PetHill Express Security Audit

Assessment Type: Express Website Scan

Objective: Identification of the attack surface and assessment of the web application's security configuration.

---

1. Methodology and Tools

The following tools were used during the assessment:

Category	Tools
Reconnaissance	Subfinder, Httpx
Network Analysis	Nmap
Fingerprinting	WhatWeb
Application Mapping	Gau, Katana
Hidden Resource Discovery	Dirsearch
Configuration Analysis	Nikto, Nuclei

The assessment included external perimeter analysis, technology stack identification, application mapping, hidden resource discovery, and web server security configuration analysis.

---

2. Assessment Results

2.1. External Perimeter

During the reconnaissance phase, active subdomains associated with the organization's infrastructure were identified. No evidence of additional test, staging, or forgotten environments directly expanding the attack surface of the assessed application was found.

2.2. Network Services

The scan identified the following publicly accessible services:

Port	Service
22/tcp	SSH
143/tcp	IMAP
8080/tcp	Web Service

The presence of these services does not constitute a vulnerability by itself; however, it increases the external attack surface and requires proper patch management and access control.

2.3. Technology Stack

Fingerprinting analysis identified the following technologies:

- Nginx
- Next.js
- React
- Node.js
- Webpack

The identified stack reflects a modern web application architecture.

2.4. Application Mapping

Automated enumeration revealed the following key application endpoints:

```
/api//api/auth/login/api/auth/register
```

These endpoints are related to authentication functionality and represent priority targets for future manual business logic and access control testing.

2.5. Hidden Resource Analysis

Directory fuzzing identified several system and service-related paths. Most discovered resources returned **403 Forbidden** responses, indicating properly configured access restrictions.

No evidence of unauthorized access to administrative or internal components was identified.

2.6. Security Configuration Analysis

The HTTP configuration review identified the absence of several recommended security headers:

- X-Frame-Options
- Content-Security-Policy (`frame-ancestors` directive)
- Strict-Transport-Security
- X-Content-Type-Options

The most significant issue is the lack of protection mechanisms against framing the application within external websites.

---

3. Identified Vulnerabilities

3.1. Clickjacking

Risk Level: Medium

Description

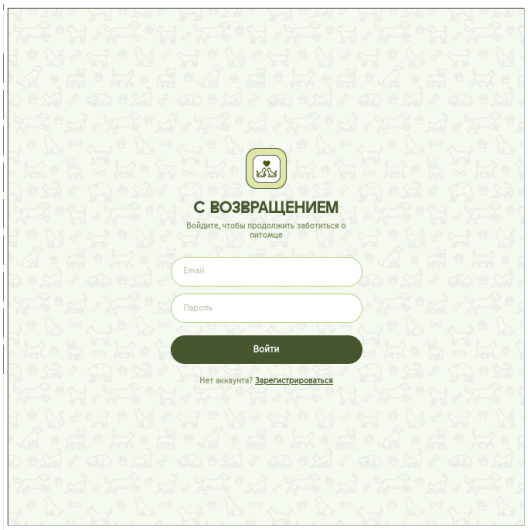
The authentication page can be rendered inside a third-party iframe due to the absence of the `X-Frame-Options` header and the `frame-ancestors` directive within the Content Security Policy.

As a result, an attacker may embed the application page into a malicious website and induce user interaction with interface elements without the user's awareness of the actual content source.

Evidence

Testing confirmed that the authentication page could be successfully loaded inside an iframe hosted on an external test environment. No browser-side or server-side restrictions prevented the embedding, confirming the exploitability of the issue.

PoC: Successful rendering of the authentication page within an external iframe.



Potential Impact

- Execution of Clickjacking attacks.
- Credential theft through deceptive interfaces.
- Unauthorized actions performed on behalf of authenticated users.
- Increased effectiveness of social engineering attacks.

Recommendations

To mitigate this vulnerability, it is recommended to prevent application pages from being embedded within external iframes.

The `X-Frame-Options` header allows browsers to determine whether a page may be rendered inside an iframe. Its implementation prevents Clickjacking attacks by blocking unauthorized framing of application content.

Recommended configuration:

```
add_header X-Frame-Options "DENY" always;
```

The `DENY` value completely prohibits framing. If framing is required within the same domain, the following option may be used:

```
add_header X-Frame-Options "SAMEORIGIN" always;
```

Additionally, it is recommended to implement Content Security Policy protection:

```
add_header Content-Security-Policy "frame-ancestors 'none';" always;
```

The `frame-ancestors` directive provides more flexible and granular control over permitted framing sources.

3.2. Security Headers Misconfiguration

Risk Level: Low

Description

The security review identified missing HTTP security headers:

- `Strict-Transport-Security (HSTS)`
- `X-Content-Type-Options`

The absence of these protections does not directly compromise the application; however, it reduces the overall security posture and may facilitate certain attack scenarios under specific conditions.

Evidence

The following security headers were not present:

```
Strict-Transport-SecurityX-Content-Type-Options
```

Potential Impact

- Exposure to SSL Stripping attacks during initial connections.
- Browser MIME-sniffing behavior.
- Reduced client-side security posture.

Recommendations

To improve client-side security protections, the implementation of additional HTTP security headers is recommended.

To protect users against SSL Stripping attacks, configure HTTP Strict Transport Security (HSTS):

```
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains"
always;
```

Once received, this header instructs browsers to use HTTPS exclusively for future connections to the application.

Additionally, protection against MIME Sniffing should be enabled:

```
add_header X-Content-Type-Options "nosniff" always;
```

This header prevents browsers from guessing content types and forces them to rely exclusively on the server-provided `Content-Type` value, reducing the risk of content misinterpretation.

After implementation, a follow-up verification of the HTTP security header configuration is recommended.

---

4. Additional Observations

SSH Version Disclosure

The assessment revealed SSH service version information. While not a vulnerability on its own, this information may assist attackers during the reconnaissance phase.

Recommendation: Maintain software updates and limit service version disclosure where feasible.

Publicly Accessible Services

Services on ports 22/tcp, 143/tcp, and 8080/tcp are accessible from the Internet.

Recommendation: Review the business necessity of exposing these services and restrict access through firewall policies whenever possible.

---

5. Conclusion

No critical vulnerabilities were identified during the assessment. Automated testing did not reveal evidence of SQL Injection (SQLi), Cross-Site Scripting (XSS), Remote Code Execution (RCE), Server-Side Request Forgery (SSRF), Local File Inclusion (LFI), Remote File Inclusion (RFI), or other critical security issues.

The most significant finding is the confirmed Clickjacking vulnerability caused by the absence of anti-framing protections.

Additional security header configuration weaknesses were identified, along with recommendations aimed at reducing the application's attack surface.

Based on the results of this assessment, no critical security deficiencies were identified. To further improve the application's security posture, it is recommended to address the identified configuration weaknesses and conduct a dedicated manual assessment of APIs, authentication mechanisms, authorization controls, and business logic.