Hack Smarter: Hunter

Web Application Penetration Test Report

Business Confidential

Date: Decemeber 15th, 2025
Project: Hunter
Version 1.0

Table of Contents

Confidentiality Statement

This article is the sole property of Hunter and Tech With Z (TWZ). This document includes private and confidential information. Duplication, dissemination, or use, in whole or in part, in any form, requires the permission of both Hack Smarter and TWZ.

Hunter may share this material with auditors under non-disclosure agreements to verify compliance with penetration testing requirements.

Disclaimer

A penetration test is viewed as a snapshot in time. The findings and recommendations are based on the information obtained during the evaluation and do not include any changes or revisions made beyond that period.

Time-limited engagements do not allow for a comprehensive assessment of all security controls. TWZ prioritized the evaluation to find the most vulnerable security controls an attacker may exploit. TWZ suggests that similar assessments be conducted on an annual basis by internal or third-party assessors to verify that the controls remain effective.

Contact Information

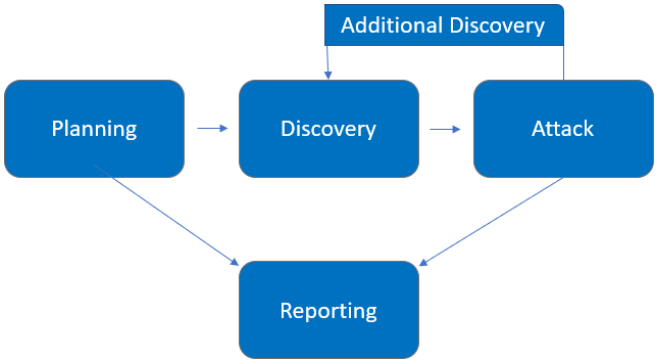
Tech With Z		
Name	Title	Contact Information
Zandro Dadulla Jr.	Penetration Tester	zandro@techwithz.com

Assessment Overview

Hunter engaged Tech With Z to assess the security status of its Web Application in relation to the latest industry standards, which involved conducting a web application penetration test. The testing conducted is based on the NIST SP 800-115 Technical Guide to Information Security Testing and Assessment, OWASP Testing Guide (v4), and tailored testing frameworks.

Phases of penetration testing activities include the following:

- Planning – Customer goals are identified, and rules of engagement are established.
- Discovery – Perform scanning and enumeration to identify possible vulnerabilities, weak points, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation, then conduct additional discovery with new access.
- Reporting – Document all discovered vulnerabilities and exploits, unsuccessful attempts, and company strengths and weaknesses.



Assessment Components

Web Application Penetration Test

This web application penetration test was conducted to simulate an attacker with no internal access or prior knowledge of the environment attempting to compromise the organization's exposed web assets. The objective was to identify security weaknesses that could be exploited to gain unauthorized access to sensitive data or application functionality.

The assessment involved active reconnaissance, including scanning and enumeration of the target's publicly accessible web infrastructure. Discovered endpoints, directories, and services were analyzed for common vulnerabilities, misconfigurations, and access control flaws. The testing process focused on identifying issues that could result in unauthorized data exposure, authentication bypass, privilege escalation, and improper access to application resources.

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Medium	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Risk Factors

Risk is measured by two factors: Likelihood and Impact:

Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, the attacker's skill level, and the client environment.

Impact

Impact measures the potential vulnerability's effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

Scope

Assessment	Details
Web Application Penetration Test	http://10.1.42.12

Scope Exclusions

TWZ did not conduct any of the following attacks during testing:

- Denial of Service (DoS) attacks against production infrastructure.
- Phishing / Social Engineering attacks.

Client Allowances

No allowances provided by the client.

Executive Summary

On December 12th, 2025, TWZ performed a web application penetration test targeting Hunter's external login portal. The objective of this assessment was to evaluate the security posture of Hunter's web application by identifying vulnerabilities and assessing the effectiveness of implemented security controls. This report provides a high-level overview of the findings, including both successful and unsuccessful exploitation attempts, as well as identified strengths and weaknesses in the application's security architecture.

Scoping and Time Limitations

Throughout the engagement, TWZ did not performed denial of service or social engineering in any testing components.

Testing was limited in duration. Web application penetration testing was approved until December 15th, 2025.

Testing Summary

The assessment evaluated Hunter's web application security posture specifically the external login portal. From an external perspective, the TWZ team performed information gathering techniques to identify possible entry points for future attacks and gather sensitive information.

The TWZ team discovered a flaw in the password reset portal where a valid username can be enumerated by analyzing server response time (Finding WPT-004). Utilizing this flaw, the TWZ team were able to send multiple requests without rate limiting (Finding WPT-003) and identified one valid username.

The team was then able to perform password brute-forcing to gain access to the account. The application allowed sending multiple requests without locking out the account (Finding WPT-001). However, even after multiple attempts of password attacks, the TWZ team were unsuccessful in gaining access to the account due to strong password complexity.

For further details on the findings, please see the Technical Findings section.

Tester Notes and Recommendations

During testing, a few things stood out: insufficient encryption on the website, insufficient authentication controls, and lack of rate limiting. The lack of rate limiting resulted in username enumeration which identified one valid account.

We advise that Hunter review its current web application infrastructure and implement TLS encryption (HTTPS). We also advise to implement rate limiting, response time delay, and CAPTCHA to reduce the number of requests in the reset password portal which could result in username enumeration.

Key Strengths and Weaknesses

The following identifies the key strengths identified during the assessment:

1. Effective protection against injection attacks (XSS, SQLi).
2. Effective protection against username enumeration in the Login page as there are no unique messages shown for both valid and invalid accounts.
3. Effective protection against username enumeration in the Password reset page using generic response message.
4. Excellent password complexity.

The following identifies the key weaknesses identified during the assessment:

1. Username enumeration via response time in the password reset page.
2. Insufficient lockout policy / rate limiting.
3. Insufficient encryption (HTTP).

Vulnerability Summary & Report Card

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

Web Application Penetration Test Findings

0	1	3	0	1
Critical	High	Medium	Low	Informational

Finding	Severity	Recommendation
Web Application Penetration Test		
WPT-001: Insufficient Lockout Policy	High	Restrict logon attempts against the login portal.
WPT-002: Insufficient encryption	Medium	Enforce HTTPS across the entire application.
WPT-003: Lack of rate limiting on Login and Password Reset endpoints	Medium	Implement request rate limiting / Implement CAPTCHA.
WPT-004: Username Enumeration via Password Reset response timing	Medium	Introduce delays to make response time for valid and invalid accounts the same.
WPT-005: Information Disclosure via HTTP Response Headers	Informational	Remove unnecessary information from HTTP response headers.

Technical Findings

Web Application Penetration Test Findings

Finding WPT-001: Insufficient Lockout Policy (High)

Description	Hunter allowed unlimited logon attempts against login portal. This configuration allowed brute force and password spraying attacks. However, due to excellent password complexity, the tester was unsuccessful in accessing the account.
CVSS Score	8.2 (High) – CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N
Risk	Likelihood: High – An attacker can utilize readily available tools such as Burp Suite, Caido, THC Hydra, etc. to perform brute-force and password spraying attacks without any restrictions to login attempts. Impact: High – If successful and given unlimited time, an attacker can gain access to a user's account sensitive information.
System	http://10.1.42.12
Tools Used	Caido
References	CWE-307: Improper Restriction of Excessive Authentication Attempts (4.17) Unsuccessful Logon Attempts: NIST SP 800-53 AC-07

Evidence / Steps to Reproduce

After identifying the valid user discussed in WPT-004, TWZ was able to perform a password spraying attack against the account without getting locked out. However, the test was unsuccessful in gaining access to the account due to excellent password complexity.

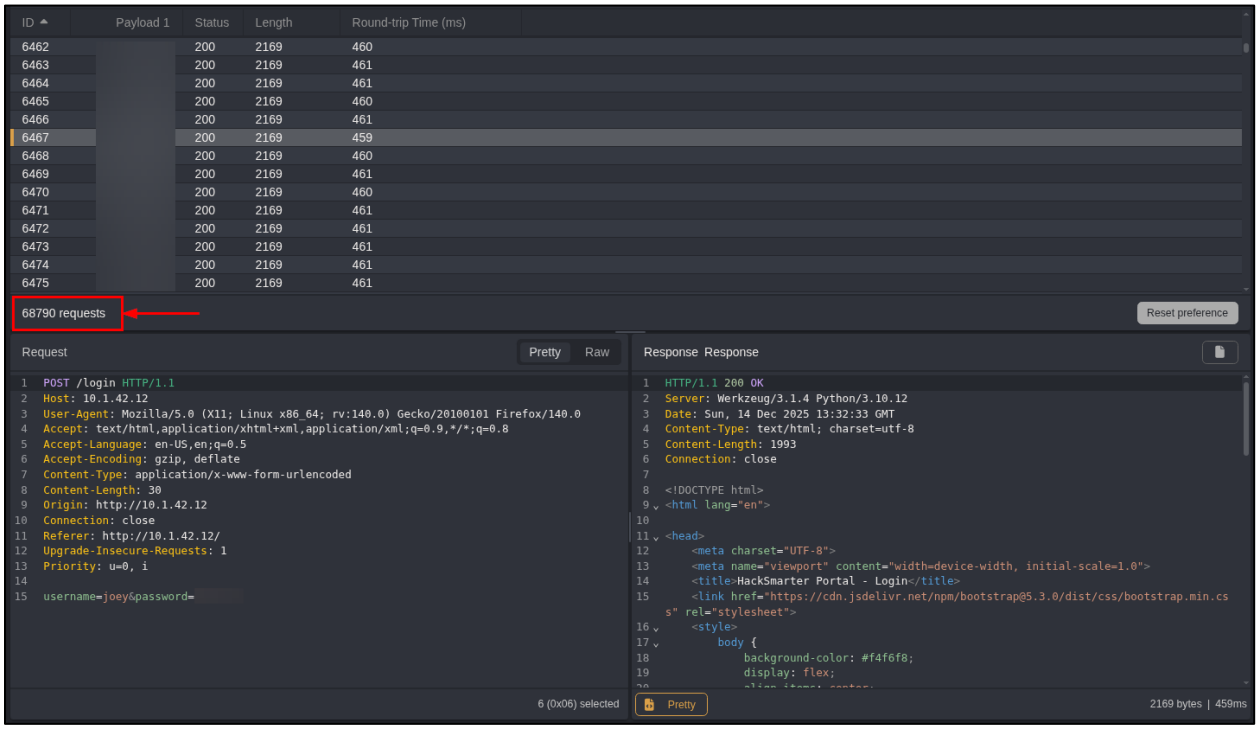


Figure 1: Caido Automate – Password Spraying Attack.

Remediation

TWZ recommends Hunter to implement lockout policy and apply rate limiting and delay mechanisms to prevent or slow down any brute-force attempts.

Finding WPT-002: Insufficient Encryption (Medium)

Description	Hunter does not use encryption on their website. Allowing cleartext transmission enables potential adversary-in-the-middle eavesdropping.
CVSS Score	5.4 (Medium) – CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N
Risk	Likelihood: Low – To conduct an adversary-in-the-middle attack, the attacker must be on the same network as the victim. Impact: High – Inadequate encryption may allow attackers to modify web content delivered to end users because it is not cryptographically signed via HTTPS.
System	http://10.1.42.12
Tools Used	Manual review
References	CWE-1428: Reliance on HTTP instead of HTTPS A02:2021: Cryptographic Failures OWASP Cheat Sheet: Transport Layer Security Cheat Sheet

Evidence / Steps to Reproduce

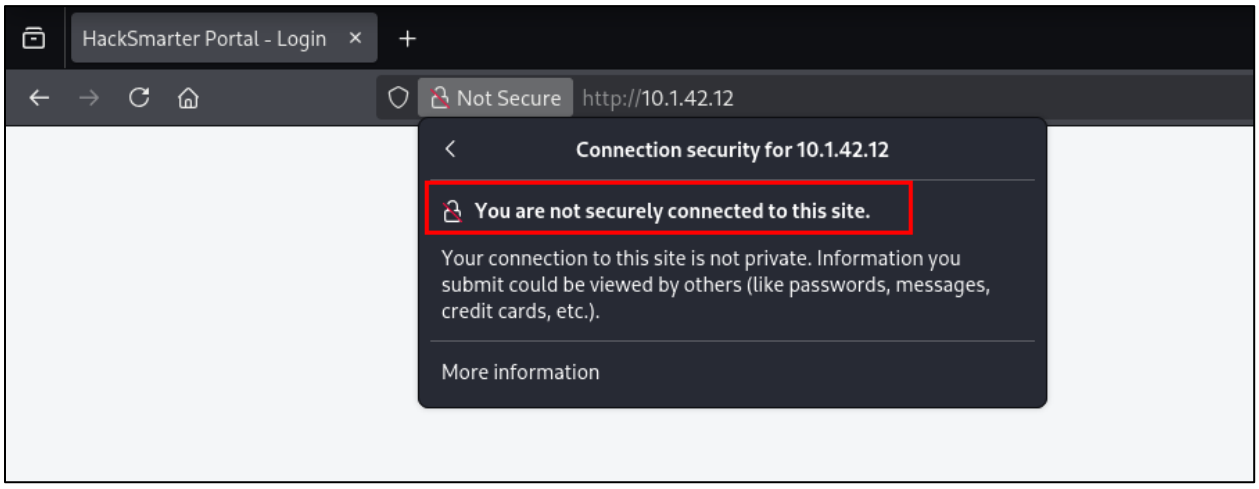


Figure 2: Hunter Login Portal – HTTP Only.

Remediation

Enable and enforce site-wide encryption.

Finding WPT-003: Lack of rate limiting on Login and Password Reset endpoints (Medium)

Description	The application does not enforce rate limits on the login and password reset (/reset) endpoints. This enables an attacker to submit unlimited login and password reset requests, aiding brute-force, password spraying, and enumeration attacks.
CVSS Score	5.3 (Medium) – CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Risk	Likelihood: High – Easily exploitable using automated tools. Impact: High – Can lead to account compromise and denial-of-service scenarios.
System	http://10.1.42.12
Tools Used	Caido
References	CWE - CWE-770: Allocation of Resources Without Limits or Throttling (4.19) CWE - CWE-799: Improper Control of Interaction Frequency (4.19)

Evidence / Steps to Reproduce

The TWZ team was able to submit multiple requests to the Login page and the Password Reset page (/reset) without getting rate limited. This allowed the team to perform username enumeration and successfully identified the valid user discussed in WPT-004.

ID	Payload 1	Status	Length	Round-trip Time (ms)
1	james	200	2160	607
2	john	200	2160	560
3	robert	200	2160	590
4	michael	200	2160	527
5	william	200	2160	547
6	david	200	2160	587
7	richard	200	2160	526
8	charles	200	2160	553
9	joseph	200	2160	597
10	thomas	200	2160	526
11	christopher	200	2160	518
12	daniel	200	2160	533
13	paul	200	2160	583
14	mark	200	2160	515
15	donald	200	2160	571
16	george	200	2160	524
17	kenneth	200	2160	590
18	steven	200	2160	602
19	edward	200	2160	580
20	brian	200	2160	532
21	ronald	200	2160	569
22	anthony	200	2160	564
23	kevin	200	2160	608
24	jason	200	2160	588
25	matthew	200	2160	551
26	gary	200	2160	571
27	timothy	200	2160	542
28	jose	200	2160	575
29	larry	200	2160	532
30	jeffrey	200	2160	574
31	frank	200	2160	566

301 requests

Reset preference

Figure 3: Caido Automate – Username Enumeration in Password Reset endpoint.

Remediation

Implement request rate limiting / Implement CAPTCHA.

Finding WPT-004: Username Enumeration via Password Reset response timing (Medium)

Description	During testing of the password reset functionality (/reset), the TWZ team discovered that the application replies differently depending on whether the specified username exists. Valid usernames might be reliably identified by sending several password reset requests and evaluating response times. This behavior enables an attacker identify real user accounts without authentication. Username enumeration greatly reduces the work necessary to carry out credential-based attacks like password spraying or brute-force attacks.
CVSS Score	5.3 (Medium) – CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Risk	Likelihood: High – Timing-based enumeration is easy to automate. Impact: Medium – Enables targeted attacks against valid accounts.
System	http://10.1.42.12/reset
Tools Used	Caido
References	Testing for Account Enumeration and Guessable User Account Time based username enumeration

Evidence / Steps to Reproduce

Using the usernames list provided by the OSINT analysts, the TWZ team was able to enumerate valid usernames by sending multiple requests to the password reset page (/reset) and analyzing the Response Time of the server.

One request stood out with a response time of 1362ms which suggest that the username used in that request is valid.

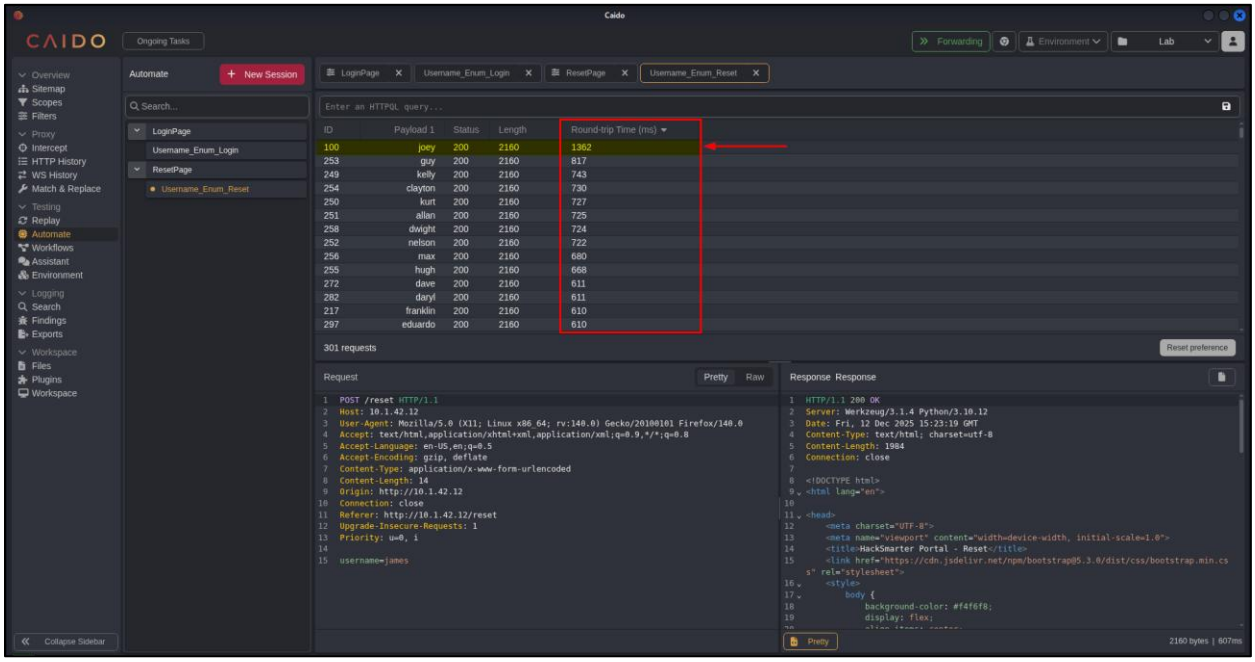


Figure 04: Caido Automate – Username Enumeration via Response Time Analysis

Remediation

Introduce delays to make response time for valid and invalid accounts the same. Implement server-side rate limiting on password reset requests (WPT-003).

Finding WPT-005: Hidden Endpoint Exposure via HTTP Response (Informational)

Description	Hunter's web server discloses detailed backend technology information through HTTP response headers. Specifically, the following information was observed: • Web Server: Werkzeug/3.1.4 • Programming Language: Python/3.10.12
CVSS Score	N/A
Risk	Likelihood: Medium – The system is accessible from the internet and is passively observable with any HTTP client or proxy tool. Impact: Low – This information could allow an attacker to fingerprint the web server to better target future exploit attempts.
System	http://10.1.42.12/
Tools Used	Caido, Manual Review
References	CWE-497: Exposure of Sensitive System Information to an Unauthorized Control Sphere OWASP Secure Headers Project: OWASP Secure Headers Project | OWASP Foundation Nginx: Module ngx_http_headers_module

Evidence / Steps to Reproduce

While analyzing HTTP response headers of the website, the TWZ team discovered that the web server reveals the server version running which is unnecessary and could allow attackers to fingerprint the server.

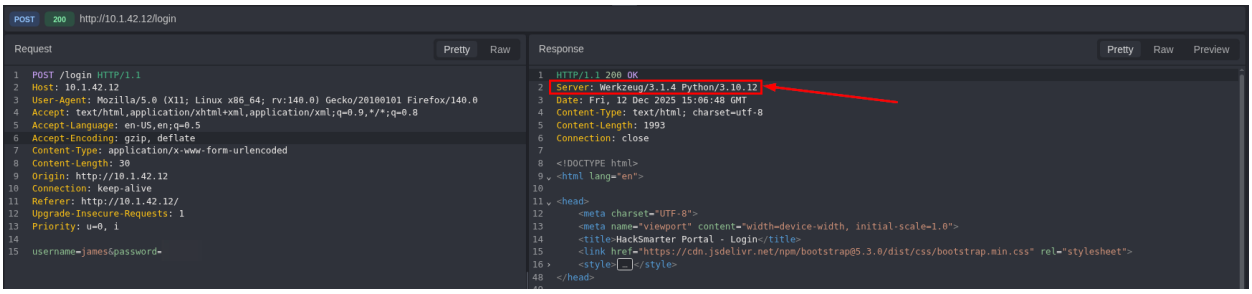


Figure 05: HTTP Response Header Showing the Server Version.

Remediation

TWZ recommends removing unnecessary information from HTTP response headers.

Conclusion & Next Steps

The web application penetration test discovered multiple medium to high severity vulnerabilities that, while not causing immediate compromise, collectively weakened the application's overall security posture. The detected concerns mostly concern information disclosure, authentication hardening gaps, and transport-layer security flaws.

Findings such as username enumeration via response timing, server version disclosure, lack of HTTPS enforcement, and lack of rate limiting provide attackers with valuable reconnaissance capabilities and increase the likelihood of successful credential-based attacks. Although no single vulnerability resulted in a direct system breach during this phase of testing, these flaws dramatically reduce the barrier to attackers performing targeted brute-force, password spraying, and man-in-the-middle attacks in real-world scenarios.

To improve the security posture of the application and reduce attack surface exposure, the following actions are recommended:

1. Harden Authentication and Account Recovery Mechanisms
    - Implement consistent responses and response timings for authentication and password reset workflows.
    - Enforce rate limiting, CAPTCHA challenges, and progressive delays to mitigate automated attacks.
    - Monitor authentication endpoints for abnormal or high-volume request patterns.
2. Secure Transport Layer Communications
    - Enforce HTTPS across the entire application.
    - Implement HTTP Strict Transport Security (HSTS) to prevent protocol downgrade attacks.
    - Redirect all HTTP traffic to HTTPS automatically.
3. Reduce Information Disclosure
    - Remove or obfuscate server and framework version information from HTTP response headers.
    - Regularly review application responses to ensure no unnecessary technical details are exposed.
4. Re-Assessment
    - Conduct a follow-up penetration test after remediation efforts are completed to validate the effectiveness of implemented controls and ensure no regressions were introduced.

LAST PAGE