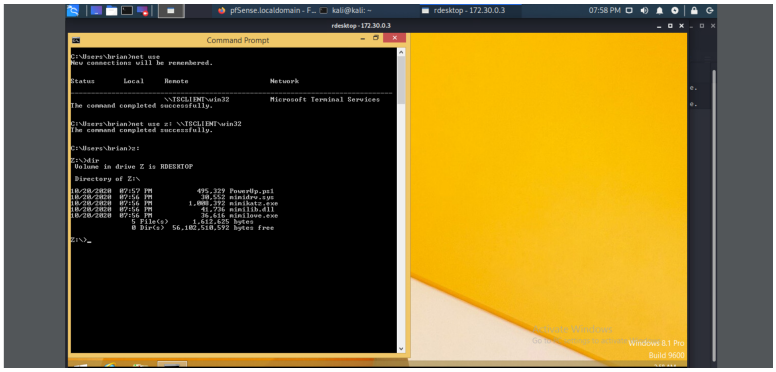
Elevating Privileges

Since we now have access to Brian's account on the herd host, as evidenced by a screenshot of the IP address and a **whoami** command in the image below, we attempted to elevate our privileges using PowerUp and Mimikatz.



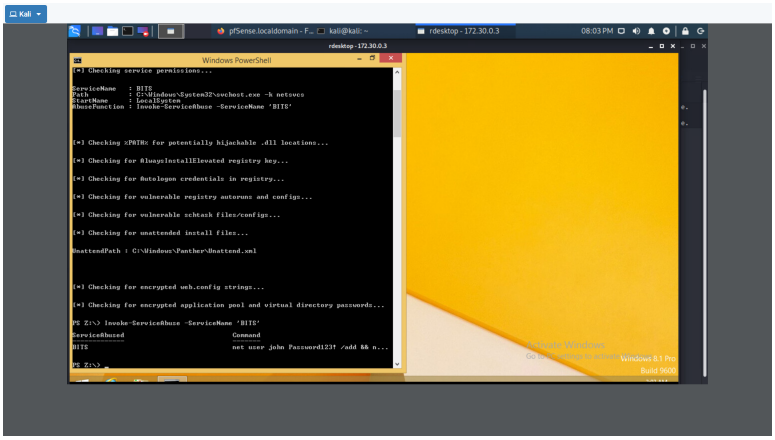
To start, we copied all of our tools to another folder in case the machine has an antivirus tool that will delete our scripts upon execution (**/tmp/foo**). We then connected to herd via remote desktop by issuing the command **rdesktop -g95% -r disk:win32=/tmp/foo 172.30.0.3**. We then connected to Brian's profile

using the credentials described in EX080. Once there, we mounted the script and executable to the Z: drive by doing **net use z: \\TSCLIENT\win32**



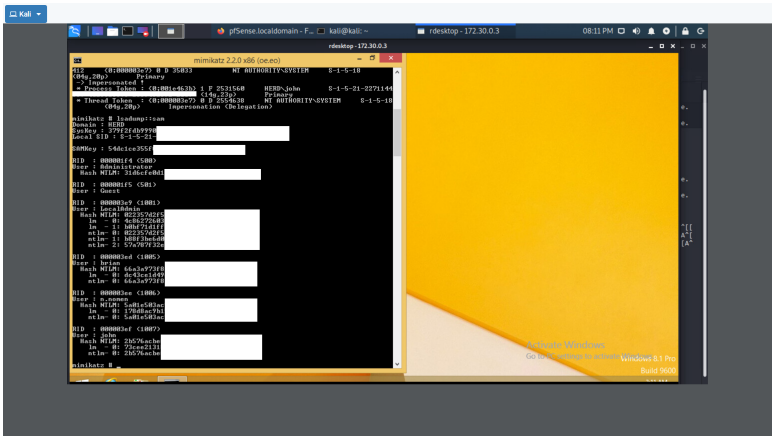
Once mounted, we started Powershell with **Powershell -exec bypass** (to enable scripts to run in Powershell) then imported the Powershell PowerUp script with **Import-Module \\TSClient\win32\PowerUp.ps1**. We can then **Invoke-AllChecks** so that the scripts can search for any misconfigurations on the machine that could help us.

We found that there is a service permission misconfiguration where we were allowed to create a new admin user account as Brian. This abuse was invoked by using the command **Invoke-ServiceAbuse -ServiceName 'BITS'** which created a new user 'john' with a default password of *Password123!*, which has admin permissions.



Once this account was created, we logged out and logged back in as john. We confirmed that we had admin permissions by successfully running Command Prompt as admin. We then proceeded to mount our tools to the Z: drive as described earlier.

Next, we successfully exfiltrated the local user hashes for the herd system by using Mimikatz. To start it, we simply write **mimikatz.exe** into the terminal once we have navigated to our Z: drive. Once our tool has started, we can enable debug mode on our system with **privilege::debug** then elevate our privileges with **token::elevate**. Once these are enabled, we are able to view the stored user hashes with **lsadump::sam**. A redacted screenshot of our findings are located below.



We saved a copy of these hashes locally to our attack box with the goal of cracking these hashes with well known hash-cracking tools.

Keys

We found two keys while looking in user folders. Key011 was located in **C:/Users/Sharon/Documents/MessageFromBigBiss.txt** and key012 was located in **C:/Users/Bogus/Documents/trismegistus.txt**. The value of these keys can be seen in the images provided below.

Zoom Links

October 19, 2020

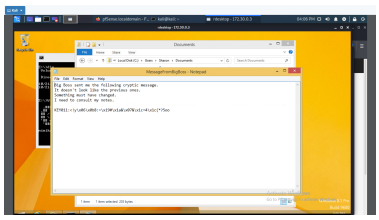


Figure 1: KEY011

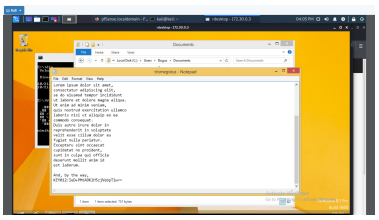


Figure 2: KEY012