



Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

01

Network Topology

02

Red Team: Security Assessment

03

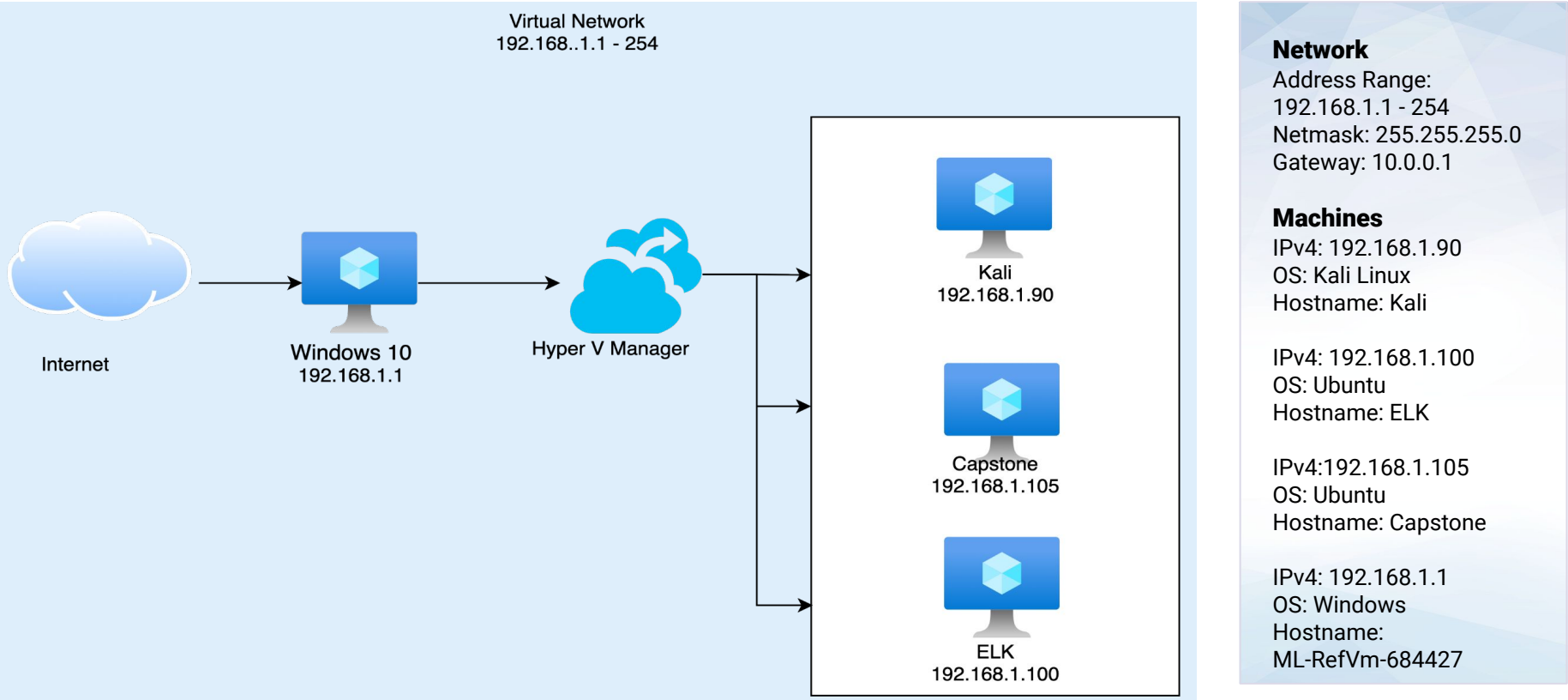
Blue Team: Log Analysis and Attack Characterization

04

Hardening: Proposed Alarms and Mitigation Strategies

Network Topology

Network Topology



The background of the slide is a dark red, almost black, geometric pattern composed of numerous overlapping triangles and polygons, creating a complex, crystalline texture.

Red Team Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Kali Linux Virtual Machine	192.168.1.90	Attacker machine
ELK Ubuntu Virtual Machine	192.168.1.100	ELK monitoring
Capstone Ubuntu Virtual Machine	192.168.1.105	Victim machine
Windows 10 Virtual Machine	192.168.1.1	Host machine. Used to access virtual network which includes Kali, ELK and Capstone virtual machines.

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Weak Password Protocol Brute force attack	Used Hydra to brute force URL. Information found allowed us to guess the username and password. Brute forced using a wordlist	The brute force allowed access to non-public files on the server.
Insecure Hashing Technique MD5 Function Hash John the Ripper	Used john to crack the hash found after brute forcing the URL.	The cracked hash allowed access to the remote file sharing server.
Weak Authentication & Information Accessibility Payload Injection	Created msfvenom payload to execute on the webdav server to enable access via meterpreter shell.	Meterpreter shell allowed escalation of privileges to exploit the entire machine.

Exploitation: Weak Password Protocol

01

Tools & Processes

Nmap
Dirb
Hydra

02

Achievements

Used nmap to find the vulnerable web server. Once found, we used dirb to search the website for information and obtained knowledge for potential usernames to use in a brute-force attack. Used Hydra and a wordlist (rockyou.txt) to brute force the login credentials

03

```
1454555 [info] 1) (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 2021-10-13 1
6:58:01
```


Exploitation: Insecure Hashing Technique

01

Tools & Processes

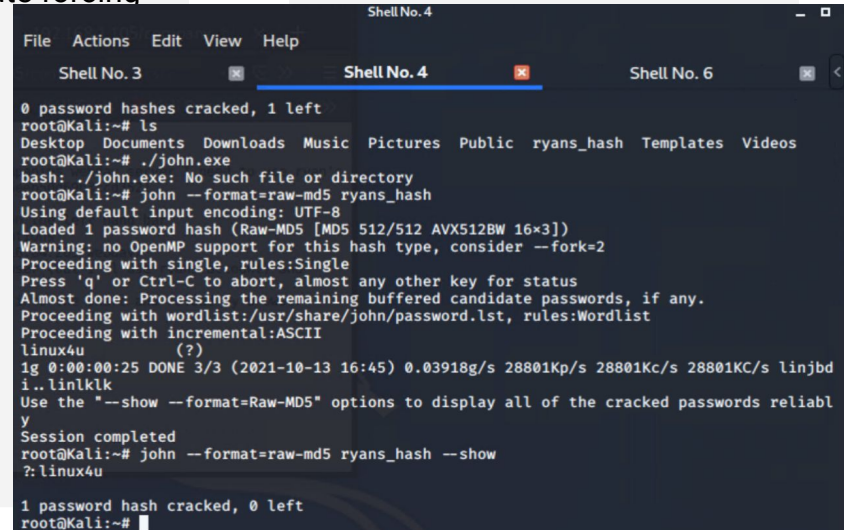
John the Ripper

02

Achievements

Used john.exe to crack the hash found after brute forcing the backend.

03



```
Shell No. 4
File Actions Edit View Help
Shell No. 3 Shell No. 4 Shell No. 6
0 password hashes cracked, 1 left
root@Kali:~# ls
Desktop Documents Downloads Music Pictures Public ryans_hash Templates Videos
root@Kali:~# ./john.exe
bash: ./john.exe: No such file or directory
root@Kali:~# john --format=raw-md5 ryans_hash
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 512/512 AVX512BW 16x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
linux4u (?)
1g 0:00:00:25 DONE 3/3 (2021-10-13 16:45) 0.03918g/s 28801Kp/s 28801Kc/s 28801Kc/s linjbd
i..linlklk
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed
root@Kali:~# john --format=raw-md5 ryans_hash --show
?:linux4u

1 password hash cracked, 0 left
root@Kali:~#
```

Exploitation: Weak Authentication & Information Accessibility

01

Tools & Processes

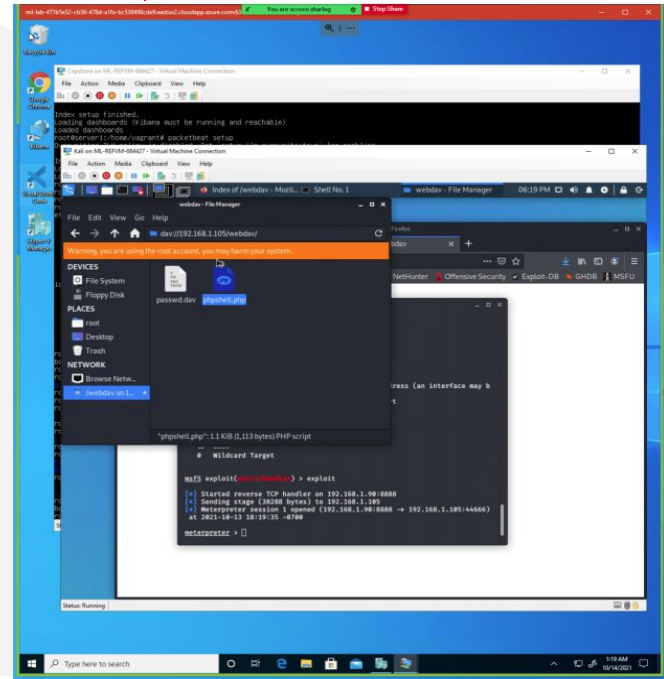
Msfvenom
Metasploit
meterpreter

02

Achievements

Created msfvenom payload to execute on the webdav server to enable access via meterpreter shell.

03





Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan

- The port scan occurred on October 13, 2021 at 20:55
- 8 packets were sent from 192.168.1.90



- User_Agent.Original field was labeled as Mozilla nmap, indicating this was a port scan



Time ▾

user_agent.original

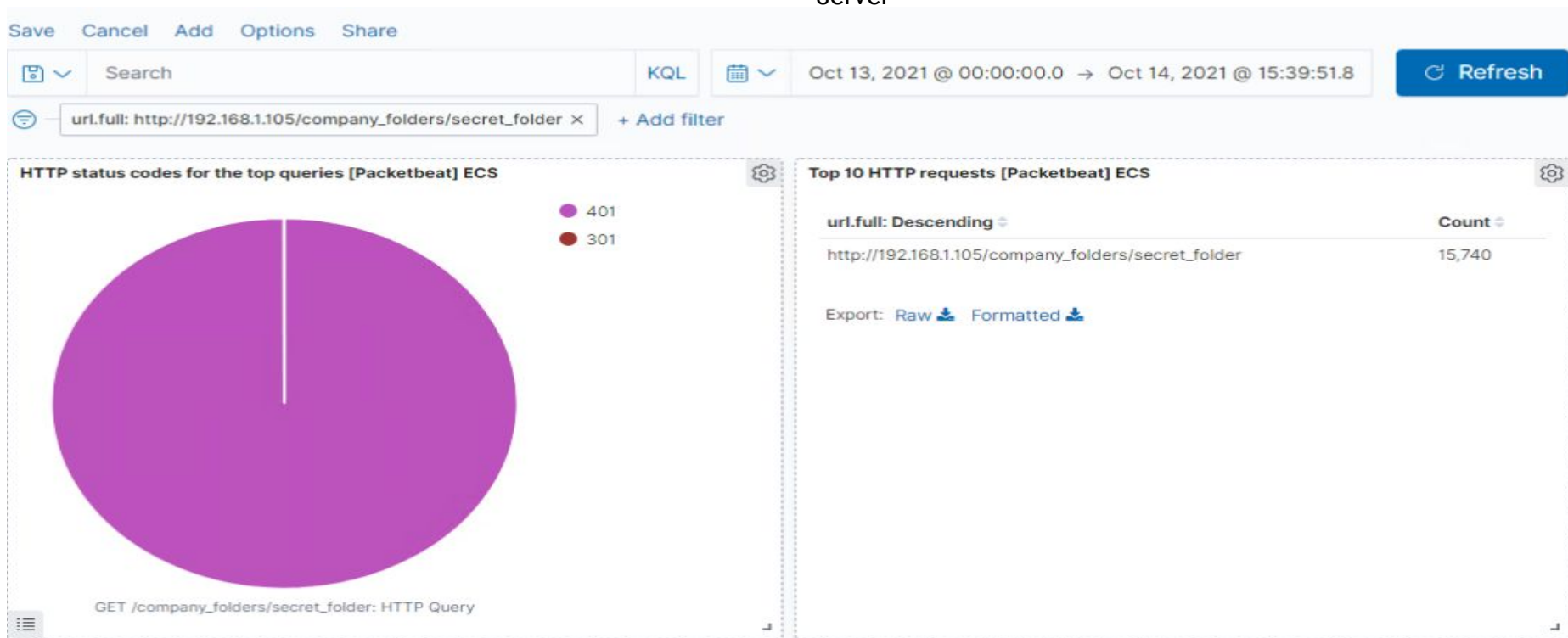
>	Oct 13, 2021 @ 23:03:01.412	Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
>	Oct 13, 2021 @ 23:03:01.412	Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
>	Oct 13, 2021 @ 23:03:01.407	Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
>	Oct 13, 2021 @ 23:03:01.407	Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
>	Oct 13, 2021 @ 23:03:01.396	Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
>	Oct 13, 2021 @ 23:03:01.396	Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
>	Oct 13, 2021 @ 23:03:01.390	Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
>	Oct 13, 2021 @ 23:03:01.390	Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)

Analysis: Finding the Request for the Hidden Directory

- The requests occurred on October 13, 2021 between 8pm and 11pm and 15,740 requests were made.



- Files requested were webdav and the Company Secret Folder, which contained instructions for how to access the webdav server



Analysis: Uncovering the Brute Force Attack

- 286,908 requests were made in the attack.
- 286,907 were made before the attacker



```
> Oct 14, 2021 @ 05:23:22.520 user_agent.original: Mozilla/4.0 (Hydra) @timestamp: Oct 14, 2021 @ 05:23:22.520 query: GET
/company_folders/secret_folder network.type: ipv4 network.transport: tcp network.protocol: http
network.direction: outbound network.community_id: 1:atAhJDr/B+u6s6bzFaMdDCNF33A= network.bytes: 163B
destination.ip: 192.168.1.105 destination.port: 80 status: Error server.ip: 192.168.1.105 server.port: 80
host.name: Kali method: get http.request.headers.content-length: 0 http.request.method: get http.request.bytes: 163B

> Oct 14, 2021 @ 05:23:22.520 user_agent.original: Mozilla/4.0 (Hydra) @timestamp: Oct 14, 2021 @ 05:23:22.520 error.message: Unmatched request
host.name: Kali url.scheme: http url.domain: 192.168.1.105 url.path: /company_folders/secret_folder
url.full: http://192.168.1.105/company_folders/secret_folder client.ip: 192.168.1.90 client.port: 51954
client.bytes: 163B type: http network.transport: tcp network.protocol: http network.direction: outbound
network.community_id: 1:FPgM5+f3LnNWJkIUoTvAYDzaP+o= network.bytes: 163B network.type: ipv4 agent.version: 7.8.0

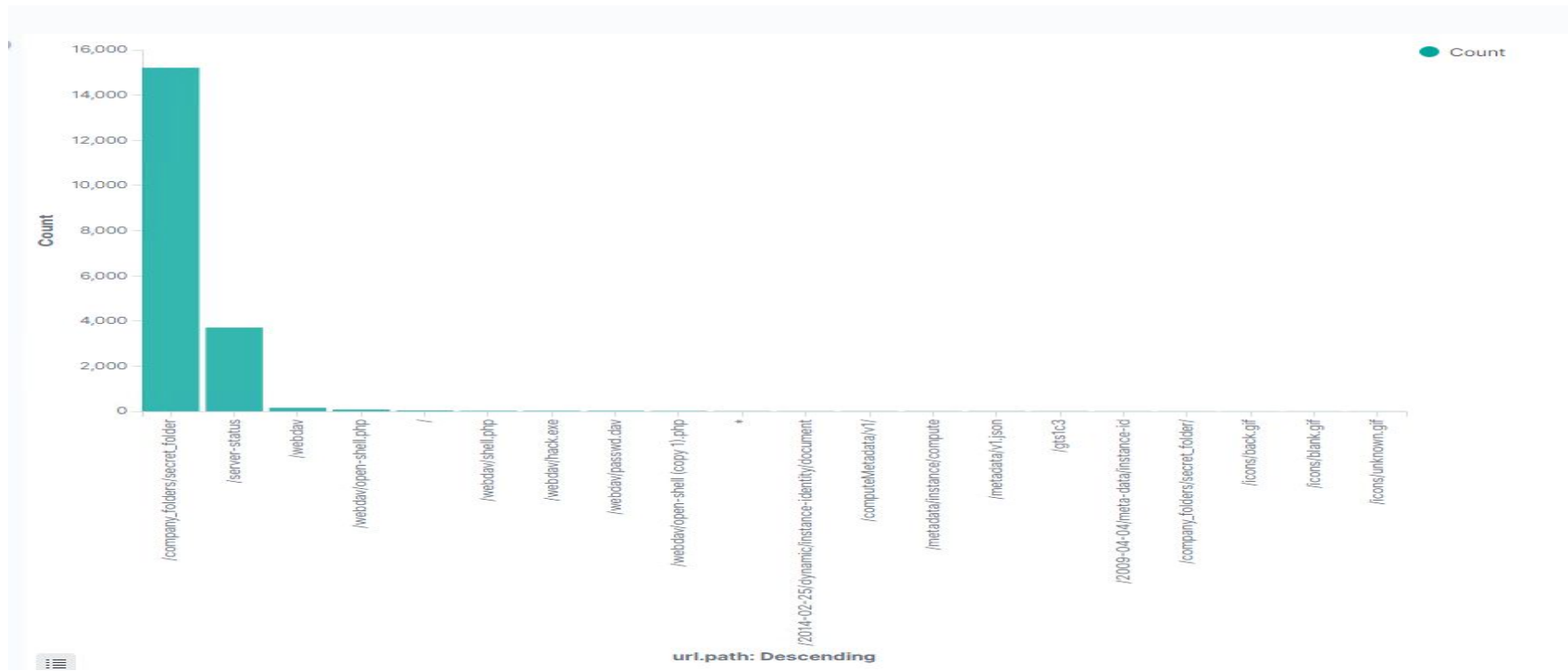
> Oct 14, 2021 @ 05:23:22.520 user_agent.original: Mozilla/4.0 (Hydra) @timestamp: Oct 14, 2021 @ 05:23:22.520 query: GET
/company_folders/secret_folder status: Error url.domain: 192.168.1.105 url.path: /company_folders/secret_folder
url.full: http://192.168.1.105/company_folders/secret_folder url.scheme: http server.port: 80 server.ip: 192.168.1.105
http.request.method: get http.request.bytes: 163B http.request.headers.content-length: 0 http.version: 1.1
destination.ip: 192.168.1.105 destination.port: 80 client.ip: 192.168.1.90 client.port: 51960 client.bytes: 163B
```

Analysis: Finding the WebDAV Connection

- 105,487 requests were made to this directory.



- Some of the files requested were passwd.dav server-status, shell.php





Blue Team

Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

What kind of alarm can be set to detect future port scans?

Set a threshold for number of requests to automatically block IP addresses that exceed the threshold.

What threshold would you set to activate this alarm?

More than 100 requests within 1 minute.

System Hardening

What configurations can be set on the host to mitigate port scans?

Block incoming and outgoing ports except 80 and 443. Block IP addresses after exceeding the threshold for requests. (This is also the solution).

Mitigation: Finding the Request for the Hidden Directory

Alarm

What kind of alarm can be set to detect future unauthorized access?

Set parameters to alert for influx of requests and status response phrases such as “Unauthorized”.

What threshold would you set to activate this alarm?

Any number greater than the total number of users who have access to the hidden directory.

System Hardening

What configuration can be set on the host to block unwanted access?

Remove the folder from public domain and set two-factor authentication

Describe the solution. If possible, provide required command lines.

Two-factor authentication. Deny certain IP addresses, as needed.

Mitigation: Preventing Brute Force Attacks

Alarm

What kind of alarm can be set to detect future brute force attacks?

Send an email to system administrators whenever there is an influx of HTTP requests or error codes from one or more IP addresses.

What threshold would you set to activate this alarm?

Any number of requests that exceeds average daily amount of HTTP traffic for the network.

System Hardening

What configuration can be set on the host to block brute force attacks?

Password complexity and the use of CAPTCHA to block automated brute-force attempts.

Describe the solution. If possible, provide the required command line(s).

Password complexity, two-factor authentication, and CAPTCHA.

Mitigation: Detecting the WebDAV Connection

Alarm

What kind of alarm can be set to detect future access to this directory?

Send an email to system administrators for any large number of HTTP requests sent to this directory. A number greater than the amount of users who have access to the webDAV directory.

What threshold would you set to activate this alarm?

A number higher than the amount of users who have access to this directory.

System Hardening

What configuration can be set on the host to control access?

Whitelist IP addresses

Describe the solution. If possible, provide the required command line(s).

Whitelist only authorized IP addresses and block all other IPs.

Mitigation: Identifying Reverse Shell Uploads

Alarm

What kind of alarm can be set to detect future file uploads?

Set up PUT request alerts to web server folders that will then email system administrators whenever files are uploaded. Setup files to be read for malicious code such as exe and php files.

What threshold would you set to activate this alarm?

Any files from unauthorized IP addresses or any new php or exe files.

System Hardening

What configuration can be set on the host to block file uploads?

Use IDS or anti-malware systems to detect and potentially block uploads. Verify file type before upload.

*The
End*