TryHackMe Corridor Lab – VAPT Report

Prepared By: Nirmiti Dhawade

Role: Red Team Intern

Platform: TryHackMe

Submission: Internship Report (The Cyber Ledger)

Date: November 2025

1. Introduction

The TryHackMe **Corridor Lab** simulates a lightweight web-based challenge where the objective is to identify hidden information, extract encoded values, and enumerate the application using foundational cybersecurity techniques. This report outlines the systematic methodology followed during the assessment, including reconnaissance, scanning, inspection, and hash-cracking activities.

The engagement followed a structured, industry-aligned workflow combining reconnaissance, analysis, and exploitation steps. Each stage includes a designated screenshot placeholder for reference.

---

2. Scope of Assessment

- Target: TryHackMe *Corridor* Lab
- Type: Web Application Enumeration & Basic Exploitation
- Skills Applied:
  - Network Scanning
  - Web Inspection
  - Hash Cracking (MD5)
  - CyberChef Operations
  - On-page Flag Identification

---

3. Methodology

The assessment followed a linear, practical approach:

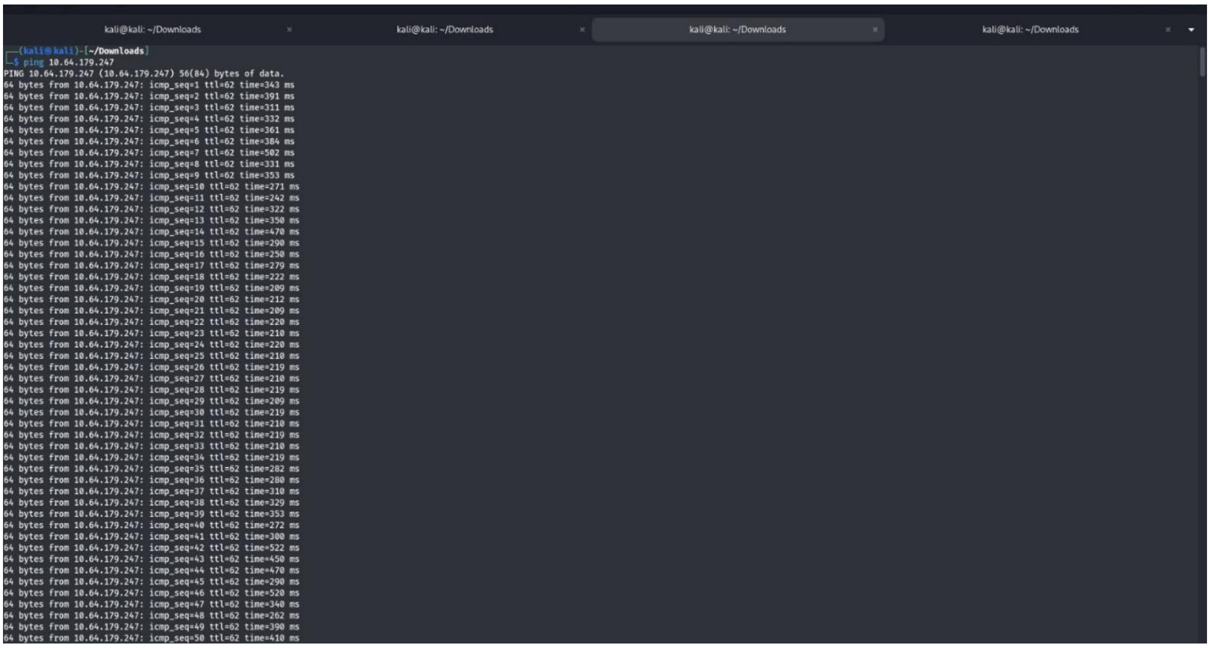
1. Connectivity Testing (Ping)
2. Service Enumeration (Nmap)
3. Web Interface Review (Port 80)
4. Hash Extraction
5. Hash Cracking (CrackStation)
6. Verification via CyberChef

7. Manual Inspection for Hidden Flag

---

4. Connectivity Validation (Ping Test)

To ensure the target was reachable, the **ping** utility was executed. Successful responses indicated that the host was active and accessible for further assessment.



5. Service Enumeration with Nmap

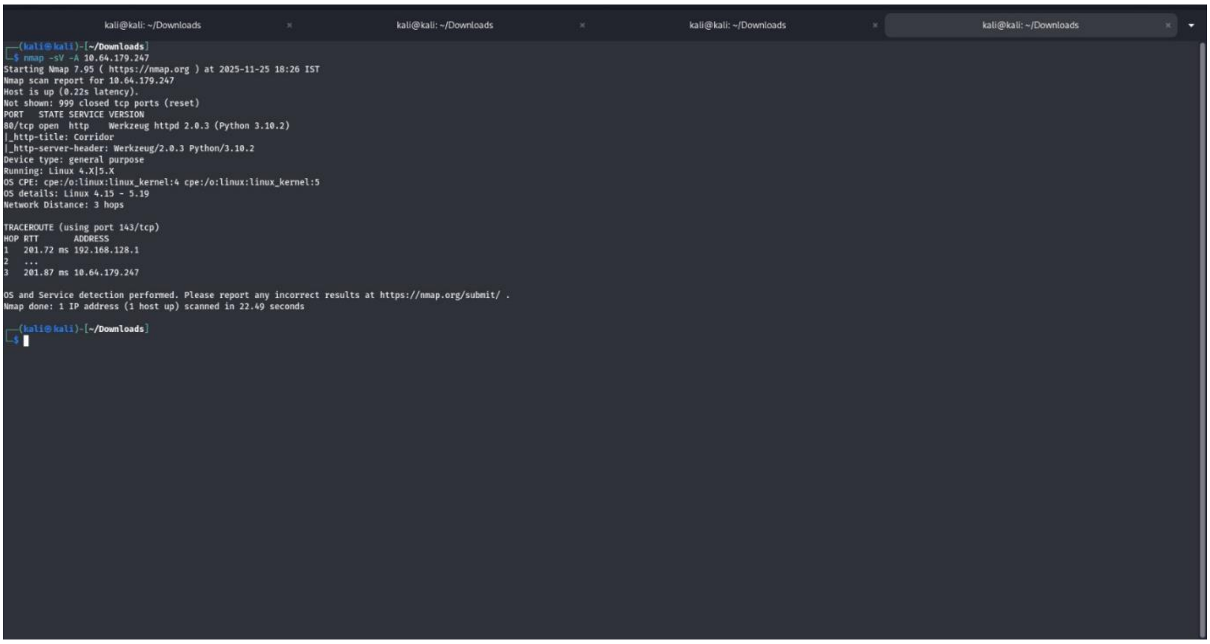
A detailed scan was conducted using:

nmap -sV -A <target-IP>

Key Observations:

- Port **80** was open.
- Detected service: HTTP
- No secondary high-risk attack surface was identified, confirming a single-page challenge structure.

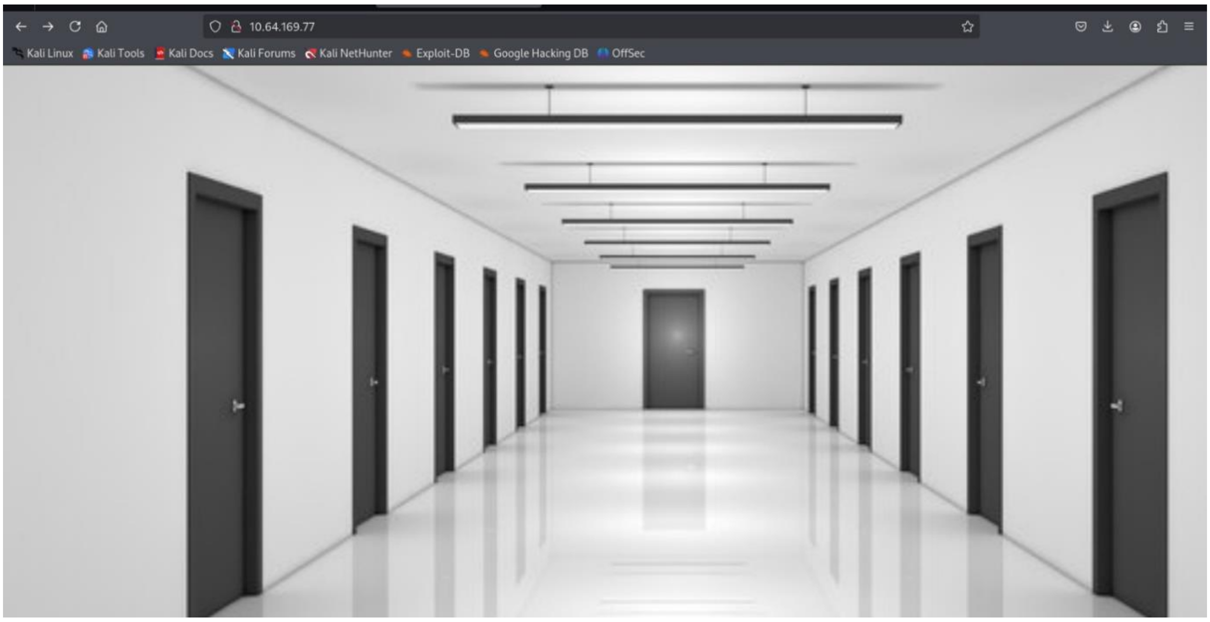
This step validated that the challenge was entirely web-based.



6. Web Access and Initial Analysis

Accessing the IP on port 80 displayed the corridor-style interface with numbers/inputs that hinted toward hidden encoded information.

No direct functionality was visible, so further inspection was required.



7. Hash Identification and Extraction

During interaction with the page, a series of encoded strings (MD5 hashes) were identified. These appeared to correspond to corridor door inputs.

Two key hashes were extracted:

- **Hash 1** → Result: *1*
- **Hash 2** → Result: *13*

These were cracked using external resources.

---

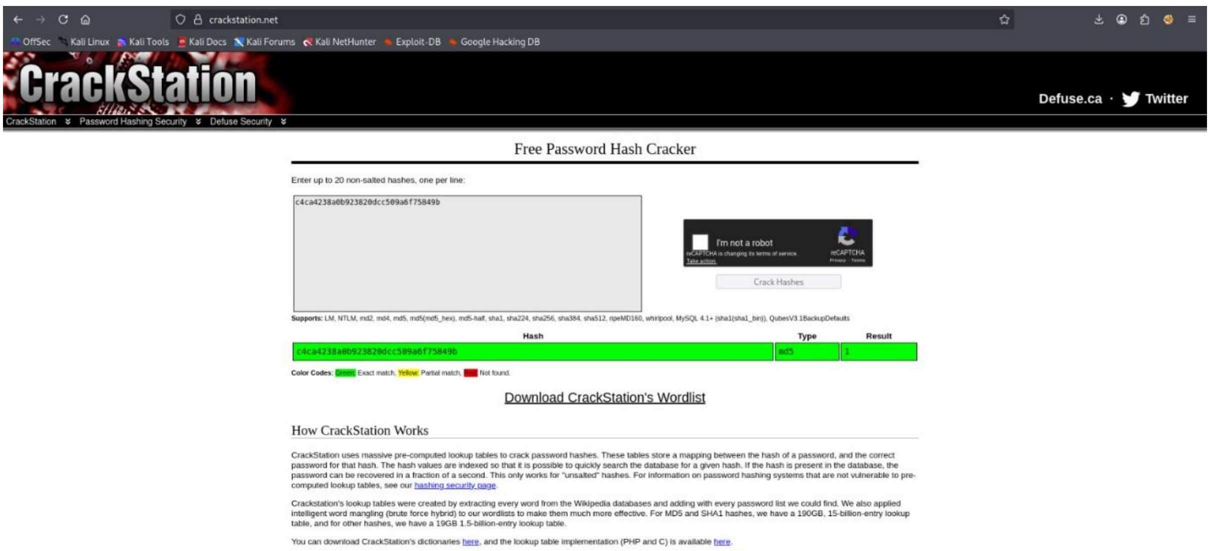
8. Cracking with CrackStation

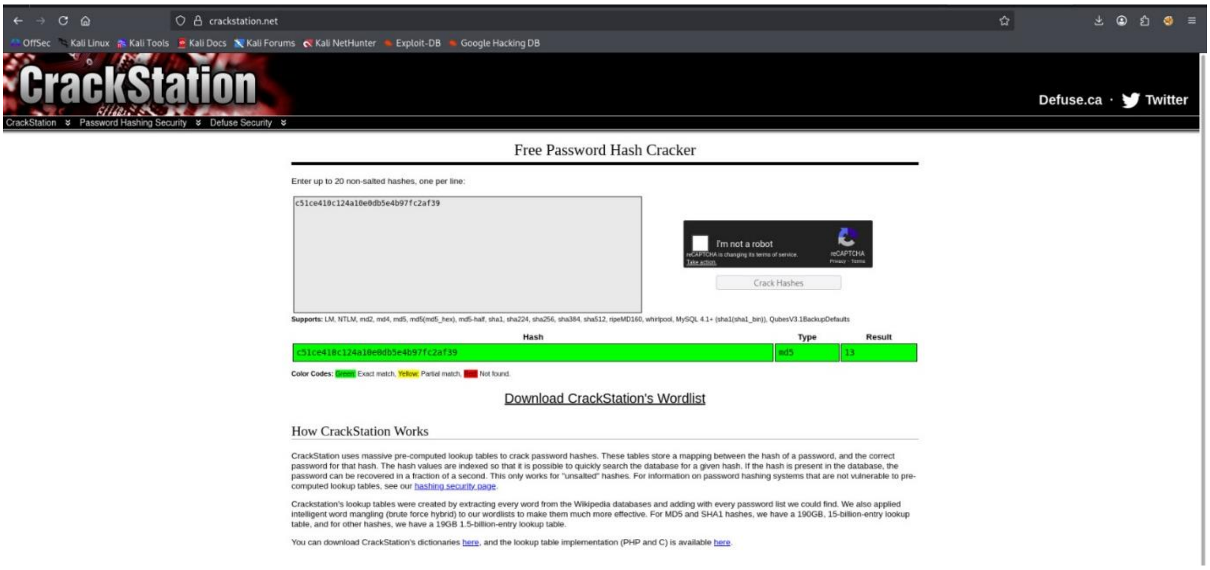
The extracted MD5 hashes were tested using **CrackStation**, allowing verification of the intended numeric values.

Cracked Results:

- Hash 1 → **1**
- Hash 2 → **13**

This validated the numeric door sequence.

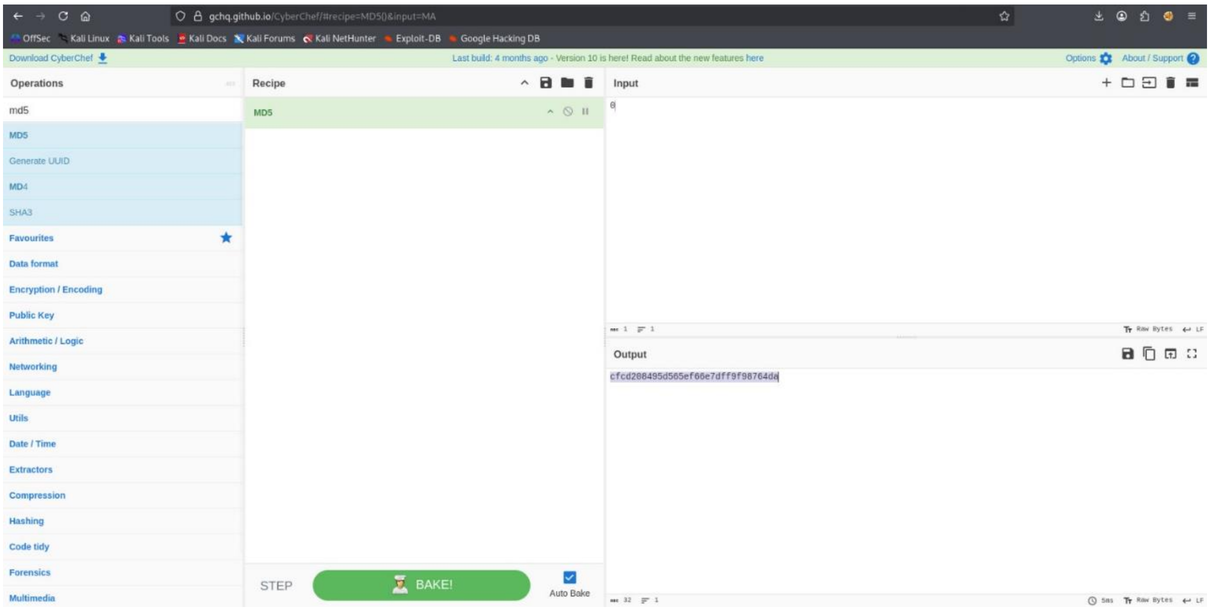




crackstation.net
OffSec
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
CrackStation
Defuse.ca · Twitter
CrackStation
Password Hashing Security
Defuse Security
Free Password Hash Cracker
Enter up to 20 non-salted hashes, one per line:
c51ce410c124a10e0db5e4b97fc2af39
I'm not a robot
reCAPTCHA is changing its terms of service.
Take action.
reCAPTCHA
Privacy - Terms
Crack Hashes
Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults
Hash
Type
Result
c51ce410c124a10e0db5e4b97fc2af39
md5
13
Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.
Download CrackStation's Wordlist
How CrackStation Works
CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our hashing security page.
Crackstation's lookup tables were created by extracting every word from the Wikipedia databases and adding with every password list we could find. We also applied intelligent word mangling (brute force hybrid) to our wordlists to make them much more effective. For MD5 and SHA1 hashes, we have a 190GB, 15-billion-entry lookup table, and for other hashes, we have a 19GB 1.5-billion-entry lookup table.
You can download CrackStation's dictionaries here, and the lookup table implementation (PHP and C) is available here.

9. Hash Verification using CyberChef

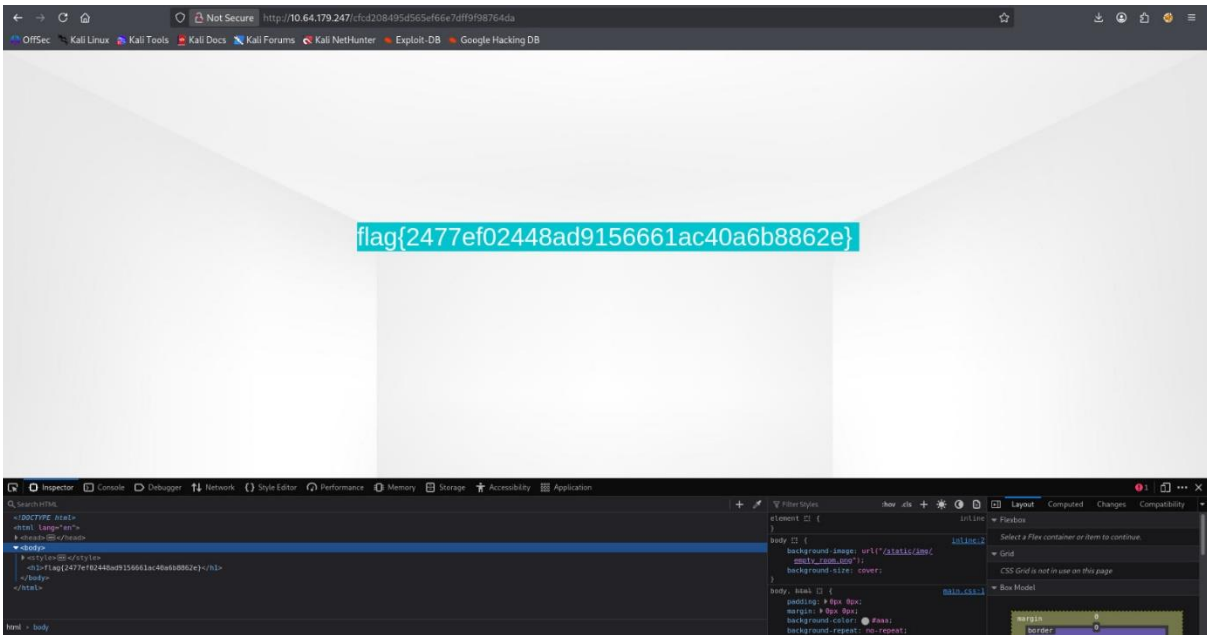
To ensure accuracy, the hash values were re-validated using **CyberChef**. By applying MD5 operations, the output matched the CrackStation result, establishing consistency.



10. Manual Web Inspection for Hidden Flag

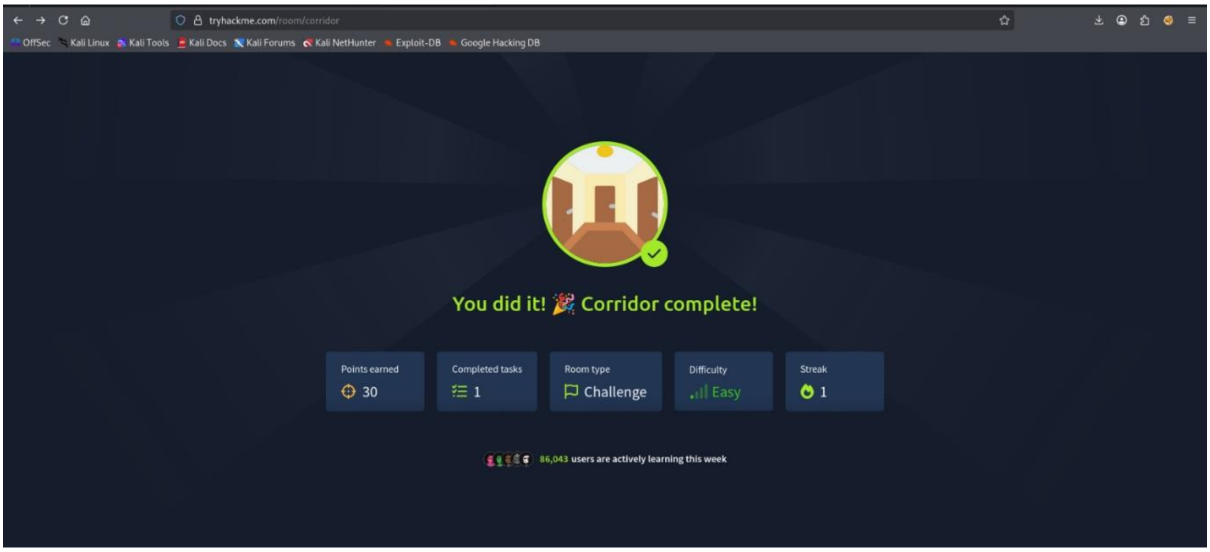
Final validation involved manually inspecting the webpage source using browser **Inspect Element**. The hidden flag was discovered within an HTML <title> attribute inside the page body.

This confirmed the conclusion of the challenge.



11. Completion Confirmation

The challenge platform confirmed successful completion.



12. Conclusion

The Corridor Lab demonstrated foundational skills essential for web-based assessments—reconnaissance, scanning, hash analysis, and source-code inspection. The structured approach helped uncover hidden values and flags efficiently. This methodology aligns with standard VAPT practices used in professional engagements.

---

13. References

- TryHackMe: Corridor Room
- CrackStation Hash Cracking Tool
- CyberChef (GCHQ)
- Nmap Official Documentation
- Browser Developer Tools (Inspect Element)
- Linux Networking Utilities (Ping)