

# **Brute-Force Attack Incident Response:**

***A Playbook, Investigation, Report and  
Guide***

**Prepared for:** SOC Analysts, Managers and  
Recruiters

**Prepared by:** [Yasif Farook](#)

**Date of Creation:** 08/03/2025

## Table of Contents

1. <b>Executive Summary</b> .....	3
2. <b>The Attack</b> .....	3
○ 2.1 Simulating a Brute-Force Attack using Hydra .....	3
○ 2.2 Identifying the Security Breach .....	5
○ 2.3 Immediate Protection Measures .....	6
○ 2.4. Detecting the Successful Password Crack.....	6
3. <b>The Response</b> .....	8
○ 3.1 Investigating the Attack with Splunk .....	8
○ 3.2 Use Splunk's Filtered Log Searches .....	10
○ 3.3 Analysis Results .....	11
○ 3.4 Attack Pattern Analysis .....	11
4. <b>The Recovery</b> .....	13
○ 4.1 Constructing Reports .....	13
○ 4.2 Log Reports .....	15
○ 4.3 Security Enhancements & Remediation .....	17
○ 4.4 Recovery Summary .....	19
5. <b>Conclusion</b> .....	19
6. <b>References</b> .....	20
7. <b>Appendix</b> .....	20

## 1.0. Executive Summary:

This document provides a detailed technical analysis of a simulated brute-force attack conducted on an SSH service as part of a penetration testing exercise. The goal of this project was to evaluate system resilience, identify security vulnerabilities, and develop robust mitigation strategies to fortify defenses against real-world cyber threats.

This report follows the NIST Cybersecurity Framework, detailing each phase of incident response: **Identification**, **Protection**, **Detection**, **Response**, and **Recovery**. The insights gained from this simulation serve not only to improve the security posture of the tested system but also to act as a playbook for security professionals responding to brute-force attacks in real-world environments. The report offers practical defensive strategies, logging enhancements, and response techniques that can be adapted across various IT infrastructures.

By simulating an attack, security weaknesses were identified in authentication mechanisms, logging policies, and access control configurations. The countermeasures outlined here emphasise proactive security measures, real-time detection strategies, and best practices for threat mitigation.

Tools & Technologies Used:

- **Kali Linux 2024.4** – Attacker Environment
- **Hydra** – Brute-force attack tool
- **Nmap** – Network scanning tool
- **Splunk** – Security Information and Event Management (SIEM) platform
- **Linux SSH Service** – Targeted system component

## 2.0. The Attack:

### 2.1. Simulating a Brute-Force Attack using Hydra:

#### 1. Identifying the Target

- local IP address of the host as 127.0.0.1
- SSH Service: Initially disabled; manually enabled for controlled testing.

```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 00:0c:29:36:38:36 txqueuelen 1000 (Ethernet)
    RX packets 589 bytes 37648 (36.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 82810 bytes 92514693 (88.2 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 82810 bytes 92514693 (88.2 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali㉿kali)-[~]
$
```

Check if SSH is running (it was disabled) and starting it:

```
(kali㉿kali)-[~]
$ sudo systemctl status ssh
o ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:sshd(8)
           man:sshd_config(5)

(kali㉿kali)-[~]
$ sudo systemctl start ssh

(kali㉿kali)-[~]
$
```

Executing the Attack:

```
(kali㉿kali)-[~]
$ hydra -t 1 -W 5 -l kali -P /tmp/rockyou_limited.txt ssh://127.0.0.1 -V > hydra_attack.log
```

Monitoring the attack using `tail -f hydra_attack.log`:

```
(kali@kali)-[~]
$ tail -f hydra_attack.log
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "123456789" - 3 of 500 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "password" - 4 of 500 [child 3] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "iloveyou" - 5 of 500 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "princess" - 6 of 500 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "1234567" - 7 of 500 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "rockyou" - 8 of 500 [child 3] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "12345678" - 9 of 500 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "abc123" - 10 of 500 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "nicole" - 11 of 500 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "daniel" - 12 of 500 [child 3] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "babygirl" - 13 of 500 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "monkey" - 14 of 500 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "lovely" - 15 of 500 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "jessica" - 16 of 500 [child 3] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "654321" - 17 of 500 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "michael" - 18 of 500 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "ashley" - 19 of 500 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "qwerty" - 20 of 500 [child 3] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "111111" - 21 of 500 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "iloveu" - 22 of 500 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "000000" - 23 of 500 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "michelle" - 24 of 500 [child 3] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "tigger" - 25 of 500 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "sunshine" - 26 of 500 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "chocolate" - 27 of 500 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "password1" - 28 of 500 [child 3] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "soccer" - 29 of 500 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "anthony" - 30 of 500 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "friends" - 31 of 500 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "butterfly" - 32 of 500 [child 3] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "purple" - 33 of 500 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "angel" - 34 of 500 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "jordan" - 35 of 500 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "liverpool" - 36 of 500 [child 3] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "justin" - 37 of 500 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "loveme" - 38 of 500 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "fuckyou" - 39 of 500 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "123123" - 40 of 500 [child 3] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "football" - 41 of 500 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "secret" - 42 of 500 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "andrea" - 43 of 500 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "carlos" - 44 of 500 [child 3] (0/0)
```

- Password successfully cracked on 500th attempt:

```
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "kali" - 500 of 500 [child 0] (0/0)
```

## 2.2. Identifying the Security Breach:

- System logs recorded multiple failed authentication attempts, indicating brute-force activity.
- Hydra's output log confirmed a successful password compromise:

```
[22][ssh] host: 127.0.0.1  login: kali  password: kali
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-03-07 08:29:30
```

## 2.3. Immediate Protection Measures:

Locked the compromised user account 'kali', to prevent further unauthorised access:

```
(kali㉿kali)-[~]  
$ sudo passwd -l kali  
[sudo] password for kali:  
passwd: password changed.
```

- Restarted **SSH service** to terminate all active sessions:

```
(kali㉿kali)-[~]  
$ sudo systemctl restart ssh
```

## 2.4. Detecting the Successful Password Crack:

In order to begin the detection phase, the log from the brute-force attack was saved to *hydra\_attack.log*. For official documentation, output was saved to Documents for further use:

```
(kali㉿kali)-[~]  
$ cp hydra_attack.log ~/Documents/hydra_attack_report.txt
```

- Running the scan to find open ports (after the attack has been launched):

```
(kali㉿kali)-[~]  
$ nmap -sV 127.0.0.1  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-07 08:51 EST  
Nmap scan report for localhost (127.0.0.1)  
Host is up (0.0000030s latency).  
Not shown: 999 closed tcp ports (reset)  
PORT      STATE SERVICE VERSION  
22/tcp    open  ssh      OpenSSH 9.9p1 Debian 3 (protocol 2.0)  
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 0.38 seconds
```

- Running (a more detailed scan):

```
(kali㉿kali)-[~]
$ nmap -A -T4 127.0.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-07 08:52 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000064s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.9p1 Debian 3 (protocol 2.0)
| ssh-hostkey:
|_  256 93:a6:6d:63:27:70:94:04:c5:f5:da:60:2c:5e:a5:e8 (ECDSA)
|_  256 90:3e:d5:e4:64:58:fe:f0:64:f7:42:22:45:f8:38:e7 (ED25519)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.63 seconds
```

- Saving scan results to a file for further analysis:

```
(kali㉿kali)-[~]
$ nmap -A -T4 127.0.0.1 -oN ~/Documents/nmap_scan_results.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-07 08:54 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000078s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.9p1 Debian 3 (protocol 2.0)
| ssh-hostkey:
|_  256 93:a6:6d:63:27:70:94:04:c5:f5:da:60:2c:5e:a5:e8 (ECDSA)
|_  256 90:3e:d5:e4:64:58:fe:f0:64:f7:42:22:45:f8:38:e7 (ED25519)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.10 seconds
```

## 3.0. The Response:

### 3.1. Investigating the Attack with Splunk:

#### 3.1.1 Open Splunk:

Begin responding to this cybersecurity risk, by opening [Splunk](#), a Security Information and Event Management (SIEM) solution:

```
(kali㉿kali)-[~]
└─$ sudo /opt/splunk/bin/splunk start
[sudo] password for kali: 
splunkd 7658 was not running.
Stopping splunk helpers ...
Done.
Stopped helpers.
Removing stale pid file... done.

Hello, Administrator

Splunk> Be an IT superhero. Go home early.

Checking prerequisites ...
  Checking http port [8000]: open
  Checking mgmt port [8089]: open
  Checking appserver port [127.0.0.1:8065]: open
  Checking kvstore port [8191]: open
  Checking configuration... Done.
  Checking critical directories ... Done
  Checking indexes ...
    Validated: _audit _configtracker _dsappevent _dsclient _dsphonehome _internal _introspect
  Done
  Checking filesystem compatibility... Done
  Checking conf files for problems... Done
  Checking default conf files for edits... Done
  Validating installed files against hashes from '/opt/splunk/splunk-9.4.1-e3bdab203ac8-linux-amd64'
  All installed files intact.
  Done
All preliminary checks passed.

Starting splunk server daemon (splunkd)...
PYTHONHTTPSVERIFY is set to 0 in splunk-launch.conf disabling certificate validation for the httplib and
Done

Waiting for web server at https://127.0.0.1:8000 to be available.....WARNING: Server Certificate
. Done

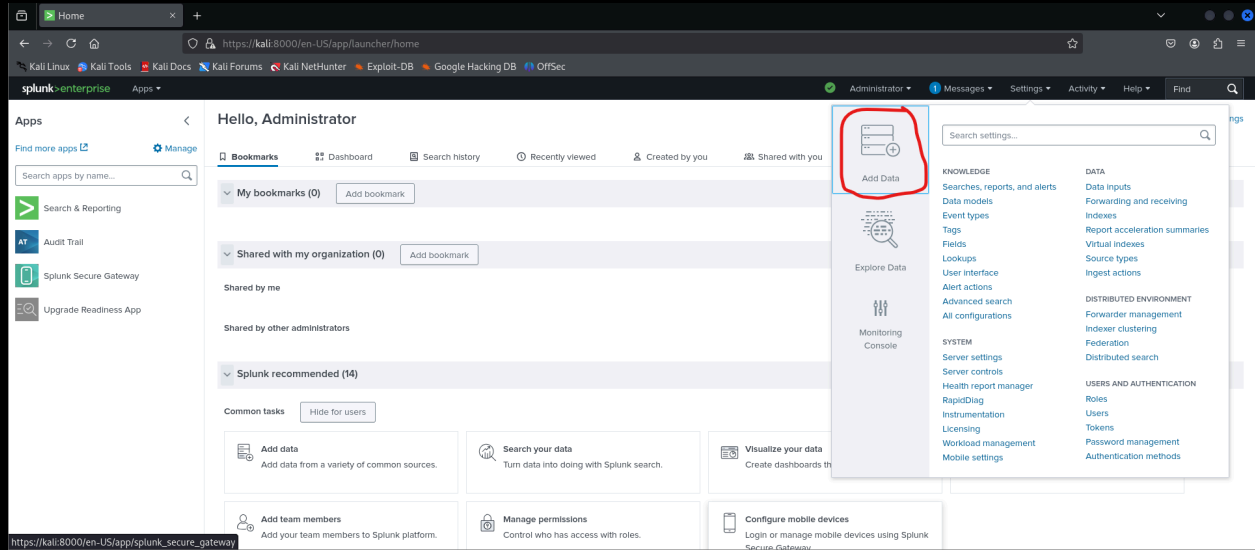
If you get stuck, we're here to help.
Look for answers here: http://docs.splunk.com

The Splunk web interface is at https://kali:8000
```



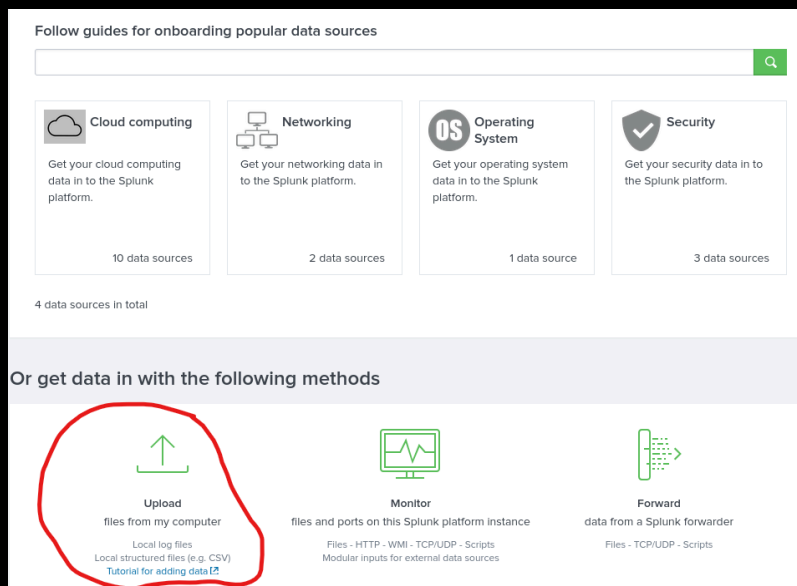
## 3.1.2 Upload Scan Results onto Splunk:

### 3.1.2.1 Add Data under Settings:



### 3.1.2.2 Upload from Local Files:

Both scan results and the Hydra attack log was saved under `/home/kali/Documents`





### 3.3 Analysis Results:

- port (22) was open at the time of attack
- Hydra successfully brute-forced the password at 7:58:45 AM on March 7, 2025.
- Real-time alerts were absent, exposing a logging and alerting deficiency.

```
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000078s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.9p1 Debian 3 (protocol 2.0)
| ssh-hostkey:
|   256 93:a6:6d:63:27:70:94:04:c5:f5:da:60:2c:5e:a5:e8 (ECDSA)
|_  256 90:3e:d5:e4:64:58:fe:f0:64:f7:42:22:45:f8:38:e7 (ED25519)
```

### 3.4 Attack Pattern Analysis:

- **Attack Rate:** 16.23 attempts per minute
- **Insufficient Logging:** System did not log permission-denied messages, reducing visibility of brute-force attempts.
- **Detection Gap:** No automatic SIEM alerting was in place to flag the attack

Logs generated from the hydra brute-force attack:

New Search

index\*\* source="hydra\_attack\_report.txt"

All time

✓ 6 events (before 3/7/25 10:40:26.000 PM) No Event Sampling

Job

Verbose Mode

Events (6)

Patterns

Statistics

Visualization

Timeline format

Zoom Out

Zoom to Selection

Deselect

1 hour per column

Format

Show: 50 Per Page

View: List

< Hide Fields

All Fields

SELECTED FIELDS

a host 1

a source 1

a sourcetype 1

INTERESTING FIELDS

# date\_hour 2

# date\_mday 1

# date\_minute 2

# date\_month 1

# date\_second 2

# date\_wday 1

# date\_year 1

# date\_zone 1

a index 1

# linecount 6

a punct 4

a splunk\_server 1

# timeendpos 4

Time

Event

> 3/7/25 10:08:29.000 PM

SSH Brute-Force Attack Simulation using Hydra  
Analysis of SSH Brute-Force Attack Using Hydra:  
Overview  
This project aimed to simulate an SSH brute-force attack using Hydra and a limited subset of the RockYou password list. The attack successfully identified the correct password, demonstrating the vulnerability of weak SSH credentials.  
Findings  
Show all 25 lines  
host = kali | source = hydra\_attack\_report.txt | sourcetype = hydra\_scan

> 3/7/25 8:29:30.000 AM

Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-03-07 08:29:30  
host = kali | source = hydra\_attack\_report.txt | sourcetype = hydra\_scan

> 3/7/25 7:58:45.000 AM

[ATTEMPT] target 127.0.0.1 - login "kali" - pass "12345678910" - 433 of 500 [child 0] (0/0)  
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "leonardo" - 434 of 500 [child 0] (0/0)  
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "jayjay" - 435 of 500 [child 0] (0/0)  
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "liliana" - 436 of 500 [child 0] (0/0)  
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "dexter" - 437 of 500 [child 0] (0/0)  
Show all 74 lines  
host = kali | source = hydra\_attack\_report.txt | sourcetype = hydra\_scan

Row 3 provided context regarding the successful password-crack. On the 7th of March 2025, at 7:58:45 AM, the machine's brute-force attack was successful. Upon detailed inspection, the correct password "kali" was guessed, on the 500th attempt. The status message a few lines above suggested the machine was attempting passwords at a rate of 16.23 tries per minute.

```
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "kali" - 500 of 500 [child 0] (0/0)
[22][ssh] host: 127.0.0.1 login: kali password: kali
1 of 1 target successfully completed, 1 valid password found
Collapse
host = kali | source = hydra_attack_report.txt | sourcetype = hydra_scan
```

Unsuccessful login attempts did not provide any "permission denied" or "unsuccessful" message. Although the system printed a message confirming the successful guess of the password "kali", once the attempt was made.

This can be noted when filtering for *index=\* source="hydra\_attack\_report.txt" "password found"*

### New Search

index=\* source="hydra\_attack\_report.txt" "password found"

✓ 1 event (before 3/7/25 10:55:13.000 PM) No Event Sampling ▾

Events (1) Patterns Statistics Visualization

Timeline format ▾ — Zoom Out + Zoom to Selection × Deselect

< Hide Fields

≡ All Fields

SELECTED FIELDS

a host 1

a source 1

a sourcetype 1

INTERESTING FIELDS

# date\_hour 1

# date\_mday 1

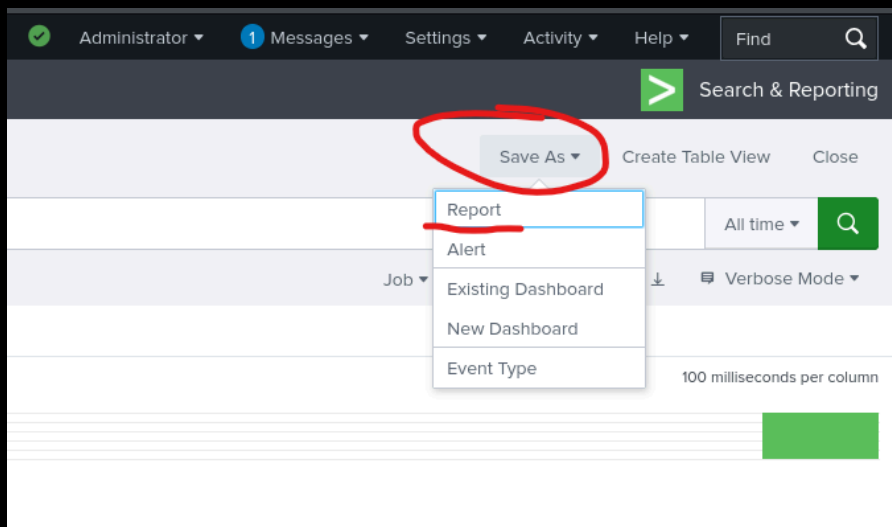
i	Time	Event
>	3/7/25 7:58:45.000 AM	[ATTEMPT] target 127.0.0.1 - login "kali" - pass "12345678910" - 433 of 500 [child 0] (0/0) [ATTEMPT] target 127.0.0.1 - login "kali" - pass "leonardo" - 434 of 500 [child 0] (0/0) ... 69 lines omitted ... [ATTEMPT] target 127.0.0.1 - login "kali" - pass "kali" - 500 of 500 [child 0] (0/0) [22][ssh] host: 127.0.0.1 login: kali password: kali 1 of 1 target successfully completed, 1 valid password found <a href="#">Show all 74 lines</a> host = kali   source = hydra_attack_report.txt   sourcetype = hydra_scan

## 4.0. The Recovery:

### 4.1 Constructing Reports:

For cybersecurity, it is best practice to save all Splunk query results as reports for later use. Because cyber threats continue to grow in sophistication and effectiveness, cyber incident reporting is not only important but also necessary for other organizations to learn from and prevent making the same mistakes (*Upguard, 2025*). Splunk has a feature to generate reports for each search.

#### 4.1.1 Export Each Search Result by Clicking "Save As → Report":



#### 4.1.2 Exporting Nmap Scan Log:

To stay organised, it is essential that reports are named relevant to the incident, so they can be accessed quickly. In order to maintain confidentiality, reports should be stored in locations where only privileged users have access. Good practice would be to include an administrative password before accessing reports. To maintain integrity, reports should only be made once the user can confirm Splunk has correctly parsed all log info, with correct file names and times.<sup>1</sup> Backups of all reports should also be made, to maintain availability of the information in case of a system malfunction or data theft, which would result in file loss.

#### 4.1.2 Exporting Nmap Scan Log:

The report for the *nmap\_scan\_results.txt* has been named *nmap\_scan\_analysis*.

**Save As Report** [X]

Title:

Description:

Content: ☒ Events

Time Range Picker: ☒ Yes ☐ No

#### 4.1.2 Exporting Hydra Attack Log:

The report for the *hydra\_attack\_report.txt* has been named *hydra\_attack\_analysis*.

**Save As Report** [X]

Title:

Description:

Content: ☒ Events

Time Range Picker: ☒ Yes ☐ No

Another method to stay organised on various cybersecurity incidents, is by creating dashboards on Splunk. Dashboards offer an interactive, visual and easy-to-understand medium of presenting a cybersecurity incident.

Save Panel to Existing Dashboard

Select an Existing Dashboard

Sort: Title (A - Z) ↓

Search By Title

Hydra Attack Compromising Kali's System

Integrity Check of Installed Files

Job Details Dashboard

jQuery Upgrade

Orphaned Scheduled Searches, Reports, and Alerts

Panel Title

Nmap Scan Report

Visualization Type

Events

Advanced Panel Settings

Cancel

Save to Dashboard

## 4.2.0 Log Reports

### 4.2.1 Nmap Scan Report:

nmap\_scan\_analysis

Based on the nmap scan run on the local network 127.0.0.1, which was commenced after the brute-force attack that successfully guessed the password to the user kali's system, thereby creating a cybersecurity risk

All time

2 events (before 3/8/25 12:00:13.000 AM)

20 per page

i	Time	Event
>	3/7/25 8:54:39.000 AM	# Nmap done at Fri Mar 7 08:54:39 2025 -- 1 IP address (1 host up) scanned in 2.10 seconds host = kali : source = nmap_scan_results.txt : sourcetype = nmap
>	3/7/25 8:54:37.000 AM	# Nmap 7.94SVN scan initiated Fri Mar 7 08:54:37 2025 as: /usr/lib/nmap/nmap --privileged -A -T4 -oN /home/kali/Documents/nmap_scan_results.txt 127.0.0.1 Nmap scan report for localhost (127.0.0.1) Host is up (0.000078s latency). Not shown: 999 closed tcp ports (reset) PORT      STATE SERVICE VERSION <a href="#">Show all 16 lines</a> host = kali : source = nmap_scan_results.txt : sourcetype = nmap

## 4.2.2 Hydra Scan Report:

hydra\_attack\_analysis

Edit

More Info

Add to Dashboard

Based on the simulated brute-force attack that was launched on the local network 1270.0.1. The purpose behind the attack was to point out vulnerabilities in the system of the user kali. The prompt successfully guessed the password to the user kali's system, thereby creating a cybersecurity risk.

All time

6 events (before 3/8/25 12:02:04.000 AM)

20 per page

i

Time

Event

>

3/7/25  
10:08:29.000 PM

SSH Brute-Force Attack Simulation using Hydra  
Analysis of SSH Brute-Force Attack Using Hydra:  
Overview  
This project aimed to simulate an SSH brute-force attack using Hydra and a limited subset of the RockYou password list. The attack successfully identified the correct password, demonstrating the vulnerability of weak SSH credentials.  
Findings  
Show all 25 lines  
host = kali | source = hydra\_attack\_report.txt | sourcetype = hydra\_scan

>

3/7/25  
8:29:30.000 AM

Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-03-07 08:29:30  
host = kali | source = hydra\_attack\_report.txt | sourcetype = hydra\_scan

>

3/7/25  
7:58:45.000 AM

[ATTEMPT] target 127.0.0.1 - login "kali" - pass "12345678910" - 433 of 500 [child 0] (0/0)  
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "leonardo" - 434 of 500 [child 0] (0/0)  
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "jayjay" - 435 of 500 [child 0] (0/0)  
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "liliana" - 436 of 500 [child 0] (0/0)  
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "dexter" - 437 of 500 [child 0] (0/0)  
Show all 74 lines  
host = kali | source = hydra\_attack\_report.txt | sourcetype = hydra\_scan

>

3/7/25  
7:58:45.000 AM

[ATTEMPT] target 127.0.0.1 - login "kali" - pass "portugal" - 301 of 500 [child 0] (0/0)  
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "laura" - 302 of 500 [child 0] (0/0)  
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "777777" - 303 of 500 [child 0] (0/0)  
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "marvin" - 304 of 500 [child 0] (0/0)  
[ATTEMPT] target 127.0.0.1 - login "kali" - pass "edent" - 305 of 500 [child 0] (0/0)

## 4.2.3 Editing Dashboards:

Edit each report added to the dashboard under Configuration, as each report requires:

Configuration

General

Visualization type

Events

Title

Hydra Attack

Description

Based on the simulated brute-force attack that wa:

Data sources

Events 1 search

Events 2 search

Visibility

Position and size

X position

10

Y position

10

Configuration

General

Visualization type

Events

Title

Nmap Scan Report

Description

Based on the nmap scan run on the local network

Data sources

Nmap Scan Report 1 search

Nmap Scan Report 2 search

Visibility

Position and size

X position

0

Y position

630



4.2.4 Final Dashboard:

### Hydra Attack Compromising Kali's System

Global Time Ran...

Last 24 hours

#### Hydra Attack

Based on the simulated brute-force attack that was launched on the local network 127.0.0.1. The purpose behind the attack was to point out vulnerabilities in the system of the user kali. The prompt successfully guessed the password to the user kali's system, thereby cr...

i	Time	Event
>	3/7/2025 10:08:29.000 PM	<div>SSH Brute-Force Attack Simulation using Hydra</div> <div>Analysis of SSH Brute-Force Attack Using Hydra:</div> <div>Overview</div> <p>This project aimed to simulate an SSH brute-force attack using Hydra and a limited subset of the RockYou password list. The attack successfully identified the correct password, demonstrating the vulnerability of weak SSH credentials.</p> <div>Findings</div> <p>The attack was executed against 127.0.0.1 (localhost) with the username kali.</p> <p>A subset of 500 passwords was used to limit execution time.</p> <p>After 499 attempts, the correct password was identified.</p> <p>Initial issues, such as connection errors and SSH rate limiting, were resolved by adjusting Hydra's task settings and ensuring SSH was running properly.</p> <div>Observations</div> <ul style="list-style-type: none"><li>- Connection Issues: The attack was initially slowed down by SSH protections, leading to errors like "Connection reset by peer."</li><li>- Brute-Force Speed: Using -t 4 limited parallel attempts, avoiding detection mechanisms while maintaining efficiency.</li><li>- System Logging: No auth.log file was found, likely due to default logging configurations in Kali Linux.</li></ul> <div>Implications</div> <ul style="list-style-type: none"><li>- Real-world risks: Weak passwords make SSH servers vulnerable to brute-force attacks.</li><li>- Mitigation: Using strong, complex passwords, limiting SSH login attempts, and enabling fail2ban can prevent such attacks.</li><li>- Alternative Defenses: Disabling password authentication and using SSH keys would make brute-force attacks ineffective.</li></ul> <div>Attack Execution Details:</div> <p>To simulate the brute-force attack:</p> <pre>hydra -t 1 -N 5 -l kali -P /tmp/rockyou_limited.txt ssh://127.0.0.1 -V &gt; hydra_attack.log</pre> <p>To monitor progress of the attack:</p> <pre>tail -f hydra_attack.log</pre> <p>Results were logged onto hydra_attack.log. This data was then saved to hydra_attack_report.txt for investigative analysis.</p> <p>Below is the log of the brute-force attack, as displayed in hydra_attack.log:</p>

#### Nmap Scan Report

Based on the nmap scan run on the local network 127.0.0.1, which was commenced after the brute-force attack that successfully guessed the password to the user kali's system, thereby creating a cybersecurity risk.

i	Time	Event
>	3/7/2025 8:54:39.000 AM	# Nmap done at Fri Mar 7 08:54:39 2025 -- 1 IP address (1 host up) scanned in 2.10 seconds
>	3/7/2025 8:54:37.000 AM	<div># Nmap 7.94SVN scan initiated Fri Mar 7 08:54:37 2025 as: /usr/lib/nmap/nmap --privileged -A -T4 -oN /home/kali/Documents/nmap_scan_results.txt 127.0.0.1</div> <div>Nmap scan report for localhost (127.0.0.1)</div> <div>Host is up (0.000078s latency).</div> <div>Not shown: 999 closed tcp ports (reset)</div> <div>PORT      STATE SERVICE VERSION</div> <div>22/tcp    open  ssh      OpenSSH 9.9p1 Debian 3 (protocol 2.0)</div> <div>  ssh-hostkey:</div> <div>  256 93:a6:6d:63:27:70:94:04:c5:f5:da:60:2c:5e:a5:e8 (ECDSA)</div>

Link to [pdf](#) version of Dashboard

4.3 Security Enhancements and Remediation

An effective recovery process for this attack would involve a process that minimises attack vectors that the simulated brute-force attack pointed out. These include:

- Open ports allowing unauthorised access
- Weak passwords
- Lack of limits on login attempts

To address this, SSH was disabled for unnecessary users, and maximum authentication attempts was reduced to 3. Refer below for remediation steps:

1. Hardening SSH Configuration:
  - Modified SSH settings to enforce stricter authentication controls

Using `sudo nano /etc/ssh/sshd_config`, the text file was edited as such:

```
#MaxAuthTries was set to 3 (Limit login attempts to 3)
#PermitRootLogin no (Disable root login)
AllowUsers kali (Restrict SSH access to other unauthorised users)
```

```
# Authentication:
#LoginGraceTime 2m
#PermitRootLogin no
#StrictModes yes
#MaxAuthTries 3
#MaxSessions 10
# ForceCommand cvs server
AllowUsers kali
```

- Restart SSH service to apply changes:

```
(kali㉿kali)-[/etc/system/local]
$ sudo service ssh restart
(kali㉿kali)-[/etc/system/local]
$
```

Be sure to verify user access, by logging in with the user's credentials. This is to ensure access is allowed. Also attempt to log in with any other user account, in order to verify that SSH access is denied.

## Further Recommendations:

### 4.3.1 Firewall Configuration (UFW & iptables):

**Firewalls** - It is worth considering configuring the system's firewall (e.g., ufw or iptables) to further restrict access.

`sudo ufw allow from 192.168.1.100 to any port 22`

```
sudo iptables -A INPUT -p tcp --dport 22 -m limit --limit 3/min -j ACCEPT
```

#### 4.3.2 Monitoring and Logging Enhancements:

- Enabled real-time alerts for failed login attempts:

```
sudo journalctl -u ssh --since "1 hour ago"
```

- Configured Splunk to detect brute-force patterns and generate automated alerts.

#### 4.3.3 Future Security Considerations:

- *Periodic penetration testing* to identify potential vulnerabilities.
- *Multi-Factor Authentication (MFA)* enforcement for SSH access.
- *Automated intrusion detection systems (IDS)* and *honeypots* for real-time attack mitigation.

*Public Key Authentication* - For further security measures, consider using public key authentication and disabling password-based logins by setting `PasswordAuthentication no` in the `sshd_config` file.

## 4.4 Recovery Summary

The recovery process aims to revive the compromised system and affected files and people from a security breach/incident. By following the above process and recommendations, the system's security risk of unauthorised access, which can be exploited by an attacker using the guessed password, can be mitigated. It is essential for the system to uphold practices that support its security posture, in order to maintain standards that uphold the CIA triad.

## 5.0. Conclusion:

This in-depth analysis of a brute-force attack revealed significant security risks associated with exposed SSH services, weak authentication policies, and insufficient logging mechanisms. The countermeasures implemented focused on hardening access control, improving detection capabilities, and enforcing industry-standard security best practices.

Moving forward, proactive security monitoring, continuous threat modeling, and automated defense mechanisms will be key to mitigating brute-force and other authentication-based cyber threats. This report serves as a practical guide for cybersecurity teams in formulating structured incident response strategies.

## 6.0. References:

UpGuard. (n.d.). *Cyber incident reporting: Best practices for protecting your organization*.

UpGuard. <https://www.upguard.com/blog/cyber-incident-reporting>

National Institute of Standards and Technology. (n.d.). *The Cybersecurity Framework*.

U.S. Department of Commerce. <https://www.nist.gov/cyberframework>

## 7.0 Appendix:

### 7.1 Files

#### Splunk Dashboard:

[Hydra Attack Compromising Kali's System 2025-03-07 at 11.52.52-0500 Splunk](#)

#### Splunk Reports:

[Hydra\\_attack\\_analysis-2025-03-08](#)

[nmap\\_scan\\_analysis-2025-03-08](#)

#### Log Files:

[Hydra\\_attack\\_report.txt](#)

[nmap\\_scan\\_results.txt](#)

### 7.2 Further Notes:

There were issues with time parsing for the *hydra\_attack\_report.txt*. This would compromise the integrity of any analysis I would make using Splunk. To ensure Splunk correctly recognised the times from the logs, I was required to enter Linux's command line, and manually change the time format for this file through the *props.conf* file. This file, and its corresponding directory, were not available on my local Kali Linux virtual machine, meaning I had to create them under the correct parent directories. I also didn't initially have write permissions to files in these directories. I fixed this by giving myself write permissions, using *sudo chmod u+w \$SPLUNK\_HOME/etc/system/local/props.conf*

After editing the *props.conf* file using *sudo nano*

`$SPLUNK_HOME/etc/system/local/props.conf`, I restarted Splunk and confirmed that the times were correctly parsed on the relevant file.