PENETRATION TEST REPORT

Lookup

Web Application Security Assessment

CONFIDENTIAL

Engagement Type	Black-box assessment
Target	lookup.thm
Exposed Services	SSH (22), HTTP (80)
Host OS	Linux / Ubuntu
Web Server	Apache 2.4.41
Tester	Saqr ElFirgany
Report Date	25 June 2026
Environment	TryHackMe — authorised lab

This document contains confidential information about the security posture of the assessed target. The engagement was performed in an isolated, authorised training environment (TryHackMe). The techniques documented here must only be used against systems you are explicitly authorised to test.

Table of Contents

1. Executive Summary 3
2. Scope & Methodology 3
3. Findings Summary 4
4. Attack Narrative (Kill Chain) 4
5. Detailed Findings 5
6. Post-Exploitation 7
7. Remediation Summary 7
8. Appendix A — Services & Testing Performed 8
9. Appendix B — Username Enumeration Technique 8

1. Executive Summary

A black-box penetration test was performed against the host `lookup.thm` to identify and exploit weaknesses in its exposed services. Starting from an unauthenticated position, the tester was able to enumerate a valid application username, brute-force its password, authenticate to the application, and reach an outdated file-manager component vulnerable to remote code execution — establishing a foothold on the host.

The compromise was enabled by a chain of three weaknesses rather than a single flaw. The login form disclosed whether a username was valid through **distinct error messages**, allowing efficient username enumeration. The application enforced **no protection against automated guessing** and accepted a weak password, so the identified account was quickly brute-forced. Finally, authenticated access exposed an **outdated elFinder file manager (v2.1.47)** carrying a publicly known remote-code-execution vulnerability.

> **Overall Risk: Critical.** Although the entry point required authentication, the combination of username enumeration, no brute-force protection, a weak password, and a known-vulnerable component allowed a remote attacker to move from zero access to code execution. The outdated elFinder component (F-03) is the most urgent item; the authentication weaknesses (F-01, F-02) are what made reaching it trivial.

2. Scope & Methodology

Scope

A single host (`lookup.thm`) was in scope, assessed from an unauthenticated, external (black-box) perspective.

Methodology

The assessment followed a structured, surface-first methodology, separating enumeration from exploitation:

- **Reconnaissance** — full port and service-version scanning.
- **Enumeration** — virtual-host and content discovery; behavioural analysis of the login form.
- **Attack-surface mapping** — each entry point (SSH brute force, login enumeration, injection) recorded as a hypothesis before exploitation.
- **Exploitation** — username enumeration, password brute force, and exploitation of the vulnerable component.
- **Post-exploitation** — foothold establishment via the authenticated file manager.

3. Findings Summary

ID	FINDING	SEVERITY	AFFECTED COMPONENT
F-01	Outdated component with known RCE — elFinder 2.1.47	CRITICAL	Web app (post-auth)
F-02	No brute-force protection & weak account password	HIGH	login.php
F-03	Username enumeration via discrepant error messages	MEDIUM	login.php

4. Attack Narrative (Kill Chain)

1. Port scanning revealed only SSH (22) and HTTP (80). The web server presented a login page.
2. Content discovery indicated the application was served under the virtual host `lookup.thm`, which was added to local host resolution to access the site.
3. The login form (`login.php`) returned **different error messages** for a valid username with a wrong password versus a non-existent username — enabling **username enumeration (F-03)**.
4. Using this discrepancy as a filter, the username `jose` was identified.
5. With no rate limiting or lockout in place, the password for `jose` was **brute-forced (F-02)**, recovering `password123`.
6. Authenticated access exposed an **elFinder file manager, version 2.1.47 (F-01)**, which carries a publicly known remote-code-execution vulnerability.
7. The known exploit was used to achieve **remote code execution** on the host, establishing a foothold.

5. Detailed Findings

Outdated Component with Known RCE — elFinder 2.1.47

F-01 · CRITICAL

Severity: CRITICAL | **Component:** elFinder file manager (post-authentication) | **Type:** Use of Component with Known Vulnerability (CWE-1035 / CWE-78)

Description

After authenticating, the application exposed an elFinder web file manager running version **2.1.47 (2019-02-25)**. This version is affected by a publicly documented command-injection / remote-code-execution vulnerability. An authenticated attacker can leverage the known exploit to execute arbitrary commands on the underlying server.

Evidence

```
elfinder.html#elf_l1_Lw
elFinder — Version 2.1.47 (2019-02-25)   → known RCE
```

The documented exploit for this version was used to achieve command execution on the host.

Impact

Remote code execution on the server as the web-service account, providing an interactive foothold and a basis for further escalation.

Remediation

- Upgrade elFinder to the latest supported release, or remove it if it is not required.
- Establish a patch-management process to track and update third-party components against known CVEs.
- Restrict the file manager to trusted networks and enforce strong authentication and authorisation around it.

No Brute-Force Protection & Weak Account Password

F-02 · HIGH

Severity: HIGH | **Component:** login.php | **Type:** Improper Restriction of Excessive Auth Attempts / Weak Password (CWE-307 / CWE-521)

Description

The login form imposed no rate limiting, account lockout, or other anti-automation controls, permitting unlimited authentication attempts. Combined with a weak, dictionary-based password on the `jose` account, this allowed the password to be recovered through automated guessing in a short time.

Evidence

```
hydra -V -f -l jose -P /usr/share/wordlists/rockyou.txt \
  lookup.thm http-post-form \
  "/login.php:username=^USER^&password=^PASS^:F=Wrong password"

# equivalent ffuf approach
ffuf -w /usr/share/wordlists/rockyou.txt -X POST \
  -u http://lookup.thm/login.php \
  -d "username=jose&password=FUZZ" -fs 74

→ recovered password: password123
```

Impact

An attacker can reliably guess weak credentials and gain authenticated access, which in this case led directly to the vulnerable file manager (F-01).

Remediation

- Implement rate limiting, progressive delays, and account lockout after repeated failed attempts.
- Enforce a strong password policy and screen against known-breached password lists.
- Consider CAPTCHA and multi-factor authentication on the login flow.

Username Enumeration via Discrepant Error Messages

F-03 · MEDIUM

Severity: MEDIUM | **Component:** login.php | **Type:** Observable Response Discrepancy (CWE-204)

Description

The login form returned a different response depending on whether the submitted username existed. A valid username with an incorrect password produced a "wrong password" message, while a non-existent username produced a "wrong username or password" message. This observable difference allows an attacker to enumerate valid usernames, narrowing a brute-force attack to confirmed accounts.

Evidence

```
# Valid user, wrong password
POST /login.php  username=jose&password=wrongpassword
→ "Wrong password. Please try again."

# Invalid user
POST /login.php  username=notexist&password=wrongpassword
→ "Wrong username or password."
```

The differing string was used as an `ffuf` filter to enumerate usernames, identifying `jose`.

Impact

Disclosure of valid usernames significantly reduces the effort required for subsequent password attacks and supports targeted social-engineering.

Remediation

- Return a single, generic authentication-failure message regardless of which field was incorrect.
- Ensure response timing and length do not otherwise reveal account validity.

6. Post-Exploitation

Exploitation of the elFinder vulnerability (F-01) resulted in remote code execution on the host, establishing an initial foothold.

Note on documentation completeness: the source assessment notes conclude at remote code execution. Post-exploitation steps — local enumeration, retrieval of the user flag, and any privilege escalation to root — were not captured. For a complete and reproducible report, these steps should be documented at the time of testing, including the exact commands run and proof of access at each privilege level.

7. Remediation Summary

ID	PRIORITY ACTION	PRIORITY
F-01	Upgrade or remove elFinder; track third-party components against known CVEs.	IMMEDIATE

ID	PRIORITY ACTION	PRIORITY
F-02	Add rate limiting / lockout and enforce a strong password policy.	HIGH
F-03	Return generic, uniform authentication error messages.	MEDIUM

8. Appendix A — Services & Testing Performed

Discovered Services

PORT	SERVICE	VERSION / NOTES
22/tcp	SSH	OpenSSH 8.2p1 Ubuntu 4ubuntu0.9
80/tcp	HTTP	Apache 2.4.41 (Ubuntu) — "Login Page"; vhost lookup.thm

Testing Performed (including ruled-out paths)

The following avenues were considered and ruled out. Recording them is part of mapping the attack surface — eliminating a path is as valuable as confirming one.

AVENUE TESTED	RESULT
SSH — credential brute force	Not pursued / unsuccessful
login.php — SQL injection	No injectable behaviour observed
Apache 2.4.41 — version-specific public exploit	None applicable

9. Appendix B — Username Enumeration Technique

Username enumeration relied on the login form returning a measurably different response for valid versus invalid usernames. The `ffuf` tool was used to filter on this difference; the relevant filters are summarised below for reproducibility.

FFUF FILTER	PURPOSE
`-fr "text"`	Hide responses containing a regex/string (used to hide the "wrong username or password" message, revealing valid users).
`-fs N`	Filter by response size — useful when a valid response differs in length.
`-fw N`	Filter by word count.
`-fl N`	Filter by line count.

Practical note. A login form commonly returns `200 OK` for both valid and invalid attempts, so filtering on status code alone (e.g. `-fc 404`) is unreliable. Compare the **Size / Words / Lines** columns instead — the account whose response differs from the rest is typically the valid one.

Tester's note. The discovery that valid and invalid usernames produced different responses was reached by manually comparing the login form's replies, and that observation drove the rest of the chain. Documenting findings in this structured form is part of building a repeatable, independent testing methodology.