Preignition

Preignition HTB

Cover

Target: HTB Machine “Preignition” **Client:** Megacorp (Fictitious) **Engagement Date:** May 2025 **Report Version:** 1.0

Prepared by: Jonas Fernandez

Confidentiality Notice: This document contains sensitive information intended solely for the recipient(s). Any unauthorized review, use, disclosure, or distribution is prohibited.

1. Introduction

Objective of the Penetration Test

The primary objective of this penetration testing engagement was to identify security weaknesses within a Linux-based target system hosted at **10.129.197.213**. Our evaluation focused on uncovering potential vulnerabilities in the web service configuration—specifically, insecure defaults and misconfigurations that could allow an attacker to easily access administrative functionalities. The goal was to expose these weaknesses and provide actionable recommendations to strengthen the overall security posture of the system.

Systems Evaluated & Methodology

The assessment centered on investigating the publicly accessible web interface of the target system. Our evaluation included:

- **Systems Evaluated:**
  - **Web Server:** Analyzing the default Nginx installation (version 1.14.2) serving a basic welcome page.
  - **Administrative Interface:** Identifying and testing the administrative endpoint discovered through directory fuzzing, including verifying the impact of using default credentials.
- **Methodology:** The testing was performed following industry-standard penetration testing methodologies:
  - **Reconnaissance:** Initial network discovery was conducted using Ping to confirm host availability and to infer that the target was Linux-based (indicated by a TTL of 63).
  - **Port Scanning:** A comprehensive Nmap scan was executed to identify open ports, revealing that only port 80 (HTTP) was accessible.
  - **Service Enumeration:** Detailed fingerprinting using both Nmap and WhatWeb confirmed the presence of an Nginx web server.
  - **Directory Fuzzing:** Utilized ffuf with a common wordlist to uncover hidden directories and files, which led to discovering the `admin.php` endpoint.

- **Authentication & Exploitation:** The admin interface was accessed using default credentials (admin:admin), which resulted in the retrieval of a flag from the administrative console—demonstrating the impact of misconfigured access controls.

Legal and Ethical Considerations

This penetration test was conducted with explicit authorization from the designated authority. All activities adhered strictly to ethical guidelines and industry best practices, ensuring that normal operations of the target system were not disrupted. The findings contained within this report are confidential and are intended solely for the designated stakeholders to support remediation efforts.

2 Methodology

The evaluation of the target machine (IP: 10.129.197.213) was conducted through a systematic and layered approach aimed at identifying weaknesses in the web service configuration. The process was as follows:

1. Host Discovery & OS Identification:

A simple `ping` test confirmed that the target system was online. The response (TTL=63) indicated that the operating system is Linux.

```
kali@kali ~ [15:28:18] $ ping -c 1 10.129.197.213
PING 10.129.197.213 (10.129.197.213) 56(84) bytes of data.
64 bytes from 10.129.197.213: icmp_seq=1 ttl=63 time=55.2 ms

--- 10.129.197.213 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 55.159/55.159/55.159/0.000 ms
```

2. Port Scanning & Service Enumeration:

A full TCP port scan was executed using Nmap with an aggressive rate to quickly identify open services. The scan revealed that only port 80 (HTTP) was open.

```
kali@kali ~/workspace/preignition [15:29:25] $ sudo nmap -sS -p- --open -n
-Pn --min-rate 5000 10.129.197.213 -oG Preignitionports
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-29 15:30 EDT
```

```
Nmap scan report for 10.129.197.213
Host is up (0.038s latency).
Not shown: 65104 closed tcp ports (reset), 430 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 11.58 seconds
```

3. Service Fingerprinting:

We performed a targeted scan on port 80 to gather detailed information about the running service. The results confirmed that an Nginx web server (version 1.14.2) was serving a default welcome page.

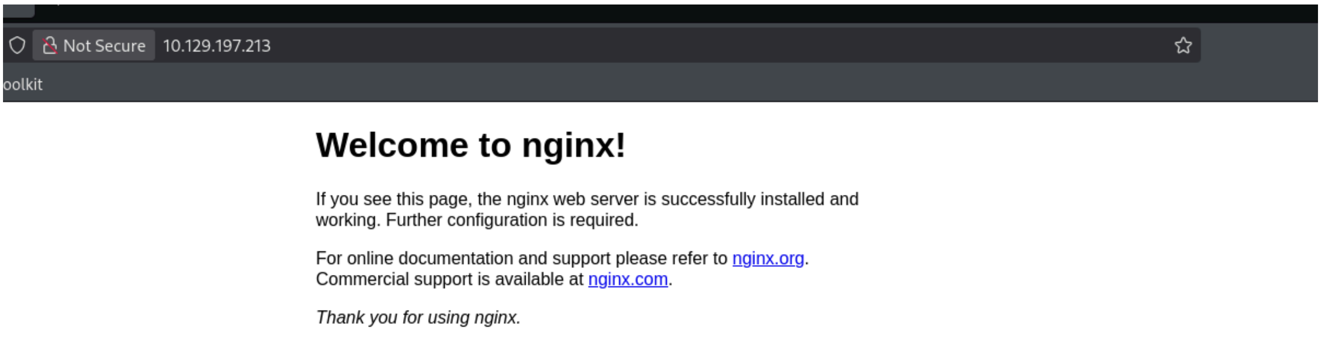
```
kali@kali ~/workspace/preignition [15:31:29] $ sudo nmap -sVC -p 80 10.129.197.213 -oN preignitionSErvices
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-29 15:32 EDT
Nmap scan report for 10.129.197.213
Host is up (0.052s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    nginx 1.14.2
|_http-server-header: nginx/1.14.2
|_http-title: Welcome to nginx!

```

4. Technology Identification:

The `whatweb` tool was used to further analyze the web service, confirming the presence of HTML5 and the Nginx version, alongside other server details.

```
kali@kali ~/workspace/preignition [15:32:24] $ whatweb 10.129.197.213
http://10.129.197.213 [200 OK] Country[RESERVED][ZZ], HTML5, HTTPServer[nginx/1.14.2], IP[10.129.197.213], Title[Welcome to nginx!], nginx[1.14.2]
```

Default website with nginx on the picture:



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

5. Directory Fuzzing:

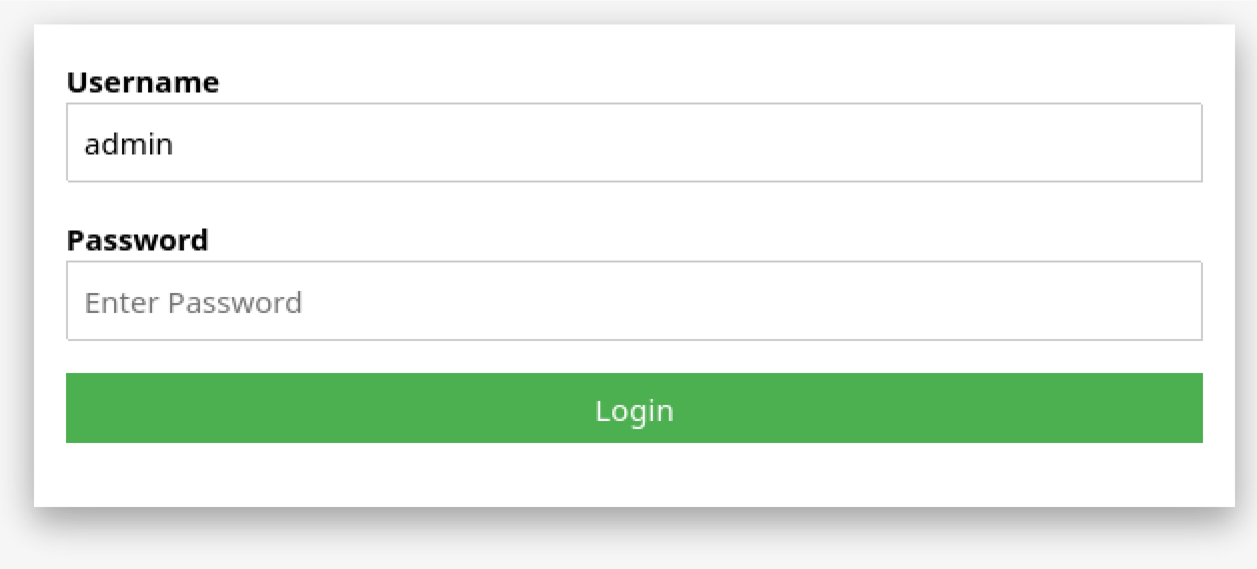
To uncover hidden or non-indexed resources, the `ffuf` tool was employed with a common web content wordlist. This process led to the discovery of the `admin.php` endpoint.

```
kali@kali ~/workspace/preignition [15:32:49] $ ffuf -w
/usr/share/wordlists/seclists/Discovery/Web-Content/common.txt  -u
http://10.129.197.213/FUZZ



        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.129.197.213/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/seclists/Discovery/Web-
Content/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-
299,301,302,307,401,403,405,500
________________________________________________
```

```
admin.php               [Status: 200, Size: 999, Words: 132, Lines: 32,
Duration: 41ms]
```

6.Authentication & Flag Retrieval:

Accessing the `admin.php` page using default credentials (`admin:admin`) granted entry to the administrative console. Once authenticated, the console displayed a flag with the message, "congratulations! your flag is:", confirming successful exploitation.



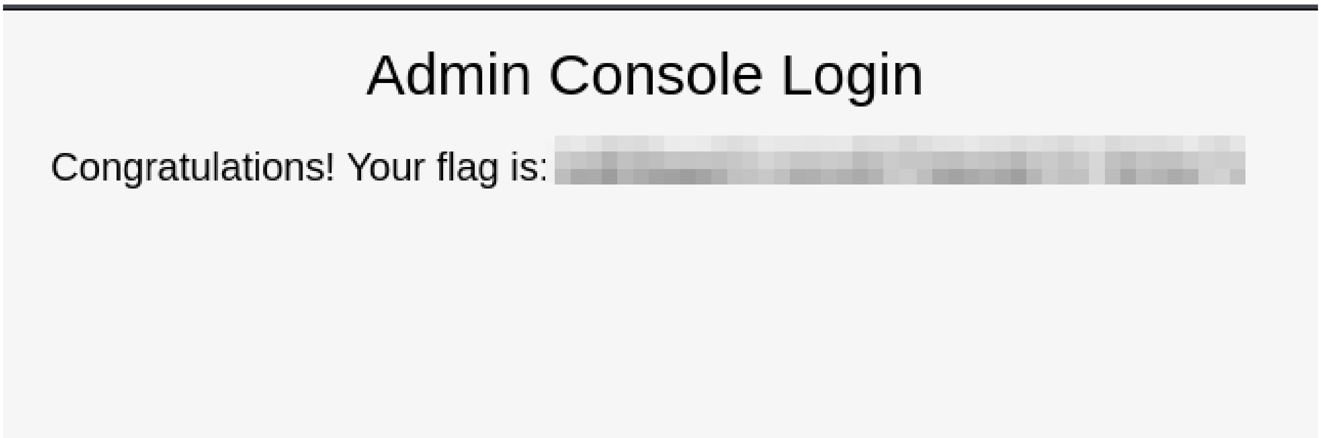
Username

admin

Password

Enter Password

And we get a flag on the Admin Console we see "Congratulations! your flag is:"

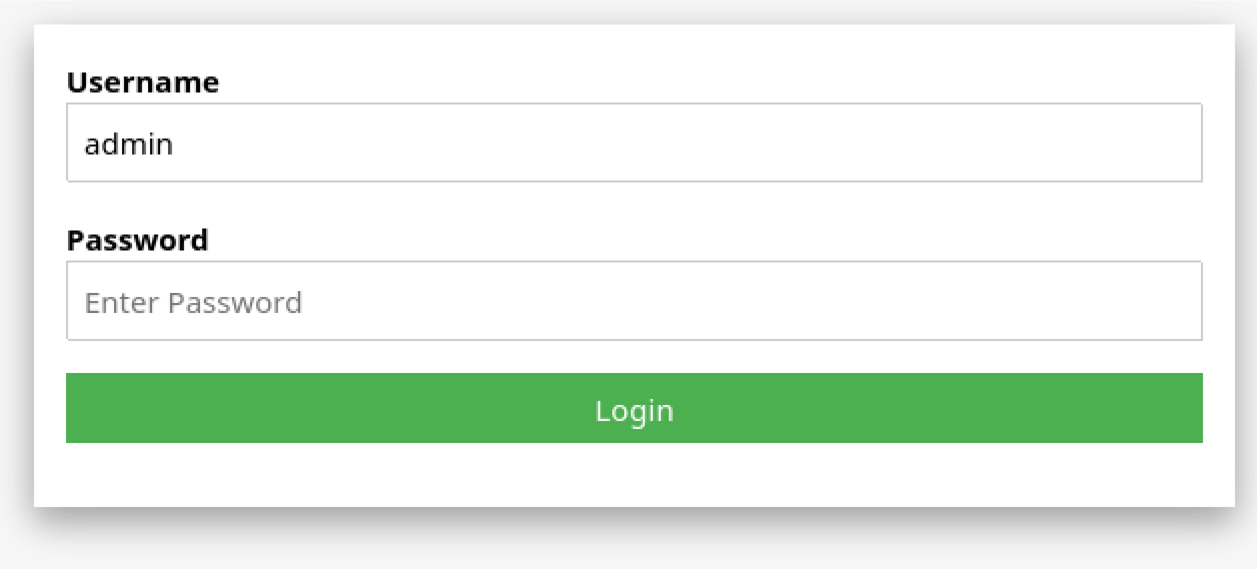


Admin Console Login

Congratulations! Your flag is: [redacted]

This methodical approach—from initial host discovery through service enumeration, directory fuzzing, and finally exploitation—enabled the swift identification and validation of vulnerabilities in the target's web interface. It underscores the critical importance of securing default configurations and routinely validating system exposures.

Vulnerability 1: Unauthorized Administrative Access via Default Credentials on Exposed Admin Interface

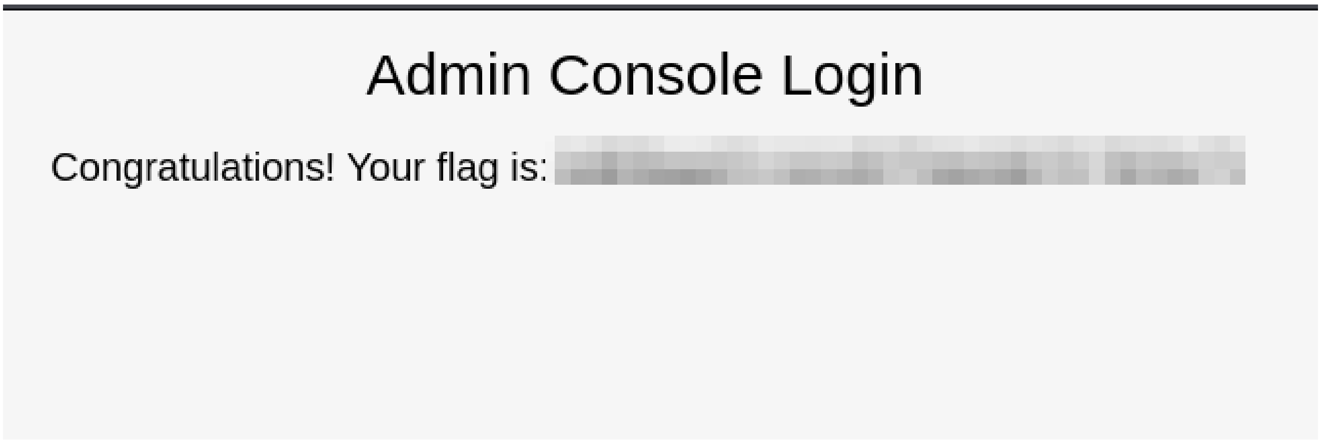


Username

admin

Password

Enter Password

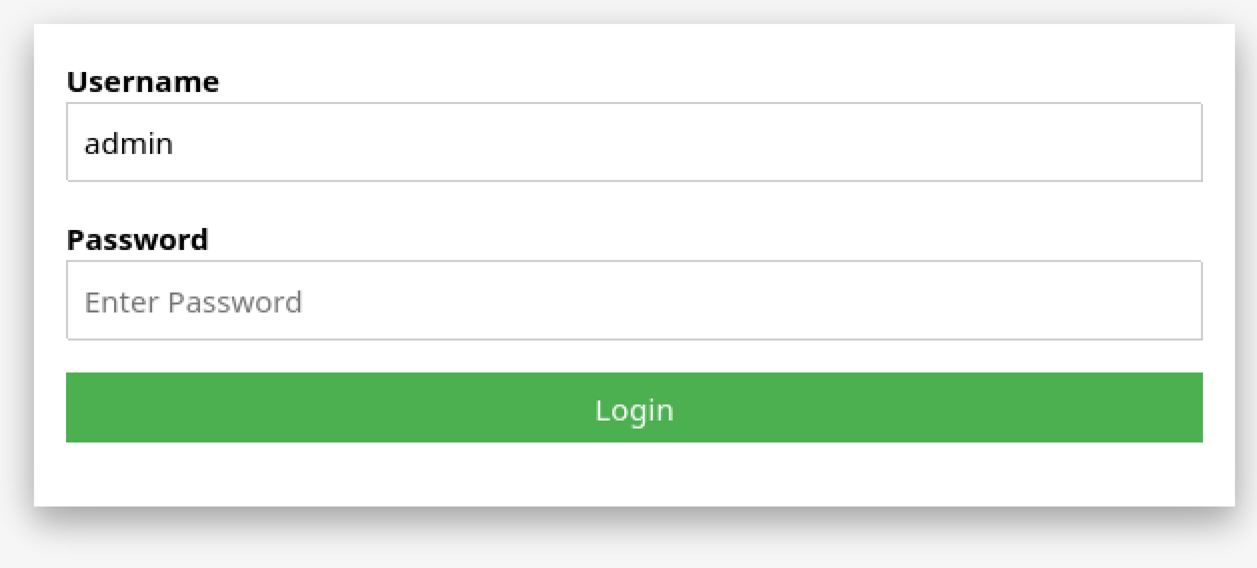


Admin Console Login

Congratulations! Your flag is: [redacted]

3 Findings

Vulnerability 1: Unauthorized Administrative Access via Default Credentials on Exposed Admin Interface

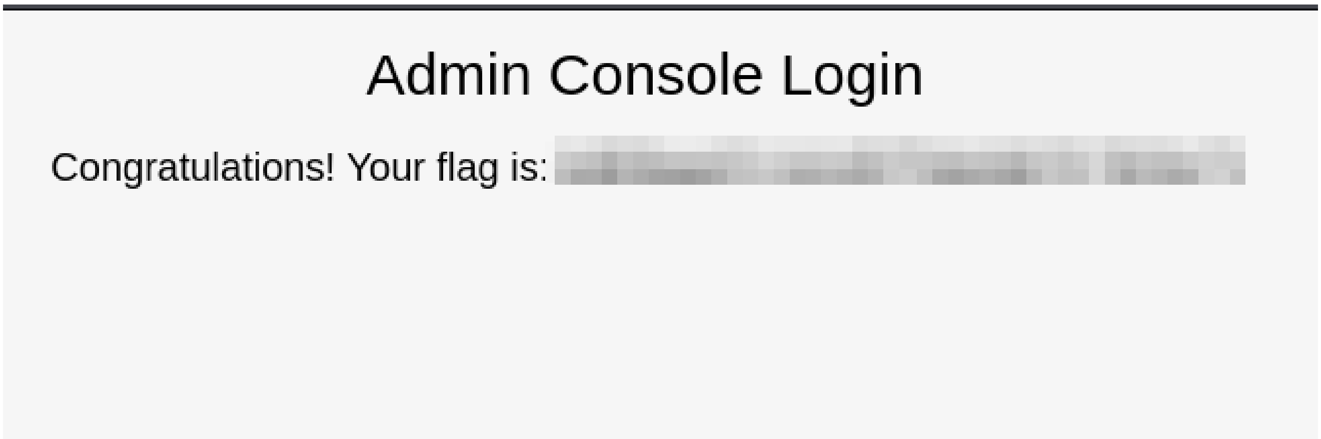


Username

admin

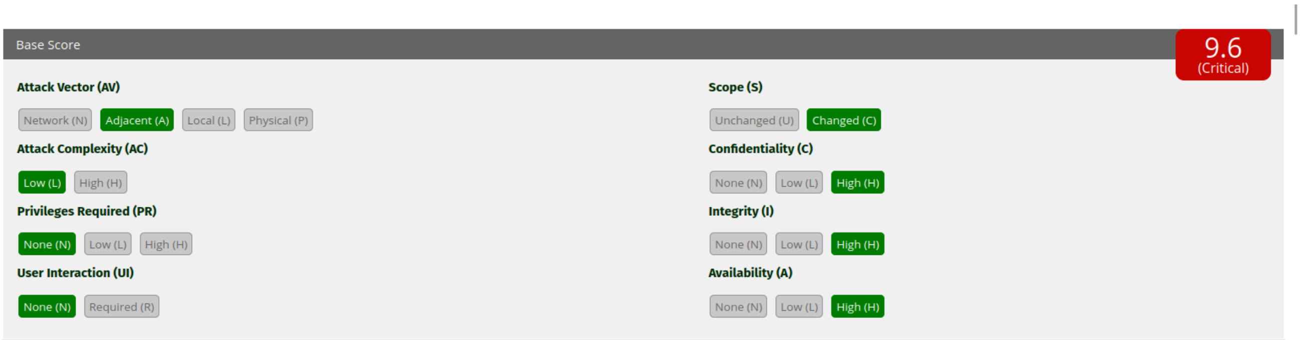
Password

Enter Password



Admin Console Login

Congratulations! Your flag is: [redacted]



Base Score 9.6 (Critical)

Attack Vector (AV)
Network (N) [✓ Adjacent (A)] Local (L) Physical (P)

Attack Complexity (AC)
[✓ Low (L)] High (H)

Privileges Required (PR)
[✓ None (N)] Low (L) High (H)

User Interaction (UI)
[✓ None (N)] Required (R)

Scope (S)
Unchanged (U) [✓ Changed (C)]

Confidentiality (C)
None (N) Low (L) [✓ High (H)]

Integrity (I)
None (N) Low (L) [✓ High (H)]

Availability (A)
None (N) Low (L) [✓ High (H)]

- **CVSS v3.1 Base Score: 9.6 (Critical) Metric Breakdown:** AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
- **Description:** A critical vulnerability was identified in the target web application's administrative interface. Our evaluation revealed that the admin page (`admin.php`) is accessible using default credentials (admin:admin), allowing unauthenticated access. This misconfiguration demonstrates a lack of proper security measures, such as enforced password changes or multi-factor authentication, which in turn exposes the system to significant risks.
- **Impact:** Exploiting this vulnerability grants an attacker complete administrative control over the web interface. Through this access, an attacker could modify critical settings, view sensitive data, execute unauthorized commands, and potentially pivot to other system components. The breach of an administrative account poses a severe risk to system integrity, confidentiality, and overall operational continuity.
- **Technical Details:**
  - **Discovery:** Using directory fuzzing with `ffuf` , the `admin.php` endpoint was discovered.
  - **Access Exploitation:** The web application was accessed via the admin interface using the default credentials (`admin:admin`), which resulted in successful authentication and the display of a flag on the admin console.
  - **Evidence of Exploitation:** The provided screenshots clearly show the login process and the subsequent retrieval of the flag message ("congratulations! your flag is:"), confirming the vulnerability.
- **Evidence:** The misconfiguration was confirmed through successful login and evidence captured from the administrative console, underscoring the critical nature of using default

credentials in production environments.

4 Recommendations

To mitigate the vulnerability associated with the exposed administrative interface accessible via default credentials, the following actions should be taken:

- **Remove Default Credentials:** Immediately replace all default login credentials (e.g., admin:admin) with strong, unique passwords. It is critical to enforce a password policy that prevents the reuse of easily guessable credentials and to implement multi-factor authentication where applicable.
- **Restrict Access:** Limit exposure of the administrative interface by configuring access controls—such as IP whitelisting or restricted VPN access—to ensure that only authorized personnel can reach the interface from trusted locations.
- **Harden the Web Server:** Regularly update the web server software and audit the configuration to disable any unnecessary services or components. Removing or disabling unused files and endpoints minimizes the overall attack surface.
- **Implement Continuous Monitoring:** Establish robust logging and monitoring facilities to detect any unauthorized access attempts. Intrusion detection systems and regular vulnerability scans should be used to promptly identify and address anomalous activities.
- **Conduct Regular Security Audits:** Periodically review and test authentication mechanisms and system configurations to ensure compliance with industry best practices. Regular penetration testing and configuration audits can help uncover any reintroduced weak defaults or misconfigurations.

By addressing these recommendations, the risk associated with insecure default configurations will be significantly reduced, thereby safeguarding sensitive data and enhancing the overall security posture of the organization.

5. Conclusion

Executive Summary

Our assessment uncovered a critical vulnerability that undermines the security posture of the target system. Imagine a secure building where every door is fortified except for one window left wide open—an unintended entry point for intruders. In our case, that “window” exists in the web application's administrative interface. The system was configured with default credentials (admin:admin), which allowed unauthorized access. This means an attacker can gain full control of the administrative functions without any authentication, potentially leading to severe consequences for business operations and reputation. Immediate remediation is essential to eliminate this exposure.

Technical Summary

Our detailed analysis revealed that the administrative console, running under an Nginx web server (version 1.14.2) on a Linux host, is accessible via default credentials. By using a directory fuzzing tool, we discovered the `admin.php` endpoint, and upon accessing it with the default username and password, we successfully authenticated into the system. This misconfiguration bypasses all standard security mechanisms, granting an attacker unrestricted privileges. With such access, a malicious actor could alter configurations, exfiltrate data, or pivot to additional, more sensitive areas of the network.

Current Security Posture and Future Steps

Current Risk Assessment: The exploitation of this vulnerability grants attackers unfettered access to critical administrative functions, making the system highly susceptible to data breaches, operational disruptions, and significant reputational damage.

Immediate and Long-Term Actions:

1. **Immediate Remediation:**
   - **Remove Default Credentials:** Replace all default login details with strong, unique passwords. Enforce a robust authentication policy.
   - **Restrict Interface Access:** Limit access to the administrative interface through IP filtering or VPN-based restrictions, ensuring that only authorized users can reach it.
2. **Enhanced Security Controls:**
   - **Implement a Layered Defense Strategy:** Adopt additional security measures such as multi-factor authentication, network segmentation, and continuous monitoring to create multiple barriers against potential breaches.
   - **Regular Security Assessments:** Conduct periodic vulnerability assessments and penetration tests to verify that all configurations adhere to current security best practices.
   - **Employee Awareness:** Provide frequent training and updates to all staff to reinforce the importance of secure credential practices and overall security hygiene.

By addressing this vulnerability promptly and comprehensively, we can strengthen our defenses and safeguard critical assets—ensuring operational continuity and maintaining trust with our stakeholders.

Appendix: Tools Used

This section details the primary tools used during the assessment, along with a brief explanation of each. These tools collectively provided insights into system configurations, identified vulnerabilities, and enabled exploitation during the pentest.

- **Ping Description:** A fundamental network diagnostic utility used to verify host availability and measure network latency. During the assessment, Ping confirmed that the target system was online and provided initial insights into the operating system based on the observed TTL value.

- **Nmap Description:** A versatile network scanning tool used to discover active hosts, open ports, and running services. Nmap was essential for mapping the target's network surface, identifying open endpoints, and verifying the services running on the host.
- **ffuf Description:** A robust directory fuzzing tool employed to discover hidden endpoints within web applications. In our evaluation, ffuf was instrumental in identifying the vulnerable `admin.php` interface, which was accessible using default credentials.
- **WhatWeb Description:** A web scanner that provides detailed information about the technologies used by a target website. WhatWeb helped fingerprint the target's web server—confirming that it was running Nginx version 1.14.2 and supporting HTML5.

These tools, when used as part of a systematic assessment methodology, provided the comprehensive insights necessary to identify and confirm the security weakness that allowed unauthorized administrative access.