HACK THE BOX

Penetration Test Report

HTB CPTS: Giveback Machine

Report of Findings

Candidate Name: Security Analyst
Target: 10.129.242.171 (giveback.htb)
Classification: CONFIDENTIAL
Version: 1.0

Contents

1 Statement of Confidentiality 2

2 Engagement Contacts 2

3 Executive Summary 2
3.1 Approach 2
3.2 Scope 2
3.3 Assessment Overview and Recommendations 2

4 Network Penetration Test Assessment Summary 3
4.1 Summary of Findings 3

5 Internal Network Compromise Walkthrough 3
5.1 Detailed Walkthrough 3

6 Remediation Summary 4
6.1 Short Term 4
6.2 Medium Term 4
6.3 Long Term 4

7 Technical Findings Details 4
7.1 1. Unauthenticated PHP Object Injection to RCE (GiveWP) 4
7.2 2. Kubernetes API Abuse and Secret Extraction 4
7.3 3. OCI Runtime Configuration Exploitation 5

A Appendix 6
A.1 Finding Severities 6
A.2 Host & Service Discovery 6
A.3 Compromised Users 6
A.4 References 6

1 Statement of Confidentiality

This document contains confidential and proprietary information regarding the security posture of the assessed environment (**giveback.htb**). The information within this report is intended solely for the authorized stakeholders. Distribution, reproduction, or disclosure of this document, in whole or in part, without explicit permission is strictly prohibited.

2 Engagement Contacts

Name	Role	Email	Phone
Security Analyst	Penetration Tester	analyst@htb.local	555-0100
System Admin	Infrastructure Owner	admin@giveback.htb	555-0101

3 Executive Summary

3.1 Approach

The assessment was conducted using a black-box methodology, mirroring the approach of a real-world threat actor. The testing aimed to identify vulnerabilities in the external-facing web application and progressively escalate privileges to achieve full system compromise. The engagement included enumeration, initial access exploitation, lateral movement within containerized environments, and privilege escalation via orchestration and runtime abuse.

3.2 Scope

The scope of this assessment was strictly limited to the following target:

- **Hostname:** giveback.htb
- **IP Address:** 10.129.242.171
- **Environment:** Linux (Kubernetes)

3.3 Assessment Overview and Recommendations

The assessment revealed critical security vulnerabilities that allowed for a complete compromise of the target infrastructure. The initial foothold was achieved by exploiting a critical vulnerability (CVE-2024-8353) in an outdated WordPress plugin (GiveWP 3.14.0), which allowed Unauthenticated PHP Object Injection leading to Remote Code Execution (RCE).

Following initial access, the assessor navigated the containerized Kubernetes environment. By abusing the Kubernetes API, administrative credentials and secrets were extracted. Finally, misconfigured OCI (Open Container Initiative) runtime specifications were exploited to escalate privileges to the host's root user.

Strategic Recommendations:

- **Patch Management:** Immediately update the GiveWP plugin to the latest secure version to mitigate the RCE vulnerability.
- **Defense in Depth:** Enforce strict container isolation and implement robust Role-Based Access Control (RBAC) within the Kubernetes environment.
- **Secret Management:** Secure and encrypt Kubernetes secrets at rest to prevent unauthorized extraction.

4 Network Penetration Test Assessment Summary

4.1 Summary of Findings

During the assessment, several vulnerabilities were identified across different layers of the technology stack.

Finding Title	Severity
1. Unauthenticated PHP Object Injection to RCE (GiveWP)	**Critical**
2. Kubernetes API Abuse and Secret Extraction	**High**
3. OCI Runtime Configuration Exploitation (Privilege Escalation)	**High**

5 Internal Network Compromise Walkthrough

5.1 Detailed Walkthrough

The attack path involved a multi-stage approach, exploiting the application layer, the container orchestration platform, and the container runtime itself.

1. **Initial Reconnaissance & Web Enumeration:** Port scanning revealed an Nginx 1.28.0 web server running WordPress version 6.8.1. Directory and plugin enumeration using `wpscan` identified the *GiveWP* donation plugin (version 3.14.0) installed on the target.
2. **Initial Access (CVE-2024-8353):** The installed GiveWP plugin was found to be vulnerable to multiple Unauthenticated PHP Object Injection flaws (CVE-2024-5932, CVE-2024-8353, CVE-2024-9634). Exploiting the CVE-2024-8353 variant allowed the execution of arbitrary commands on the underlying web server, granting a reverse shell as the `www-data` user.
3. **Container Enumeration & Lateral Movement:** Upon gaining access, the environment was identified as a Kubernetes container. Standard container enumeration techniques were utilized to understand the pod's network and service account configuration.
4. **Kubernetes API Abuse:** By extracting the service account tokens mounted within the compromised pod, the Kubernetes API was queried. Weak RBAC permissions allowed the extraction of sensitive secrets, facilitating lateral movement across the cluster.
5. **Privilege Escalation (OCI Runtime Abuse):** The final stage of the attack involved exploiting custom binaries with `sudo` privileges and misconfigured OCI runtime specifications. By manipulating the container runtime configurations, the container isolation was bypassed, granting full root privileges on the underlying host system.

6 Remediation Summary

6.1 Short Term

- **Update GiveWP:** Immediately upgrade the GiveWP WordPress plugin to a version patched against CVE-2024-5932, CVE-2024-8353, and CVE-2024-9634.
- **Rotate Secrets:** Invalidate and rotate all compromised Kubernetes secrets, service account tokens, and administrative credentials.

6.2 Medium Term

- **Review Custom Binaries:** Conduct a thorough security review of all custom binaries utilizing `sudo` privileges within the containerized environment.
- **Audit Kubernetes RBAC:** Audit and restrict Kubernetes Role-Based Access Control configurations to enforce the principle of least privilege.

6.3 Long Term

- **Harden Container Runtimes:** Implement secure OCI runtime configurations. Prevent unauthorized manipulation of runtime specs and enforce strict container isolation boundaries.
- **Defense in Depth:** Establish a comprehensive vulnerability management program covering the application layer, orchestration platform, and host systems.

7 Technical Findings Details

7.1 1. Unauthenticated PHP Object Injection to RCE (GiveWP)

Severity: Critical
Affected Component: WordPress GiveWP Plugin (v3.14.0) on giveback.htb

Description: The application utilizes an outdated version of the GiveWP plugin which is vulnerable to unauthenticated PHP Object Injection (CVE-2024-8353). Because the plugin poorly deserializes user-supplied data, an unauthenticated attacker can craft a malicious serialized payload that, when processed by the application, results in arbitrary code execution.

Impact: This vulnerability allows a remote, unauthenticated attacker to execute system commands on the web server, leading to a complete compromise of the web application layer and acting as a foothold into the internal Kubernetes environment.

Remediation: Update the GiveWP plugin to the latest version recommended by the vendor. Ensure all web application components are monitored via a strict patch management policy.

7.2 2. Kubernetes API Abuse and Secret Extraction

Severity: High
Affected Component: Kubernetes Cluster (RBAC / Service Accounts)

Description: The compromised web pod possessed overly permissive service account tokens. These tokens were utilized to authenticate against the internal Kubernetes API. Due to improper RBAC limitations, the account was authorized to query and extract sensitive secrets from the cluster.

Impact: Extraction of Kubernetes secrets allows an attacker to pivot, escalate privileges within the cluster, and access sensitive backend databases or infrastructure components.

Remediation: Implement the Principle of Least Privilege for all Kubernetes service accounts. Ensure pods only have access to the specific secrets required for their operational tasks. Enable Kubernetes Secret encryption at rest.

7.3 3. OCI Runtime Configuration Exploitation

Severity: High
Affected Component: Container Runtime / Host OS

Description: The environment contained custom binaries executable via `sudo` that interacted with the Open Containers Initiative (OCI) runtime specifications. By manipulating the OCI runtime configuration files, the attacker was able to instruct the runtime to execute malicious payloads with elevated privileges outside the boundaries of the container.

Impact: This misconfiguration completely breaks container isolation, allowing an attacker to escape the container and achieve root-level administrative access on the underlying host machine.

Remediation: Restrict the ability for non-root container users to modify OCI runtime configurations. Avoid granting `sudo` privileges to custom binaries that interact with low-level container orchestrations unless strictly validated.

A Appendix

A.1 Finding Severities

The following table dictates the severity ratings used throughout this report based on the potential impact on confidentiality, integrity, and availability.

Rating	CVSS Score Range
Critical	9.0 – 10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9
Info	0.0

A.2 Host & Service Discovery

IP Address	Port	Service	Notes
10.129.242.171	80	HTTP	Nginx 1.28.0, WordPress 6.8.1

A.3 Compromised Users

Username	Type	Method	Notes
www-data	Local / Service	CVE-2024-8353	Reverse shell via GiveWP vulnerability.
root	Local / Admin	OCI Runtime Abuse	Container escape granting host root access.

A.4 References

- CVE-2024-5932: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-5932
- Kubernetes Security Best Practices: https://kubernetes.io/docs/concepts/security/
- OCI Runtime Specification: https://github.com/opencontainers/runtime-spec