Penetration Test Report

TryHackMe - SimpleCTF (practice)

Prepared by

Ambar Roy

Contact: ambarroy11@gmail.com

Report date: October 19, 2025

Engagement type: Practice / Training pentest (CTF-style)
Scope: SimpleCTF lab on TryHackMe (single VM/challenge)
Tester: Freelance security practitioner (no company affiliation)

Contents

Executive Summary

This report documents the findings of a practice penetration test against the SimpleCTF TryHackMe challenge. The purpose of this assessment is educational, aimed at practicing reconnaissance, enumeration, exploitation, and reporting skills.

Key points:

- Engagement: Practice lab; no production assets.
- Scope: Single VM labeled "SimpleCTF" on TryHackMe.
- Methodology: Reconnaissance, enumeration, exploitation, and post-exploitation using tools including nmap, gobuster, SSH, Python scripts, and wordlists.
- Summary of findings: Three main findings were identified — CMS identification, service enumeration, and exploitation via SQL injection.

1 Scope and Rules of Engagement

Scope

- Target: SimpleCTF VM (TryHackMe)
- IP / Hostname: 10.201.127.76
- Engagement type: Educational/practice — authorized by TryHackMe for this environment.

Rules of Engagement

- Only the specified VM was tested.
- No actions outside the TryHackMe lab environment were performed.
- All findings are for educational purposes only.

1.1 Methodology and Tools

High-level methodology:

1. Reconnaissance — network scan and port/service discovery.
2. Enumeration — web content, directories, and service-specific analysis.
3. Exploitation — verified vulnerabilities and captured PoC evidence.
4. Reporting — documented findings, impact analysis, and remediation.

Tools used:

- **Network/host:** nmap, netcat (nc), ping
- **Web:** gobuster/dirbuster, curl, wget
- **Exploitation:** Python scripts (from Exploit-DB), SSH, wordlists (rockyou.txt)
- **Other:** strings, grep, vim, Burp Suite (optional)

2 Risk Rating and Definitions

- **High:** Immediate, significant impact (credential compromise, root access, full system compromise).
- **Medium:** Significant but limited impact (sensitive information exposure, partial compromise).
- **Low:** Information disclosure or low-risk exposure (version info, metadata, non-critical paths).

3 Findings

Finding #1 — CMS Identification via Directory Discovery

Finding #1: CMS version disclosure via directory enumeration **Risk: Low / Informational**

Affected Asset: http://10.201.127.76

Summary:
Directory enumeration revealed a '/simple' path, exposing that the website runs *CMS Made Simple* version 2.2.8. This information disclosure could assist attackers in finding version-specific exploits.

Description:
A gobuster scan revealed the '/simple' directory, which redirects to 'http://10.201.127.76/simple/'. Visiting the page shows a footer displaying the CMS and version.

Commands / PoC:

```
gobuster dir -u http://10.201.127.76 -w /usr/share/dirbuster/wordlists/
    ↪ directory-list-lowercase-2.3-medium.txt -t 100
```

Page footer observed:

> © Copyright 2004 - 2025 - CMS Made Simple
> This site is powered by CMS Made Simple version 2.2.8

Impact:
Low-risk information disclosure; makes targeted attacks easier if the version is known to have vulnerabilities.

Remediation:

1. Remove CMS/version info from page footers or metadata.
2. Keep CMS and plugins updated to the latest secure version.
3. Apply general web hardening (WAF, restricted admin paths, log monitoring).

Finding #2 — Host and Service Enumeration

Finding #2: Service discovery and potential attack vectors **Risk: Medium**

Affected Asset: 10.201.127.76:21,80,2222

Summary:
Network and service enumeration identified FTP (anonymous login allowed), HTTP (Apache 2.4.18), and SSH on a non-standard port (2222). Combined with the CMS version, this increases attack surface.

Description:
Exploitation research indicated a SQLi in CMS Made Simple $< 2.2.10$. Nmap enumeration revealed additional potential attack vectors via FTP and SSH.

Commands / PoC:

```
searchsploit "CMS␣Made␣Simple␣v␣2.2.8"
```

```
nmap -A 10.201.127.76
```

Nmap output highlights:

- FTP (21): vsftpd 3.0.3, anonymous login allowed.
- HTTP (80): Apache/2.4.18, robots.txt disallows '/' and '/openemr-5_{013}'.
- SSH (2222): OpenSSH 7.2p2, non-standard port.

Impact:
Potential for credential discovery and further exploitation via SQLi, FTP, or SSH. Medium risk in a practice lab; in production could lead to full compromise.

Remediation:

1. Update CMS to version >= 2.2.10.
2. Disable or secure FTP (anonymous login not allowed).
3. Harden SSH: enforce keys, limit IPs, disable password auth.
4. Remove sensitive paths from public web directories.
5. Monitor logs for suspicious activity.

Finding #3 — SQL Injection Exploitation and Credential Compromise

Finding #3: Credential discovery via SQL injection leading to root access Risk: High

Affected Asset: http://10.201.127.76/simple/, SSH:2222

Summary:
A public exploit targeting CMS Made Simple < 2.2.10 was executed, resulting in discovery of account credentials. Using these credentials, sensitive files, including root-owned files, were accessed on the host.

Description:
The Exploit-DB script (46635.py) was copied and executed against '/simple/'. The script leverages SQL injection to enumerate users and crack passwords. Authenticated access to the host was then achieved, and root-owned files were viewed.

Commands / PoC:

```
# Copy exploit locally
searchsploit -m 46635

# Execute exploit
python2 46635.py -u http://10.201.127.76/simple/ --crack -w /usr/share/
    ↪ wordlists/rockyou.txt

# Outputs redacted: user credentials discovered
```

Post-exploitation actions:

- Authenticated SSH access using discovered credentials.
- Access to sensitive and root-owned files using vim.

Impact:
High — full system compromise possible, data exfiltration, and further lateral movement.

Remediation:

1. Upgrade CMS to 2.2.10 and patch plugins/extensions.
2. Secure credentials storage (strong hashing, MFA).
3. Restrict access to sensitive files; enforce least privilege.
4. Harden SSH and network access (keys, IP restrictions, disable password auth).
5. Monitor and audit logs; revoke exposed credentials.

Appendices

Appendix A: Common Commands and Tools

```
# Port scan (top 1000)
nmap -sC -sV -T4 -oN nmap_top1000.txt 10.201.127.76

# Full TCP scan
nmap -p- -T4 -oN nmap_allports.txt 10.201.127.76

# Web discovery
gobuster dir -u http://10.201.127.76/ -w /usr/share/wordlists/dirb/
    ↪ common.txt -t 50

# Download file
wget http://10.201.127.76/path/to/file -O downloaded-file

# Connect via netcat
nc 10.201.127.76 <port>
```

Appendix B: Evidence Files

- Place screenshots in the folder `screenshots/` and reference using `\EvidenceImage`.
- Example: `screenshots/finding3-exploit.png`

Contact and Notes

Prepared by: Ambar Roy
Contact: ambarroy11@gmail.com

Educational/practice engagement; all findings are lab-based.