OWASP Analysis of Skypunch Technology

performed by:

West Virginia University Department of Management Information Systems

A National Center of Excellence for Cybersecurity in Critical Infrastructure

in partnership with:

Joint Force Headquarters-Department of Defense Information Network and U.S. Cyber Command

under the direction of:

Dr. Mohammed Ahmad, Professor of Cybersecurity, Management Information Systems

August 2025

Table of Contents

Introduction

In collaboration with Skypunch Technology, a leading innovator in online election systems, our team of graduate Business Cybersecurity students from West Virginia University conducted a comprehensive application security audit aligned with the OWASP Application Security Verification Standard (ASVS) 4.0.3. This initiative is part of Skypunch's ongoing mission to meet federal security expectations and ensure the integrity, confidentiality, and availability of its voting infrastructure.

The primary objectives of this assessment were to:

- Evaluate the security posture of Skypunch's online voting platform.
- Verify compliance with OWASP ASVS and other relevant cybersecurity controls.
- Provide evidence-based recommendations to address risks and enhance resilience.

Under the guidance of Dr. Mohammad Ahmad and in consultation with Skypunch Technology President David Simms, the team developed a targeted scope and methodology. Testing was conducted using a combination of approaches, with Skypunch providing controlled access and documentation to support thorough evaluation. The audit focused on 66 OWASP ASVS controls, encompassing key areas such as authentication, session management, data protection, input validation, and API security.

Skypunch operates within a robust AWS environment utilizing IaaS, PaaS, and SaaS components, which demanded a layered security review. Various tools were employed to simulate real-world attack scenarios and validate the effectiveness of implemented defenses. Where applicable, controls were tested manually, and system responses were examined through browser inspection, API manipulation, and configuration review.

The findings from this audit affirm that Skypunch has implemented strong security controls across most areas tested. Our analysis underscores Skypunch's proactive security culture, demonstrated by its secure software development lifecycle, use of multi-factor authentication, effective session termination, and implementation of a Web Application Firewall (WAF). The audit serves not only as a verification tool but also as a strategic resource to guide future enhancements and align Skypunch with the highest standards of trust and transparency in election technology.

As cyber threats and election security concerns continue to rise, regular third-party assessments grounded in OWASP ASVS will be essential. This report provides a clear, actionable roadmap to strengthen Skypunch's already impressive security foundation and ensure its readiness for federal-level compliance.

Methodology

Our audit of Skypunch Technology was conducted using a standards based approach that followed the OWASP ASVS v4.0.3. This was selected because of Skypunch Technology's future goals of supporting high valued elections all the way to the federal level.

Control Identification and Scoping

The audit began with an introduction to the OWASP 4.0.3 controls and their relevance to Skypunch Technology. This was followed by a thorough review by team members of the control and selection of 12-15 controls to audit per person.

Manual Testing and Validation

Each control was manually tested through a variety of tools. Some auditors used F12 browser development tools, and others elected for tools such as Postman and Burpsuite. Specific cases included:

- Injecting spoofed authentication headers
- Inspecting HTTP response headers to attempt to find anti-caching directives
- Simulating failed authentication attempts to discover if secure logging and error handling was taking place

The use of these manual tests are supported with screenshots and traffic observations to validate the findings.

Policy Review and Control Verification

For cases where manual testing was not able to satisfy controls or the testing was limited because of platform constraints, such as backend logging visibility, the team requested and relied on review of official documentation, including the SkyPunch Web Application Security Policy. Controls were verified based on whether the policy could satisfy OWASP control objectives, in areas as these:

- Secure Software Development Lifecycle (SDLC)
- Session management and token protection
- Data retention and caching policies

Recommendations

Overall, the security posture at Skypunch is very strong. The vast majority of controls tested passed first try without the need to patch a hole. However, a handful of controls were either partially or completely failed. These controls should be given further attention in order to meet the requirement as written out by OWASP. Specifically, we recommend refining input validation rules, correcting redirect handling behavior, and configuring missing cookie security flags. Once these adjustments are made, the affected controls should be retested to verify that the changes resolved the underlying issues and bring those areas into alignment with OWASP ASVS expectations.

Furthermore, it is recommended that Skypunch develop and update internal security policies to reflect OWASP guidance where gaps were identified. Next, it is also advised that Skypunch establish a recurring review process (both on a time interval and after any major configuration changes) using the OWASP framework. Skypunch should also consider integrating an automated scanning tool to catch regressions as soon as possible. Finally, Skypunch should be encouraged to ensure evidence and change logs are available to support future attestations of security posture.

Controls Descriptions

V1 Architecture, Design and Threat Modeling

Chapter: Architecture, Design and Threat Modeling
Section: Secure Development Lifecycle
Section ID: V1.1
Control ID: V1.1.1
Level: 2 and 3
CWE: N/A
Control Description:
Verify the use of a secure software development lifecycle that addresses security in all stages of development.
Status:
✅
Notes:
SkyPunch Technologies has documented its Secure Software Development Lifecycle (SDLC) process as outlined in the "Web Application Security Policy – July 2025." The policy uses security considerations throughout all phases of development, from planning to deployment and then finally maintenance. It references OWASP resources such as the Code Review Guide and the Cheat Sheet Series, and mandates the use of Amazon Inspector to scan Lambda functions for vulnerabilities. These practices demonstrate a structured and security-focused approach to software development, satisfying the requirements of this control.

Chapter: Architecture, Design and Threat Modeling
Section: Access Control Architecture
Section ID: V1.4
Control ID: V1.4.4
Level: 2 and 3
CWE: 284
Control Description:
Verify the application uses a single and well-vetted access control mechanism for accessing protected data and resources. All requests must pass through this single mechanism to avoid copy and paste or insecure alternative paths. [(C7) https://top10proactive.owasp.org/]
Status:

✅

Notes:

I examined the cfm files for the main, my account, and Skyguard pages. When users log in to or create an account on ElectionsOnline, the application uses a single access control mechanism called “login-form_v2.cfm,” which contains the code required for authentication and user database entry. The application processes tasks such as verifying username existence, checking user roles (e.g., administrator), and logging out through a custom tag named “<CF_login-form_v2” with the same arguments passed through it, as can be shown below:

```
<CF_login-form_v2
	username = "Email:"
	username_field_type = "email"
	usernamelength="100"
	passwordlength="32"
	tableWidth="50"
	pleaseLogin="Please login."
	loginFailed="Login failed. Please try again."
	submitButton="Submit"
	languages="en"
autocomplete="on">
```

Chapter: Architecture, Design and Threat Modeling
Section: Access Control Architecture
Section ID: V1.4
Control ID: V1.4.5
Level: 2 and 3
CWE: 275

Control Description:

Verify that attribute or feature-based access control is used whereby the code checks the user's authorization for a feature/data item rather than just their role. Permissions should still be allocated using roles. [(C7) https://top10proactive.owasp.org/]

Status:

✅

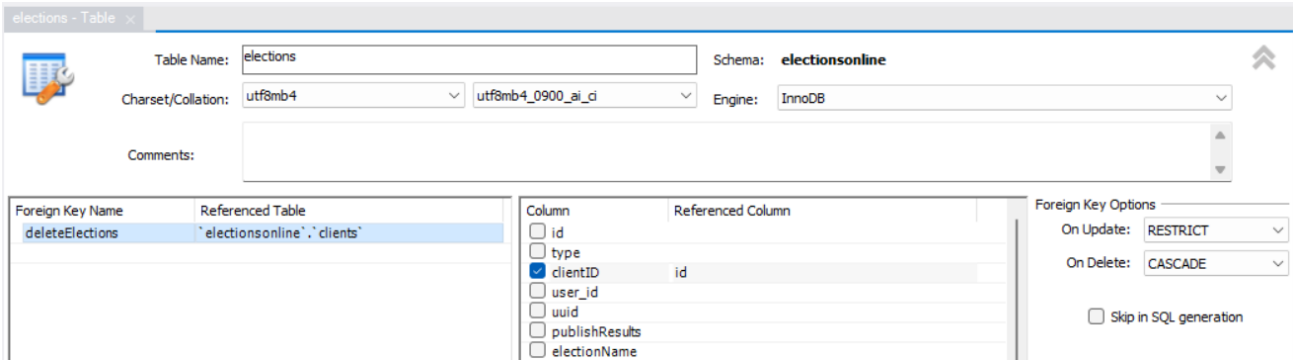
Notes:

The screenshots below are taken from a SQL database provided by David, specifically the "elections" table. This table includes an "elections.clientID" column, which shows a relationship with the "clients.id" column. As a result, the primary key in the "clients" table is referenced in a one-to-many relationship, allowing it to appear multiple times in the clientID field of the elections table. The system subsequently restricts certain features based on user roles (such as admin, public sector, or private sector), ensuring that attribute or feature-based access controls are applied.

Table: elections

Columns:

id	mediumint UN AI PK
type	char(3)
clientID	mediumint UN
user_id	int UN
uuid	char(35)
publishResults	tinyint(1)
electionName	varchar(75)
sendAnnouncement	tinyint(1)
fromName	varchar(60)
emailCount	tinyint UN
startDate	date
reminderDate	date



Chapter: Architecture, Design and Threat Modeling
Section: Configuration Architecture
Section ID: V1.14
Control ID: V1.14.4
Level: 2 and 3
CWE: N/A
Control Description:
Verify that the build pipeline contains a build step to automatically build and verify the secure deployment of the application, particularly if the application infrastructure is software defined, such as cloud environment build scripts.

Status:
✅

Notes:

Although not in a traditional CI/CD environment, the deployment process includes multiple safeguards aligned with the intent of this control. Secure credential storage, version control, review practices, and environment separation demonstrate secure deployment handling.

Chapter: Architecture, Design and Threat Modeling
Section: Configuration Architecture
Section ID: V1.14
Control ID: V1.14.5
Level: 2 and 3
CWE: 265
Control Description:

Verify that application deployments adequately sandbox, containerize and/or isolate at the network level to delay and deter attackers from attacking other applications, especially when they are performing sensitive or dangerous actions such as deserialization.

Status:

✅

Notes:

The application is deployed in an isolated VPC with no other services located within it. This setup effectively meets the intent of sandboxing and network isolation by limiting exposure and reducing the risk of cross-application compromise.

V2 Authentication

Chapter: OWASP ASVS Chapter VV2
Section: V2.1
Control ID: V2.1.1
Level: 2 and 3
CWE: 307

Control Description:

Control verifies the implementation and effectiveness of a secure password policy.

Status:

✅

Notes:

Testing Methodology:

Conducted an interview with the lead developer and maintainer of the system, David Simms. Conducted manual testing and inspection using browser tools (DevTools, network inspection). Reviewed application behavior to determine the presence and correctness of security control.

Confirmed through discussion with David Simms (Skypunch Technology) that user passwords cannot be changed. Passwords are system-generated using strong, random strings that meet complexity standards. Manual password creation is not permitted, ensuring uniform policy enforcement. Passwords are system-generated and meet NIST guidelines for complexity, length, and randomness. User-defined passwords are not permitted, reducing the risk of weak credentials.

Chapter: Authentication
Section: General Authenticator Security

Control ID: V2.2.1
Level: 2 and 3
CWE: 307

Control Description:

Control verifies the implementation and effectiveness of preventing brute-force logins.

Status:

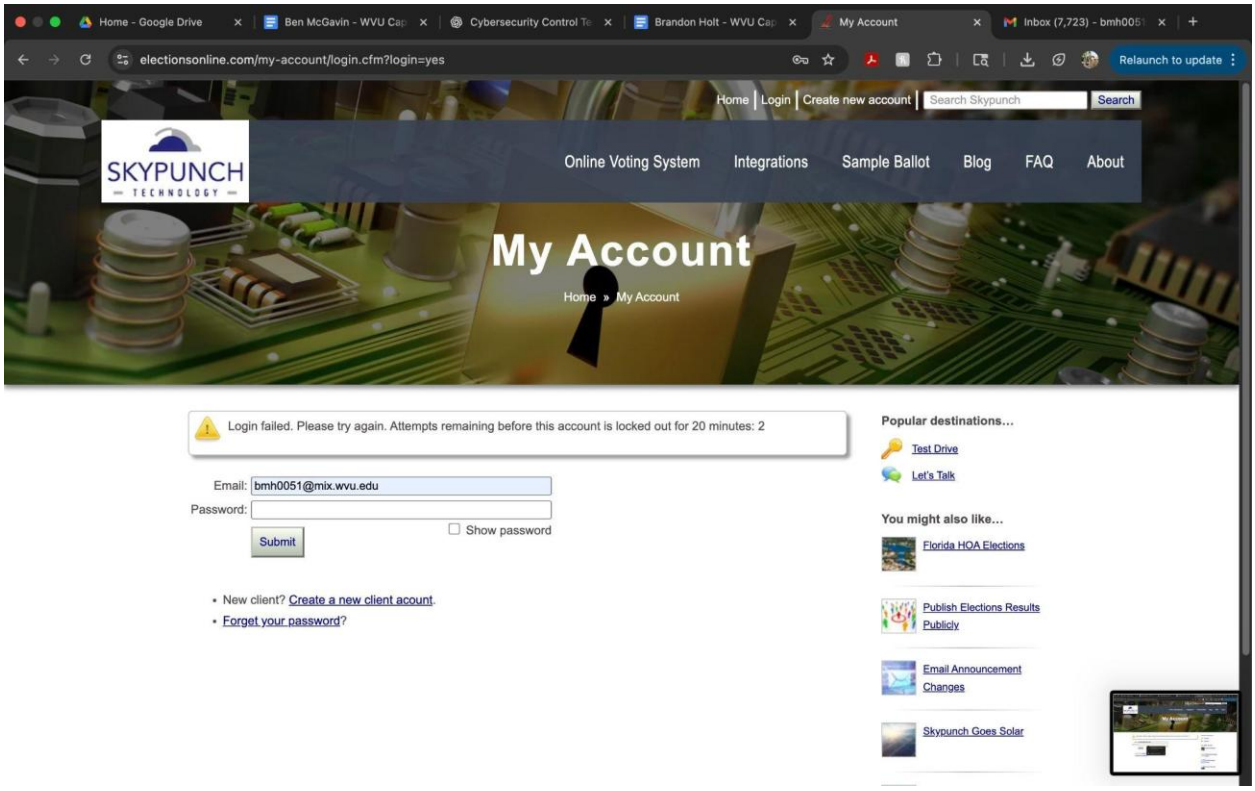
✅

Notes:

Testing Methodology:

Conducted manual testing and inspection using browser tools (DevTools, network inspection). Reviewed application behavior to determine the presence and correctness of security control.

Noted 3 failed attempts before password lockout.



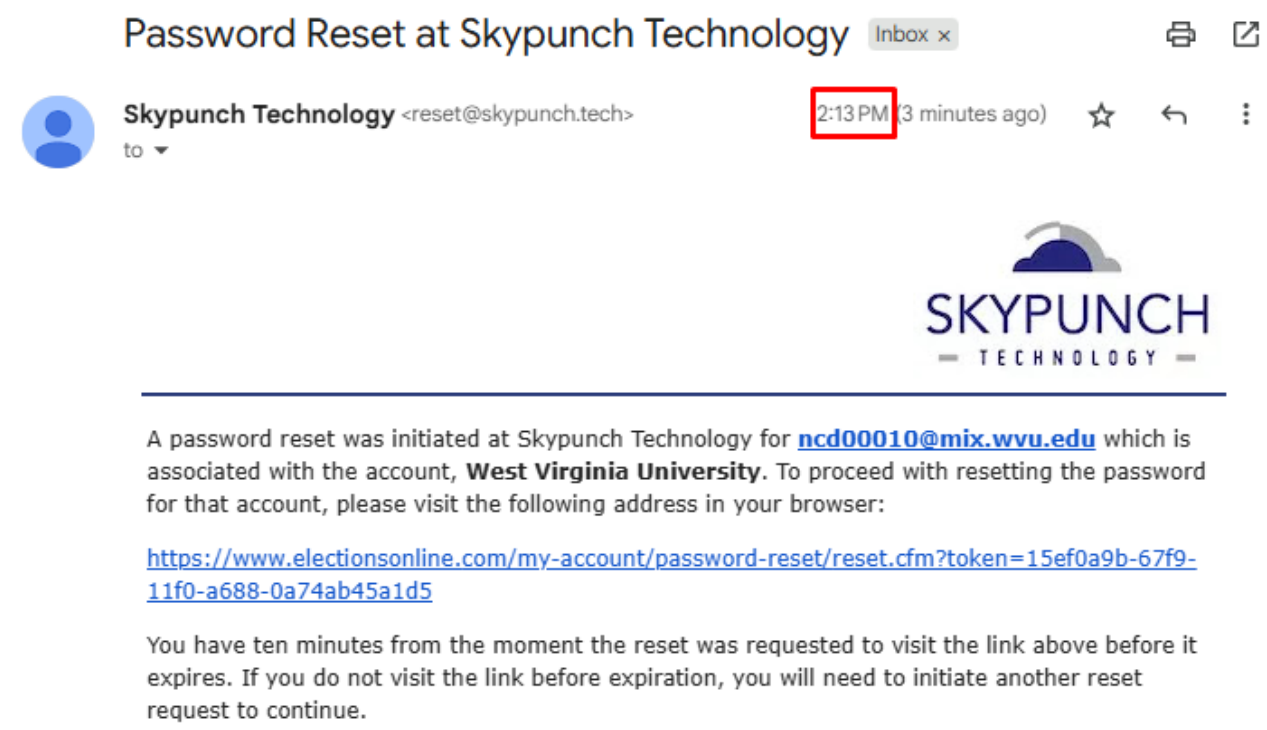
Chapter: Authentication
Section: General Authenticator Security
Section ID: V2.2

Control ID: V2.2.4
Level: 3
CWE: 308
Control Description:
Verify impersonation resistance against phishing, such as the use of multi-factor authentication, cryptographic devices with intent (such as connected keys with a push to authenticate), or at higher AAL levels, client-side certificates.

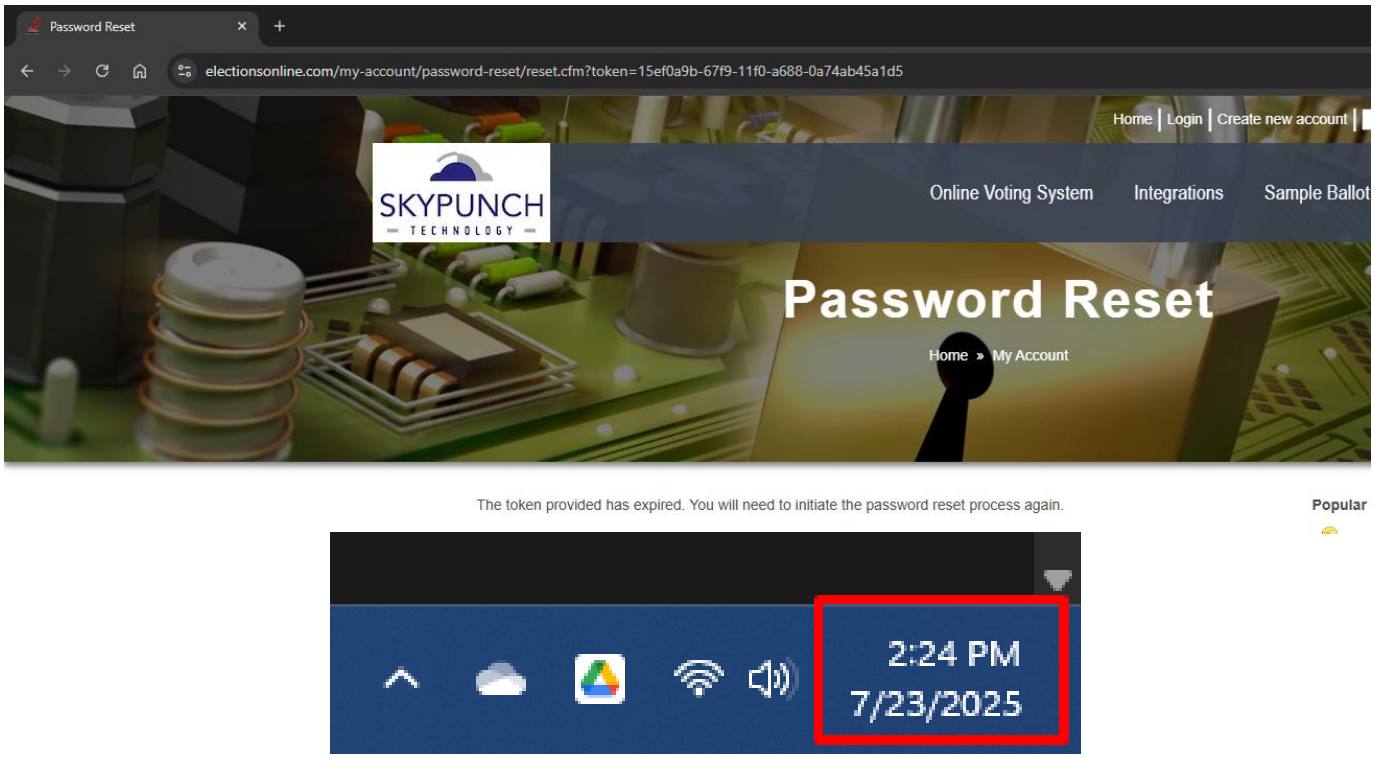
Status:
✅
Notes:
MFA is in place and requires TOTP codes, which are resistant to phishing attacks. The control was verified through testing and functioning as intended.

Chapter: Authentication
Section: General Authenticator Security
Section ID: V2.2
Control ID: V2.2.6
Level: 3
CWE: 308
Control Description:
Verify replay resistance through the mandated use of One-time Passwords (OTP) devices, cryptographic authenticators, or lookup codes.
Status:
✅
Notes:
Replay resistance was through the password reset process and verifying that the reset link in the email expires after 10 minutes. A screenshot of the initial timestamp of the

password reset link is provided below:



After 10 minutes, clicking the link confirms that the OTP token expires as demonstrated in this screenshot:



Chapter: Authentication
Section: Authenticator Lifecycle
Section ID: V2.3
Control ID: V2.3.3
Level: 2, 3
CWE: 287
Control Description:
Verify that renewal instructions are sent with sufficient time to renew time bound authenticators.
Status:

Notes:
Direct access to the login flow was not available for testing. Administrative confirmation was used to validate the implementation of MFA.

Chapter: Authentication
Section: Credential Recovery
Section ID: V2.5
Control ID: V2.5.2
Level: 1, 2, and 3
CWE: 640
Control Description:
Verify the use of a secure software development lifecycle that addresses security in all stages of development.

Status:


Notes:
I tested the password recovery workflow for the SkyPunch platform and confirmed that no password hints or knowledge-based authentication methods (e.g., secret questions) are currently being used. The recovery process involves a secure email token followed by mandatory two-factor authentication (2FA) to receive a temporary password. This ensures account recovery is resistant to guessing attacks and aligns with the control.

Chapter: Authentication
Section: Credential Recovery
Section ID: V2.5
Control ID: V2.5.3
Level: 1, 2, and 3

CWE: 640

Control Description:

Verify password credential recovery does not reveal the current password in any way.

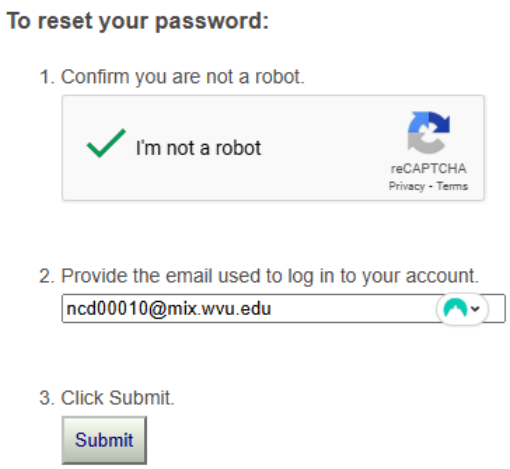
[(C6) https://top10proactive.owasp.org/]

Status:

Notes:

When going through the password recovery phase, the system does not reveal current password in reset password email, OTP, and anywhere else in the system up until the point of the generated new password (see screenshots below for the step-by-step process).

1. Captcha and account email

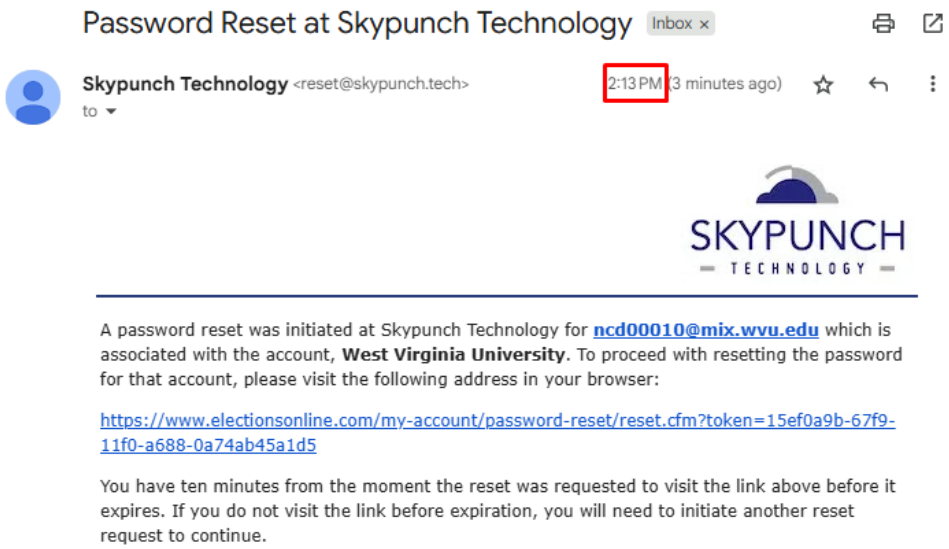


2. Match found email

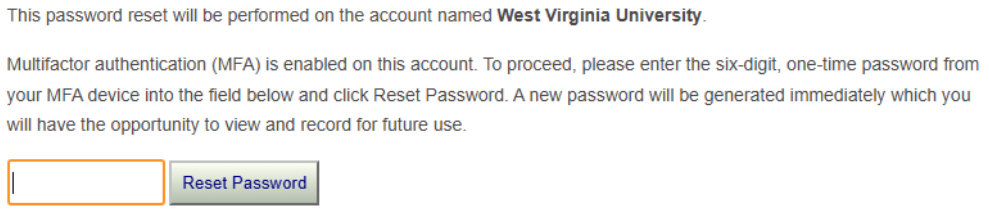
If a match is found for ncd00010@mix.wvu.edu:

- You may expect to receive an email momentarily containing a link for resetting the password for the associated account.
- You will have ten minutes to click that link before it expires.
- The email will come from reset@skypunch.tech. Check your spam folder if it does not appear in your inbox.
- Finally, if no email arrives, you may assume no match was found and try using another email.

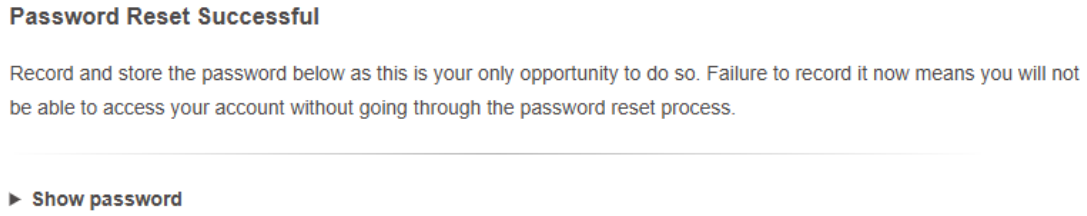
3. Reset email with reset link



4. MFA to confirm password reset



5. Password successfully reset message with drop down to record new generated password



Chapter: Authentication
Section: Credential Recovery
Control ID: V2.5.6
Level: 2 and 3
CWE: 640

Control Description:

Control verifies the implementation and effectiveness of password reset and recovery.

Status:

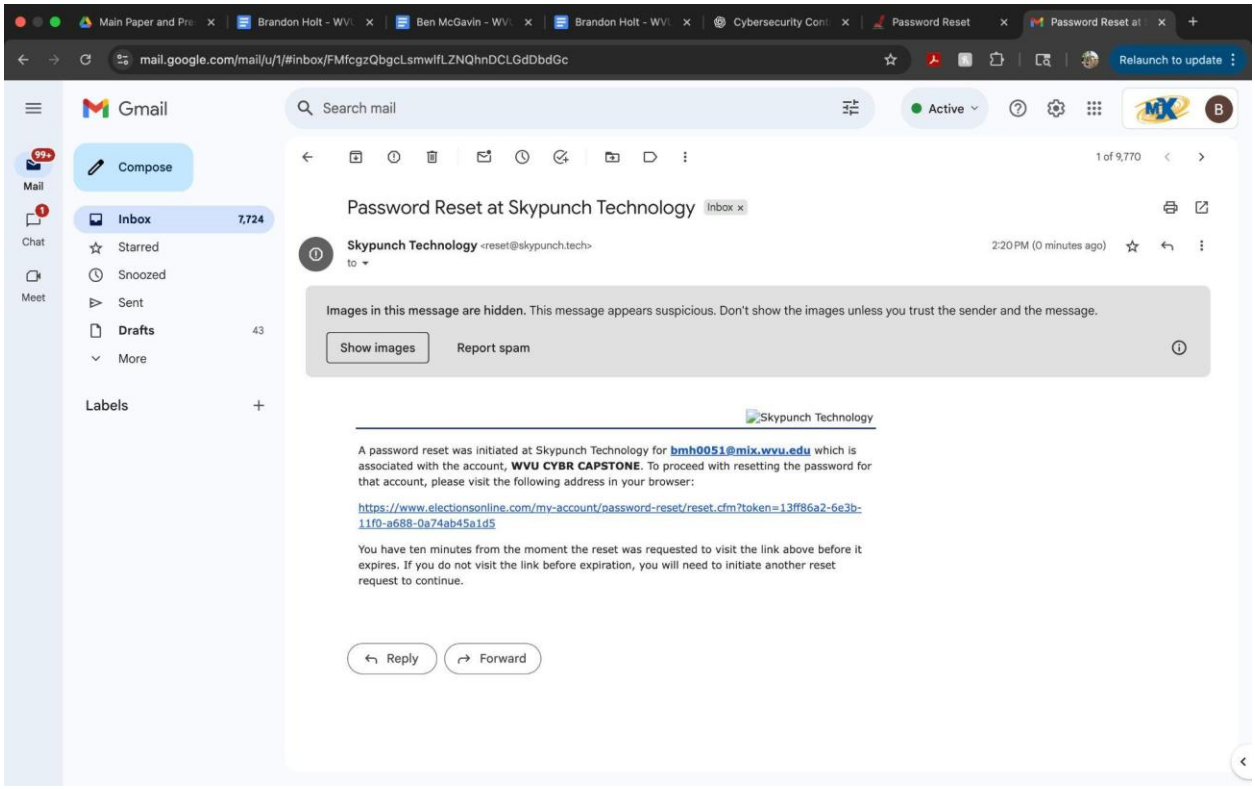
Notes:

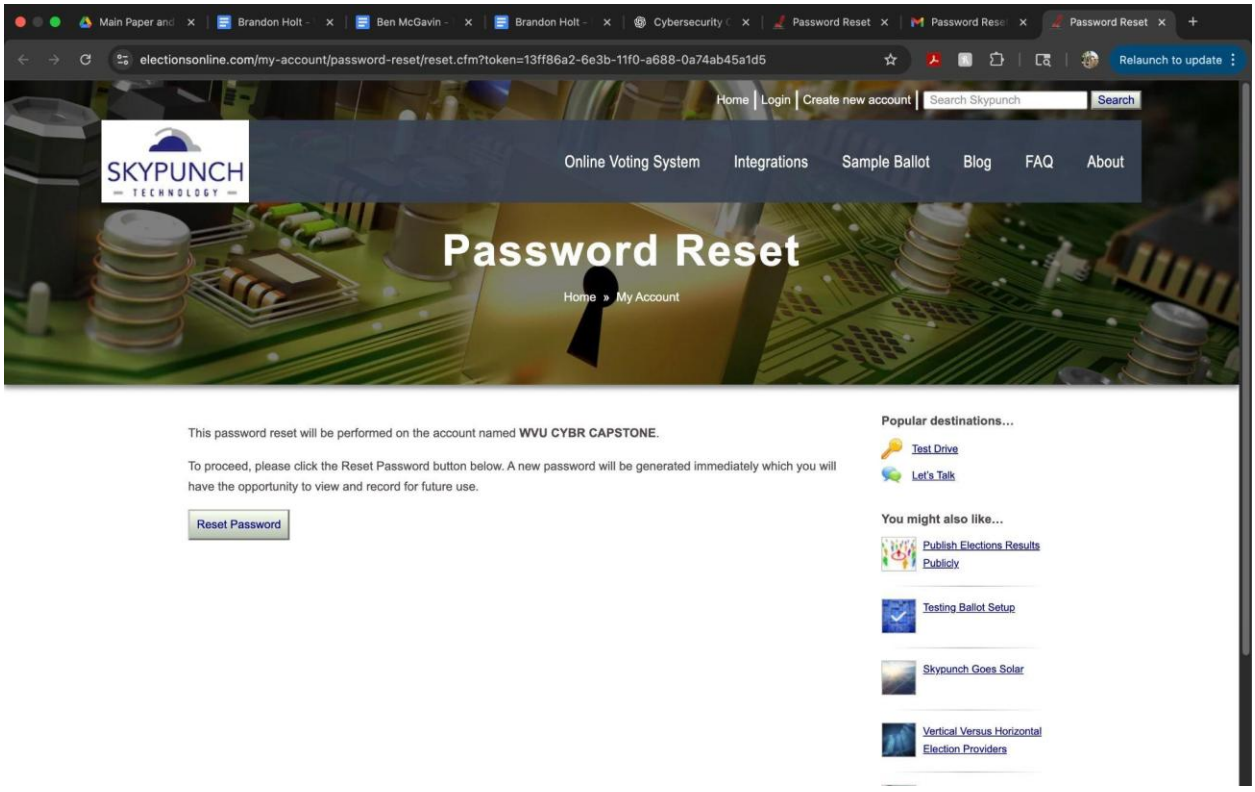
Testing Methodology:

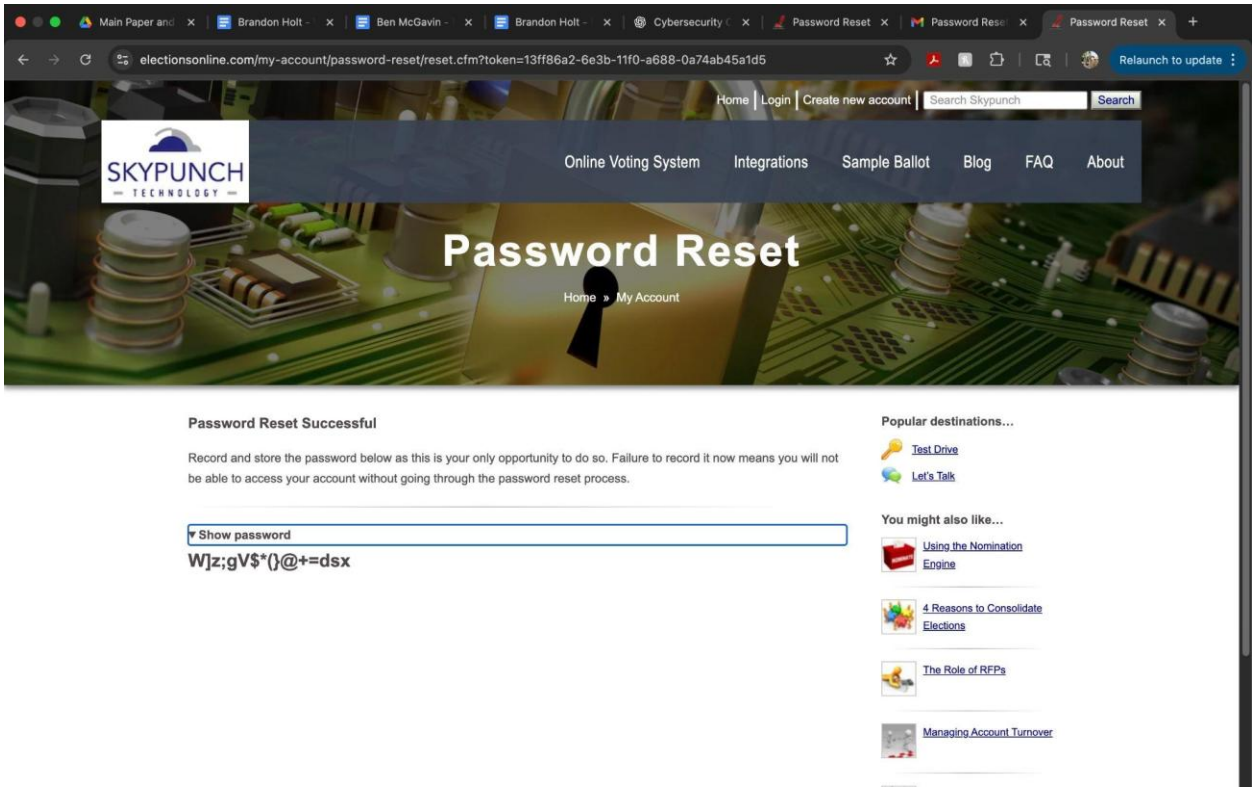
Conducted manual testing and inspection using browser tools (DevTools, network inspection). Reviewed application behavior to determine the presence and correctness of security control.

User receives an email link that is valid for 10 minutes. A new randomly generated password will be granted to the user.Figure: Password Reset Email Figure: Password Reset Link - Upon clicking “Reset Password” user is granted a new randomly generated password. Figure: User Granted New Password

Password reset feature generates secure random password; process completed successfully via email.







Chapter: Authentication
Section: Lookup Secrets
Section ID: V2.6
Control ID: V2.6.1
Level: 2 and 3
CWE: 640
Control Description:
Verify that lookup secrets can be used only once.

Status:
✅

Notes:
I tested the SkyPunch password recovery process and verified that email-based recovery tokens are single-use. After successfully using the token to log in and reset the password, I attempted to reuse the same account recovery link and then received the message:

“The token provided does not match any tokens on record.”
This confirms that lookup secrets are invalidated after use, satisfying the control.

Chapter: Authentication
Section: One Time Verifier

Control ID: V2.8.1
Level: 2 and 3
CWE: 613
Control Description:

Control verifies the implementation and effectiveness of multi-factor authentication (MFA).

Status:

✅ Pass

Notes:

Testing Methodology:

Conducted manual testing and inspection using browser tools (DevTools, network inspection). Reviewed application behavior to determine the presence and correctness of security control.

Multi-factor authentication is enforced using TOTP-compatible apps such as Google Authenticator and Authy. These OTPs expire after a short amount of time. The login process requires a one-time code in addition to username and password, providing strong resistance to unauthorized access.

MFA enforced using TOTP with Google Authenticator and Authy compatibility; implementation confirmed.

Chapter: OWASP ASVS Chapter VV2
Section: Cryptographic Verifier
Control ID: V2.9.3
Level: 2 and 3
CWE: 327

Control Description:

Control ensures that device-based authenticators, such as trusted devices, biometric methods, or mobile MFA apps, are used only as secondary factors in a multi-factor authentication flow. The goal is to prevent reliance on devices alone for authentication and to enforce reauthentication when access occurs from unrecognized or new devices.

Status:

✅

Notes:

Testing Methodology:

Conducted manual testing and behavioral observation across desktop and mobile devices. Used Chrome DevTools and browser inspection to track session state and login behavior. Attempted login from a secondary device (mobile phone) to observe authentication response. Evaluated whether the application enforced reauthentication or additional verification from the new device.

Logging in from a new device triggered mandatory reauthentication via MFA, confirming device-based verification is enforced. This behavior ensures added protection when accessing accounts from unfamiliar devices.

When logging in from a new device (mobile phone), the system prompted for reauthentication using time-based one-time password (TOTP) via a mobile authenticator app. No biometric authentication was used, but the presence of device-specific reauthentication aligns with the spirit of this control: enforcing a secondary factor when trust cannot be assumed. The system did not offer an option to "remember this device," which minimizes persistent trust risk.

V3 Session Management

Chapter: Session Management
Section: Session Termination
Control ID: V3.3.1
Level: 2 and 3
CWE: 613

Control Description:

Control ensures that user sessions are invalidated automatically after a defined period of inactivity. Proper session timeout behavior is essential to minimize the risk of unauthorized access if a user walks away from a logged-in session.

Status:

Notes:

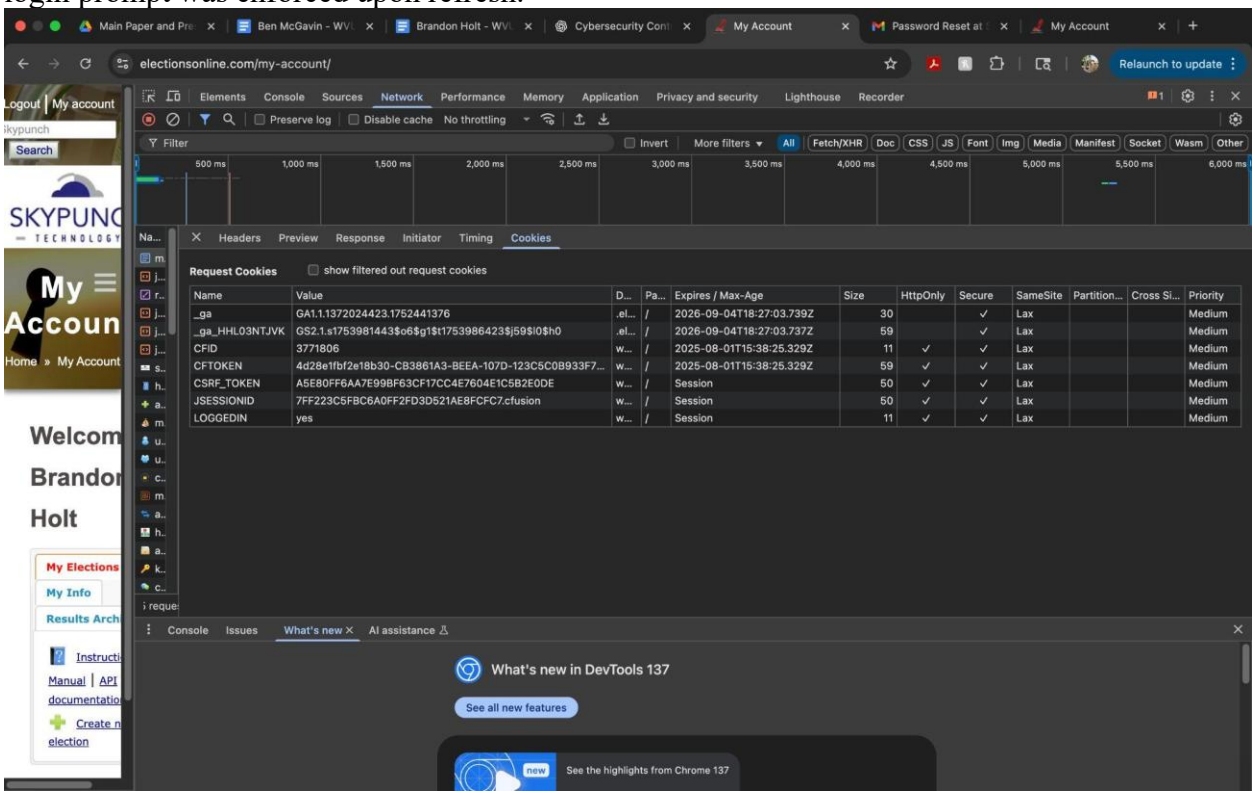
Testing Methodology:

Logged into the SkyPunch Technology user dashboard and remained idle for over 25 minutes. After

extended inactivity, refreshed the page and was automatically redirected to the login screen, confirming the session had expired. Used Chrome DevTools to inspect cookies under the Application and Network tabs. Observed that the key session cookies (JSESSIONID, CSRF_TOKEN, LOGGEDIN) were all set with Session expiration values, meaning they automatically clear when the session ends or the browser is closed.

Session timeout enforced properly and session cookies expire as expected. Session invalidation after inactivity is working as intended. Supporting Evidence Figure: Cookies Expiration Configuration

Session timeout behavior was correctly triggered after prolonged inactivity. Session cookies were configured to expire with the session (Session max-age), and flagged with both Secure and HttpOnly attributes for added protection. No residual access was possible after timeout, and the login prompt was enforced upon refresh.



Chapter: Session Management
Section: Session Termination
Section ID: V3.3
Control ID: V3.3.4
Level: 2 and 3
CWE: 613
Control Description:

Verify that users are able to view and (having re-entered login credentials) log out of any or all currently active sessions and devices.

Status:

✅

Notes:

After confirming on different computers, if the user doesn't respond for a while and attempts to click on any of the account-specific features, the user is prompted to log back in.

`

Chapter: Session Management

Section: Cookie-based Session Management

Section ID: V3.4

Control ID: V3.4.1

Level: 1, 2, and 3

CWE: 614

Control Description:

Verify that cookie-based session tokens have the 'Secure' attribute set.

Status:

✅

Notes:

The session cookie is marked secure.

Chapter: Session Management

Section: Cookie-based Session Management

Control ID: V3.4.2

Level: 2 and 3

CWE: 1004

Control Description:

Control verifies the implementation and effectiveness of use of secure cookies for session data. Cookies used for session tracking must include the Secure flag (to restrict transmission over HTTPS), HttpOnly flag (to prevent JavaScript access), and a properly configured SameSite attribute (to mitigate cross-site request forgery). These flags collectively help ensure that session data remains confidential and is only accessible under controlled circumstances.

Status:

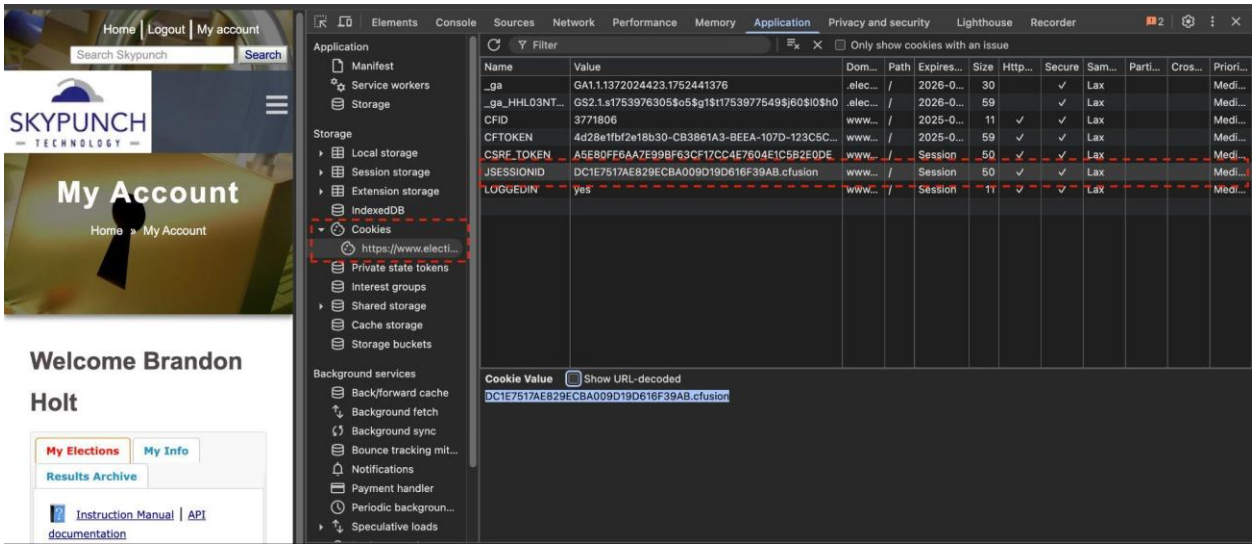
Notes:

Testing Methodology:

Inspected the session cookies set by the application using Chrome DevTools. Navigated to the Application > Cookies section and reviewed attributes set on the JSESSIONID cookie. Evaluated whether the required security attributes (Secure, HttpOnly, and SameSite) were present and enforced. Verified that cookies were only set over HTTPS and did not transmit in plaintext over HTTP.

The session cookie is properly configured with Secure, HttpOnly, and SameSite attributes. It is transmitted only over HTTPS and is not accessible via client-side scripts, meeting secure session handling requirements. Supporting Evidence

The session cookie (JSESSIONID=DC1E7517AE829ECBA009D19D616F39AB.cfusion) is set by a ColdFusion application server and includes the Secure, HttpOnly, and SameSite=Lax attributes. This configuration protects the session from client-side access, ensures transport over HTTPS, and defends against CSRF in standard scenarios. The HttpOnly attribute is present, which restricts access to the cookie from client-side scripts, reducing XSS-related risk. The SameSite attribute is set to Lax, providing protection against cross-site request forgery (CSRF) in most cases while still allowing safe cross-origin navigation. The cookie's expiration is set to “Session,” meaning it is removed when the browser is closed, further reducing exposure risk.



Chapter: Session Management
Section: V3.4 Cookie-based Session Management
Section ID: V3.4
Control ID: V3.4.3
Level: 1, 2, and 3
CWE: 16

Control Description:

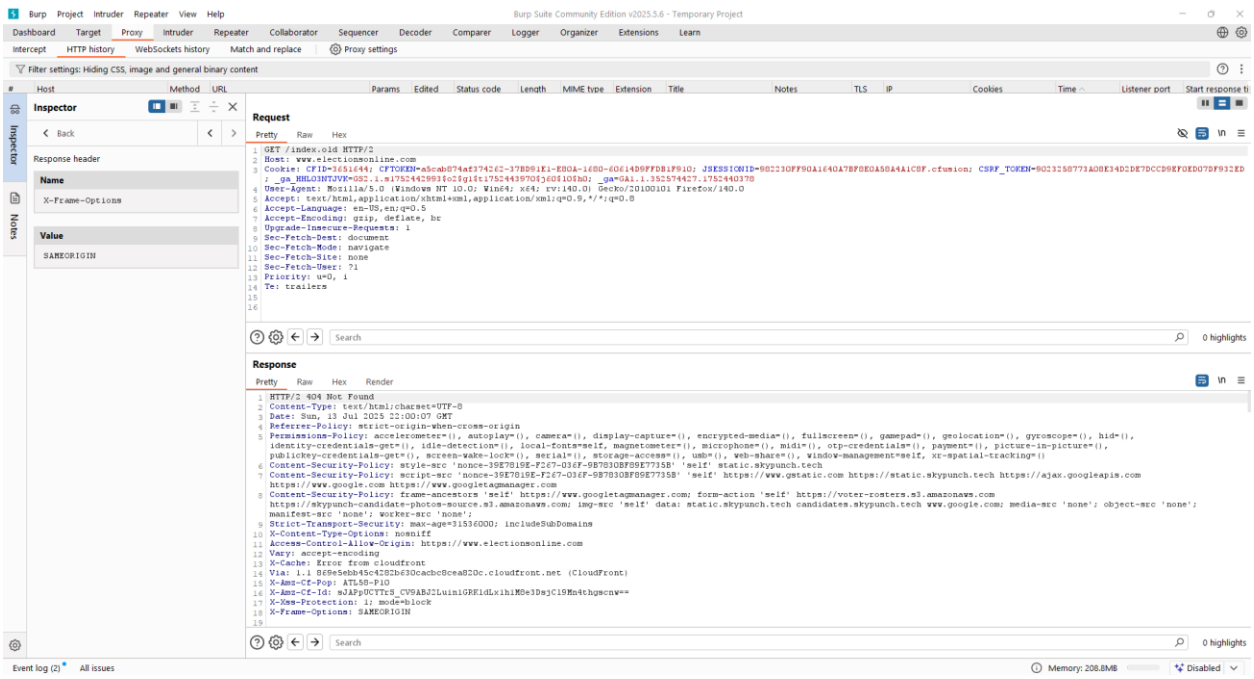
Verify that cookie-based session tokens utilize the 'SameSite' attribute to limit exposure to cross-site request forgery attacks.

Status:

N/A

Notes:

Used browser developer tools and Burp to inspect cookies. Most cookie flags are present, but SameSite is missing.



V4 Access Control

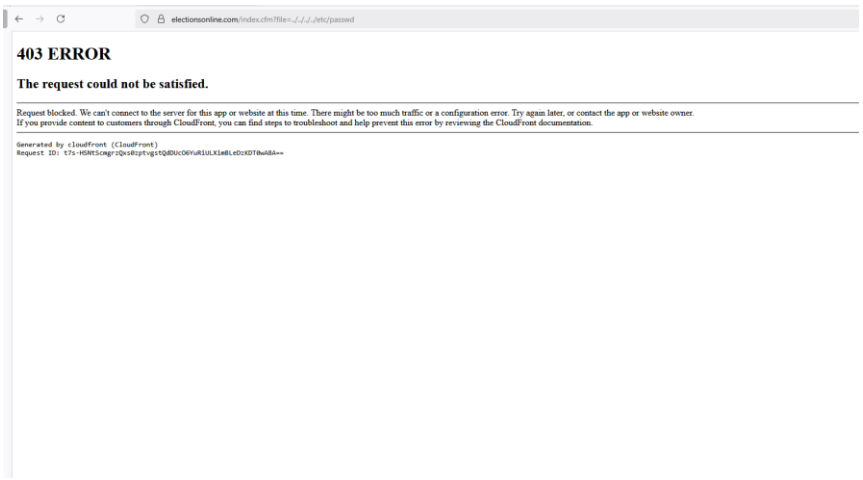
Chapter: Access Control
Section: Operation Level Access Control
Section ID: V4.2
Control ID: V4.2.1
Level: 1, 2, and 3
CWE: 639
Control Description:
Verify that sensitive data and APIs are protected against Insecure Direct Object Reference (IDOR) attacks targeting creation, reading, updating and deletion of records, such as creating or updating someone else's record, viewing everyone's records, or deleting all records.

Status:
✅

Notes:

Used Burp Repeater to submit:

/view?file=../../../../etc/passwd. Server responded with 403. Traversal protection is working.



Chapter: Access Control
Section: Administrative Interfaces
Section ID: V4.3
Control ID: V4.3.1
Level: 2 and 3
CWE: 287

Control Description:

Verify administrative interfaces use appropriate multi-factor authentication to prevent unauthorized use.

Status:

Notes:

I created an election account on the SkyPunch platform that had capabilities of generating an election and verified that multi-factor authentication (MFA) is enforced during login. The platform requires the use of the Duo Mobile app for time-based one-time password (TOTP) authentication, with a new 6-digit code generated every 30 seconds. This confirms that the administrative interface is protected by MFA, fully satisfying this control.

V5 Validation, Sanitization and Encoding

Chapter: Validation, Sanitization and Encoding
Section: Input Validation
Section ID: V5.1
Control ID: V5.1.4
Level: 1, 2, and 3
CWE: 20

Control Description:

Verify that structured data is strongly typed and validated against a defined schema including allowed characters, length and pattern (e.g. credit card numbers, e-mail addresses, telephone numbers, or validating that two related fields are reasonable, such as checking that suburb and zip/postcode match). [(C5) https://top10proactive.owasp.org/]

Status:

✗

Notes:

According to Sections 4.1 and 4.3 of the CISA report prepared for Skypunch, the application demonstrates robust performance in addressing vulnerabilities, achieving a 98% vulnerability resolution rate with few unresolved issues and no exposure of sensitive information. There was no evidence of improper input validation except in cases related to broken access control; however, some input fields (such as phone numbers and names) do not restrict character types, contributing to the identification of structured data that could lead to injection attacks (see screenshots for good and bad examples). It is recommended to address this issue and resolve for next year's analysis.

4.1 Report Card

WAS REPORT CARD

Low
Security Risk

2
Total Webapps

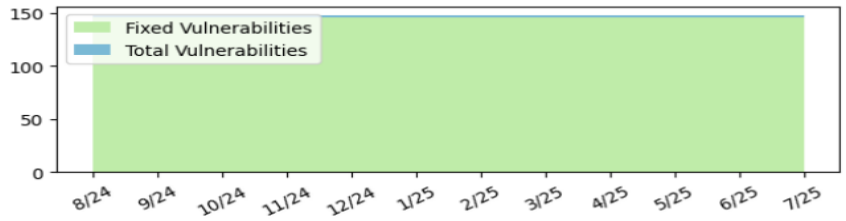
66
Information Gathered

HIGH LEVEL FINDINGS

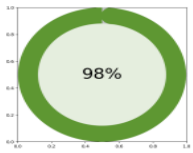
Scan Date: 02 Jul 2025

ACTIVE VULNERABILITIES	NEW VULNERABILITIES
2	0
REOPENED VULNERABILITIES	**SENSITIVE CONTENT**
0	0

TRACKING VULNERABILITIES: FOUND VS. FIXED



REDMEDIATION METRICS

% VULNERABILITIES FIXED TO DATE	MAX AGE OF ACTIVE CRITICALS (DAYS)	MAX AGE OF ACTIVE URGENTS (DAYS)
	0	0

4.3 Vulnerabilities by OWASP Category

The following graph lists total vulnerabilities based on OWASP categories. For more information about OWASP, see Appendix B: FAQ.

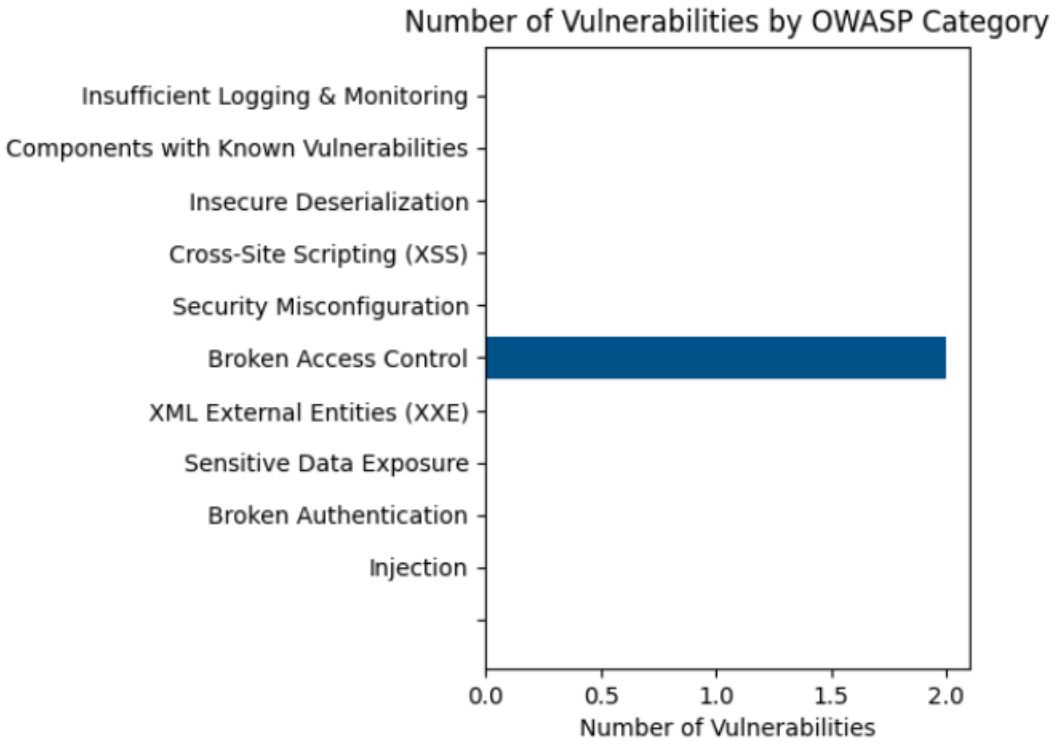
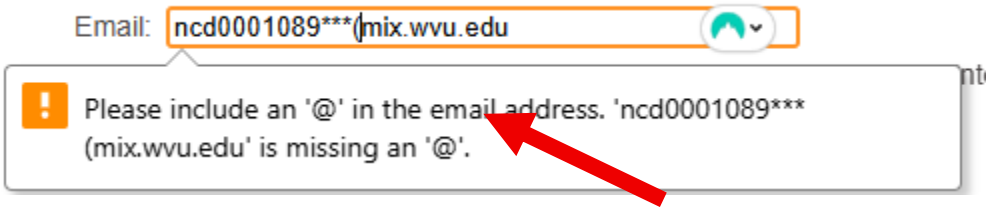


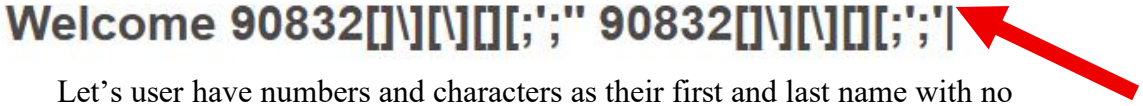
Figure 2: Total Vulnerabilities by OWASP

Good example:



- Must include @ in email address

Bad example:



- Let's user have numbers and characters as their first and last name with no restrictions

- Having letters and characters as phone number with no restrictions

Chapter: Validation, Sanitization and Encoding
Section: Input Validation
Section ID: V5.1
Control ID: V5.1.5
Level: 1, 2, and 3
CWE: 601

Control Description:

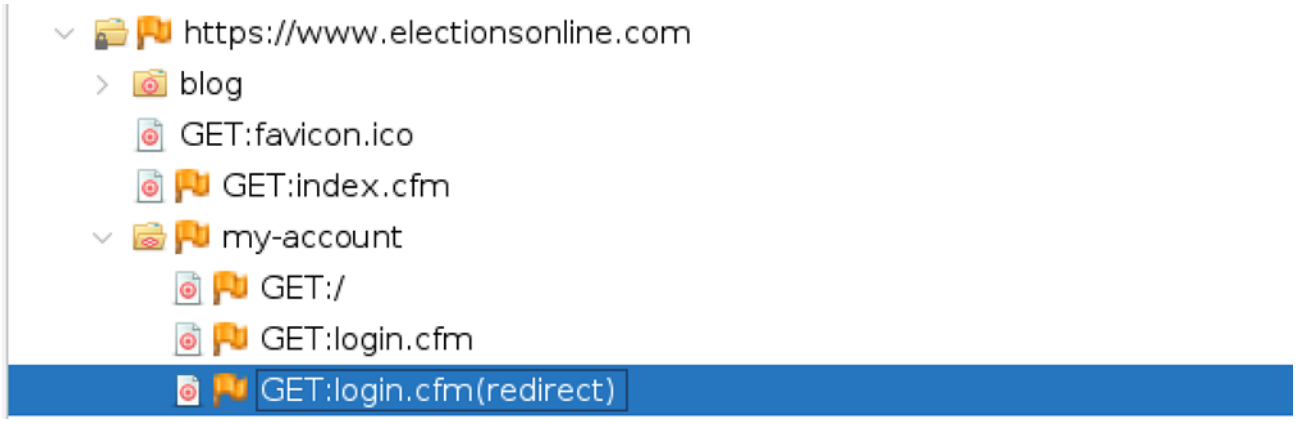
Verify that URL redirects and forwards only allow destinations which appear on an allow list, or show a warning when redirecting to potentially untrusted content.

Status:

Notes:

To test URL redirects and forwards, I used OWASP ZAP and performed a Manual Explore. To make it easier to test for URL redirects, I performed an initial simple redirect example from the browser

https://electionsonline.com/login.cfm?redirect=https://google.com.
Navigating through the Sites directory reveals a “GET:login.cfm(redirect)” site as can be seen below:



Setting this as my scope and performing an active scan with the default settings, the system provides mostly status codes of 200 (see screenshot below), but these do not get redirected to untrusted content—they simply stay on the login page. The other status codes shown also deny access to untrusted redirects.

ID	Req. Timestamp	Resp. Timestamp	Method	URL	Code	Reason
3,860	8/1/25, 6:46:15 PM	8/1/25, 6:46:15 PM	GET	https://www.electionsonline.com/my-account/login.cfm?redirect=www.goo...	200	OK
3,861	8/1/25, 6:46:15 PM	8/1/25, 6:46:15 PM	GET	https://www.electionsonline.com/my-account/login.cfm?redirect=www.goo...	200	OK
3,862	8/1/25, 6:46:15 PM	8/1/25, 6:46:15 PM	GET	https://www.electionsonline.com/my-account/login.cfm?redirect=www.goo...	200	OK
3,863	8/1/25, 6:46:15 PM	8/1/25, 6:46:15 PM	GET	https://www.electionsonline.com/my-account/login.cfm?redirect=www.goo...	200	OK
3,865	8/1/25, 6:46:15 PM	8/1/25, 6:46:16 PM	GET	https://www.electionsonline.com/my-account/login.cfm?redirect=www.goo...	200	OK
3,866	8/1/25, 6:46:16 PM	8/1/25, 6:46:16 PM	GET	https://www.electionsonline.com/WEB-INF/web.xml	403	Forbidden
3,867	8/1/25, 6:46:16 PM	8/1/25, 6:46:16 PM	GET	https://www.electionsonline.com/WEB-INF/applicationContext.xml	403	Forbidden
3,868	8/1/25, 6:46:16 PM	8/1/25, 6:46:16 PM	GET	https://www.electionsonline.com/WEB-INF/classes/4/01.class	404	Not Found
3,872	8/1/25, 6:46:16 PM	8/1/25, 6:46:16 PM	GET	https://www.electionsonline.com/WEB-INF/classes/www/w3/org.class	404	Not Found
3,873	8/1/25, 6:46:16 PM	8/1/25, 6:46:16 PM	GET	https://www.electionsonline.com/my-account/login.cfm?-s	200	OK
3,874	8/1/25, 6:46:16 PM	8/1/25, 6:46:16 PM	GET	https://www.electionsonline.com/WEB-INF/classes/loose/dtd.class	404	Not Found
3,875	8/1/25, 6:46:16 PM	8/1/25, 6:46:16 PM	POST	https://www.electionsonline.com/my-account/login.cfm?-d+allow_url_includ...	403	Forbidden
3,878	8/1/25, 6:46:16 PM	8/1/25, 6:46:16 PM	POST	https://www.electionsonline.com/my-account/login.cfm?-d+allow_url_includ...	403	Forbidden
3,882	8/1/25, 6:46:17 PM	8/1/25, 6:46:17 PM	GET	https://www.electionsonline.com/my-account/login.cfm?redirect=7455935...	200	OK
3,883	8/1/25, 6:46:17 PM	8/1/25, 6:46:17 PM	GET	https://www.electionsonline.com/my-account/login.cfm?redirect=https%3A...	200	OK
3,884	8/1/25, 6:46:17 PM	8/1/25, 6:46:17 PM	GET	https://www.electionsonline.com/my-account/login.cfm?redirect=https%3A...	200	OK
3,886	8/1/25, 6:46:17 PM	8/1/25, 6:46:17 PM	GET	https://www.electionsonline.com/my-account/login.cfm?redirect=5%3BURL...	200	OK
3,887	8/1/25, 6:46:17 PM	8/1/25, 6:46:17 PM	GET	https://www.electionsonline.com/my-account/login.cfm?redirect=URL%3D%...	200	OK
3,889	8/1/25, 6:46:17 PM	8/1/25, 6:46:17 PM	GET	https://www.electionsonline.com/my-account/login.cfm?redirect=https%3A...	200	OK
3,890	8/1/25, 6:46:17 PM	8/1/25, 6:46:17 PM	GET	https://www.electionsonline.com/my-account/login.cfm?redirect=5%3BURL...	200	OK
3,893	8/1/25, 6:46:17 PM	8/1/25, 6:46:17 PM	GET	https://www.electionsonline.com/my-account/login.cfm?redirect=https%3A...	200	OK
3,895	8/1/25, 6:46:17 PM	8/1/25, 6:46:18 PM	GET	https://www.electionsonline.com/my-account/login.cfm?redirect=http%3A%...	200	OK

Chapter: Validation, Sanitization and Encoding
Section: Input Validation
Control ID: V5.1.3
Level: 2 and 3
CWE: 20

Control Description:

Control ensures that input validation is enforced on a trusted service layer, such as the backend or application’s core logic tier. Relying on client-side or distributed validation introduces risk, while central enforcement guarantees consistent and secure input handling. The control helps mitigate injection attacks, inconsistent validation behavior, and input bypasses across application interfaces.

Status:

✅

Notes:

Testing Methodology:

Conducted manual black-box testing on multiple input vectors across the application, including the search bar, login form, and password reset flow. Submitted a range of malformed and invalid input data: Empty fields Script tags (<script>alert(1)</script>) Special characters Excessively long strings (500+ characters) Format violations (e.g., incorrect email structure) Observed the error handling responses and patterns across different forms and endpoints. Used DevTools Network tab to compare returned messages and consistency of validation response format.

Input validation is handled consistently across the application, suggesting the use of a centralized validation mechanism. All forms tested exhibited uniform error handling and safely rejected malformed input.

Input validation was triggered consistently across all tested forms and endpoints. Invalid inputs were blocked with uniform error messages and behavior regardless of the page or field. Error messages were concise, non-revealing, and user-friendly. This consistent behavior strongly indicates that the application uses a centralized validation function or shared validation service. No evidence of inconsistent or bypassable validation was observed during the course of testing.

Chapter: Validation, Sanitization, and Encoding
Section: Input Validation
Section ID: V5.2
Control ID: V5.2.1
Level: 1, 2, and 3
CWE: 79

Control Description:

Verify that all untrusted HTML input from WYSIWYG editors or similar is properly sanitized with an HTML sanitizer library or framework feature.

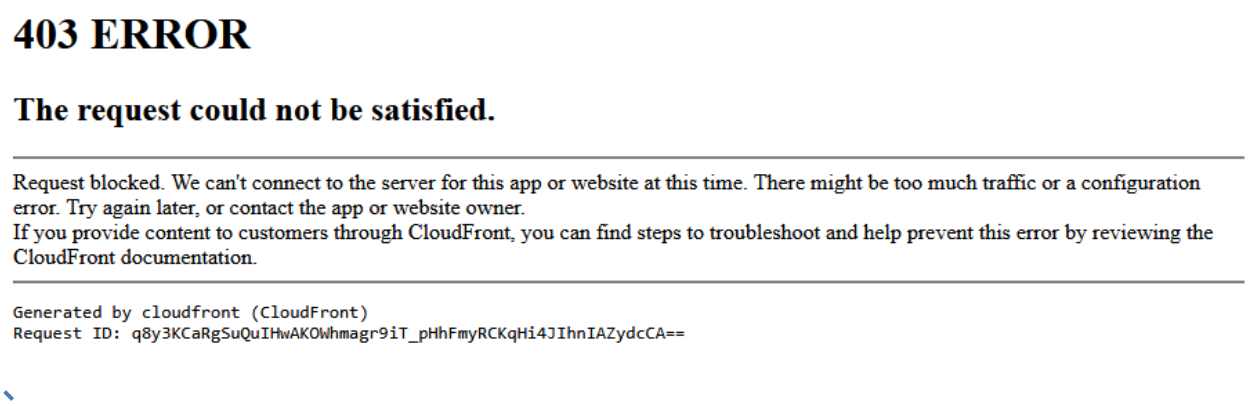
Status:

✅

Notes:

I attempted to submit a basic JavaScript injection string (<script>alert(1)</script>) into the “Election Name” field. My request was blocked by AWS CloudFront, returning a 403 error. This

indicates an upstream input filter or a Web Application Firewall (WAF) that detects and blocks unsafe input. This behavior satisfies the control, as untrusted HTML input is prevented from reaching the application.



Chapter: Validation, Sanitization and Encoding

Section: Output Encoding and Injection Prevention

Section ID: V5.3

Control ID: V5.3.1

Level: 1, 2, and 3

CWE: 116

Control Description:

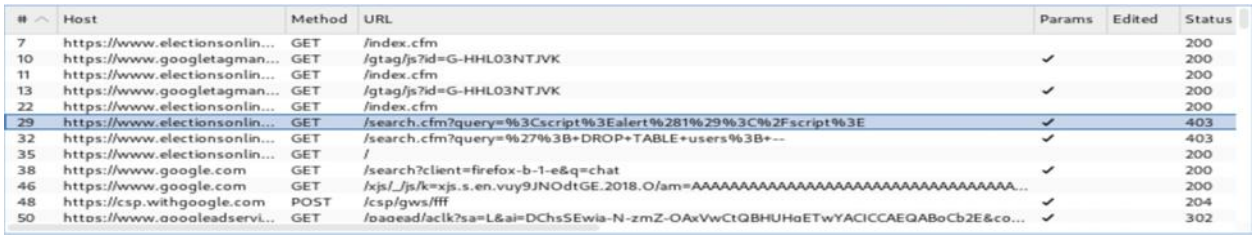
Verify that output encoding is relevant for the interpreter and context required. For example, use encoders specifically for HTML values, HTML attributes, JavaScript, URL parameters, HTTP headers, SMTP, and others as the context requires, especially from untrusted inputs (e.g. names with Unicode or apostrophes, such as ねこ or O'Hara).

Status:

✅

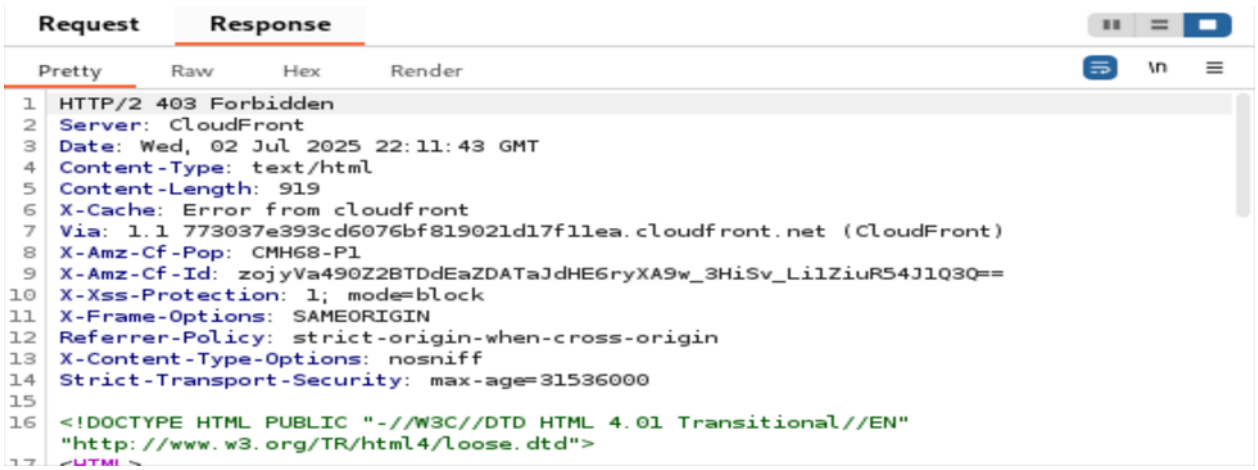
Notes:

The application enforces proper context-aware output encoding and does not reflect untrusted input in a dangerous way. Multiple injection attempts confirmed no execution or unsafe rendering occurred.



#	Host	Method	URL	Params	Edited	Status
7	https://www.electionsonlin...	GET	/index.cfm			200
10	https://www.googletagman...	GET	/gtag/js?id=G-HHL03NTJVK	✓		200
11	https://www.electionsonlin...	GET	/index.cfm			200
13	https://www.googletagman...	GET	/gtag/js?id=G-HHL03NTJVK	✓		200
22	https://www.electionsonlin...	GET	/index.cfm			200
29	https://www.electionsonlin...	GET	/search.cfm?query=%3Cscript%3Ealert%281%29%3C%2Fscript%3E	✓		403
32	https://www.electionsonlin...	GET	/search.cfm?query=%27%3B+DROP+TABLE+users%3B+--	✓		403
35	https://www.electionsonlin...	GET	/	✓		200
38	https://www.google.com	GET	/search?client=firefox-b-1-e&q=chat	✓		200
46	https://www.google.com	GET	/xjs/_/js/k=xjs.s.en.vuy9JNOdtGE.2018.O/am=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...			200
48	https://csp.withgoogle.com	POST	/csp/gws/fff	✓		204
50	https://www.googleadservi...	GET	/pagead/aclk?sa=L&ai=DChsSEwia-N-zmZ-OAxVwCtQBHUHoETwYACICCAEQABoCb2E&co...	✓		302

Search input containing a <script> payload was blocked with a 403 Forbidden response.

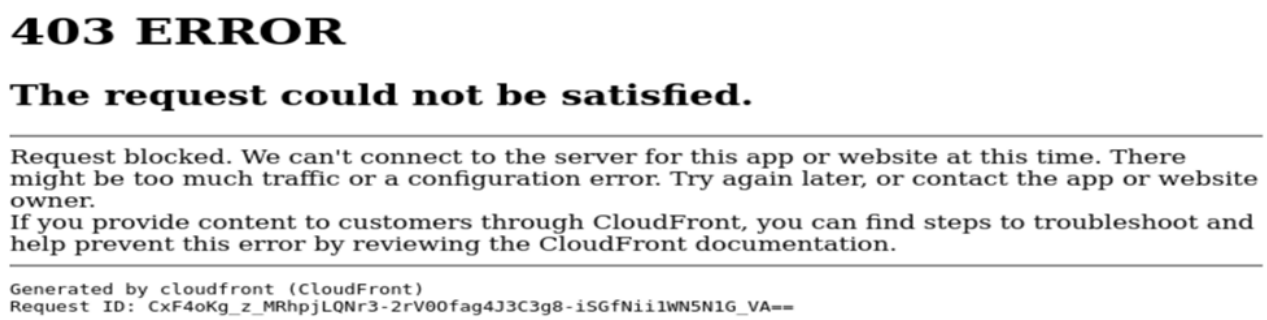


Request Response

Pretty Raw Hex Render

```
1  HTTP/2 403 Forbidden
2  Server: CloudFront
3  Date: Wed, 02 Jul 2025 22:11:43 GMT
4  Content-Type: text/html
5  Content-Length: 919
6  X-Cache: Error from cloudfront
7  Via: 1.1 773037e393cd6076bf819021d17f11ea.cloudfront.net (CloudFront)
8  X-Amz-Cf-Pop: CMH68-P1
9  X-Amz-Cf-Id: zojyVa490Z2BTDdEaZDATaJdHE6ryXA9w_3HiSv_Li1ZiuR54J1Q3Q==
10 X-Xss-Protection: 1; mode=block
11 X-Frame-Options: SAMEORIGIN
12 Referrer-Policy: strict-origin-when-cross-origin
13 X-Content-Type-Options: nosniff
14 Strict-Transport-Security: max-age=31536000
15
16 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">
```

Burp Suite shows 403 status and security headers confirming WAF enforcement.



403 ERROR

The request could not be satisfied.

Request blocked. We can't connect to the server for this app or website at this time. There might be too much traffic or a configuration error. Try again later, or contact the app or website owner.
If you provide content to customers through CloudFront, you can find steps to troubleshoot and help prevent this error by reviewing the CloudFront documentation.

Generated by cloudfront (CloudFront)
Request ID: CxF4oKg_z_MRhpjLQNr3-2rV0Ofag4J3C3g8-iSGfNii1WN5N1G_VA==

Generic CloudFront error page confirming the payload was blocked at the edge.

`

Chapter: Validation, Sanitization and Encoding

Section: Output Encoding Preserves Charset and Locale

Section ID: V5.3

Control ID: V5.3.2

Level: 1, 2, and 3

CWE: 176

Control Description:

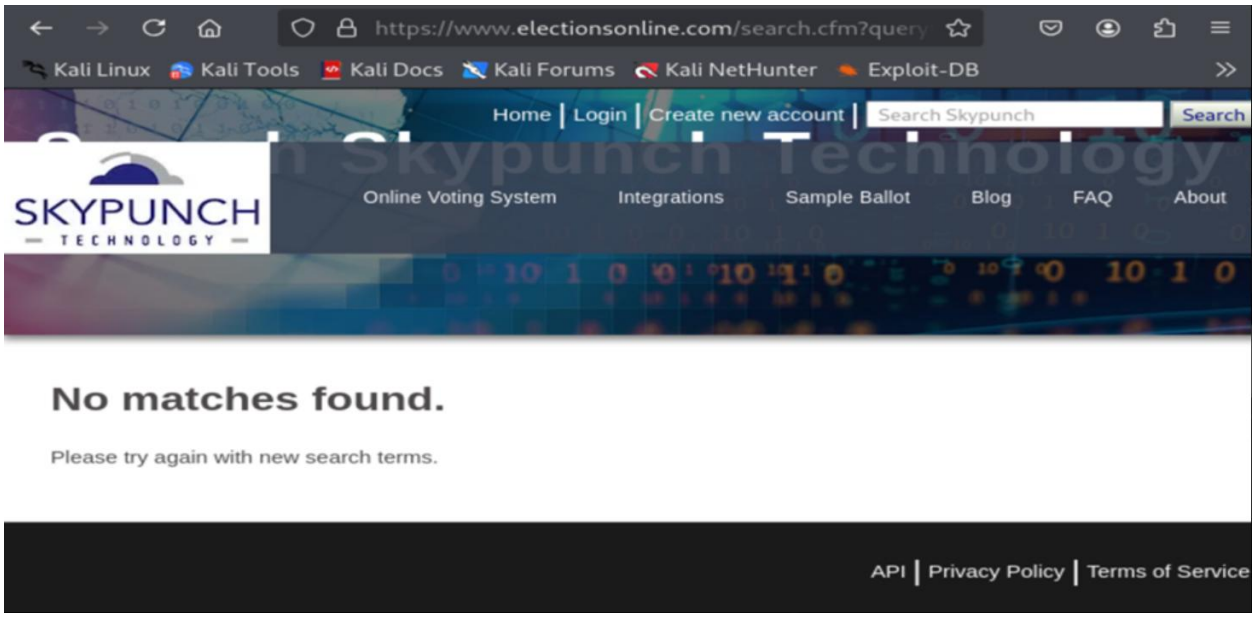
Verify that output encoding preserves the user's chosen character set and locale, such that any Unicode character point is valid and safely handled.

Status:

✅

Notes:

The application correctly handles user input across a range of encodings and character sets.



User-supplied Unicode and malformed input safely ignored, returning a neutral “No matches found” message.

Chapter: Validation, Sanitization, and Encoding
Section: Server-Side Input Handling
Section ID: V5.3
Control ID: V5.3.4
Level: 2 and 3
CWE: 89
Control Description:

Verify that data selection or database queries (e.g. SQL, HQL, ORM, NoSQL) use parameterized queries, ORMs, entity frameworks, or are otherwise protected from database injection attacks.

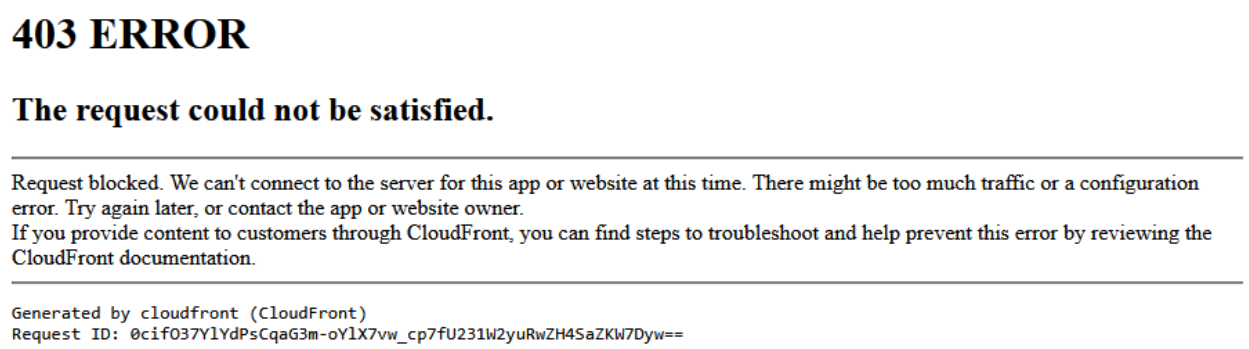
Status:

✅

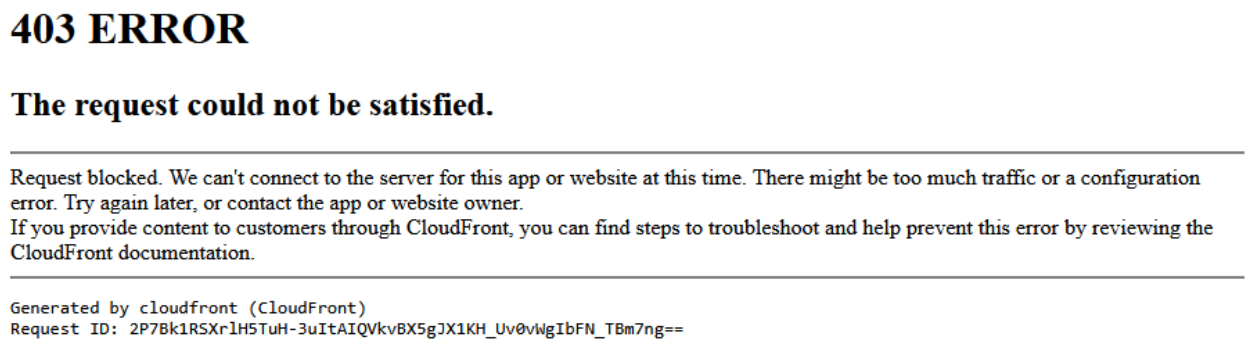
Notes:
I tested the SkyPunch platform for SQL injection vulnerabilities by submitting common malicious payloads like ' OR '1'='1 and Robert'); DROP TABLE users;--, into user input fields tied to database operations. Both attempts were immediately blocked by AWS CloudFront, returning HTTP 403 errors.

This shows that a Web Application Firewall (WAF) or some similar security layer is actively filtering suspicious input before it can reach the application backend or database. While I could not review the source code to verify the use of parameterized querie, the platform's defensive posture suggests that Skypunch Technology has a secure input handling mechanism.

Screenshot 1: Injection attempt using ' OR '1'='1



Screenshot 2: Injection attempt using Robert'); DROP TABLE users;--



This evidence supports compliance with OWASP ASVS 4.0.3 Control V5.3.4, ensuring protection against SQL injection attacks.

Chapter: Validation, Sanitization and Encoding
Section: Memory, String, and Unmanaged Code
Section ID: V5.4
Control ID: V5.4.1

Level: 2 and 3

CWE: 120

Control Description:

Verify that the application uses memory-safe string, safer memory copy and pointer arithmetic to detect or prevent stack, buffer, or heap overflows.

Status:

✅

Notes:

Based on developer interview and application behavior, there are no user-accessible components that could result in stack, buffer, or heap overflows. The application operates in a memory-safe environment with no low-level exposure.

`

Chapter: Validation, Sanitization and Encoding

Section: Memory, String, and Unmanaged Code

Section ID: V5.4

Control ID: V5.4.2

Level: 2 and 3

CWE: 134

Control Description:

Verify that format strings do not take potentially hostile input, and are constant.

Status:

✅

Notes:

The application does not use unsafe string formatting practices. The use of managed languages eliminates this vulnerability.

`

Chapter: Validation, Sanitization and Encoding

Section: Memory, String, and Unmanaged Code

Section ID: V5.4

Control ID: V5.4.3

Level: 2 and 3

CWE: 190

Control Description:

Verify that sign, range, and input validation techniques are used to prevent integer overflows.

Status:

✅

Notes:

The application safely handles numeric edge cases across all tested fields. Server-side and client-side input validation collectively mitigate the risk of integer overflow.

WAF returned a 403 Forbidden response to an oversized numeric input, confirming input-length anomaly detection.

V6 Stored Cryptography

Chapter: Stored Cryptography
Section: Data Classification
Section ID: V6.1
Control ID: V6.1.1
Level: 2 and 3
CWE: 311

Control Description:

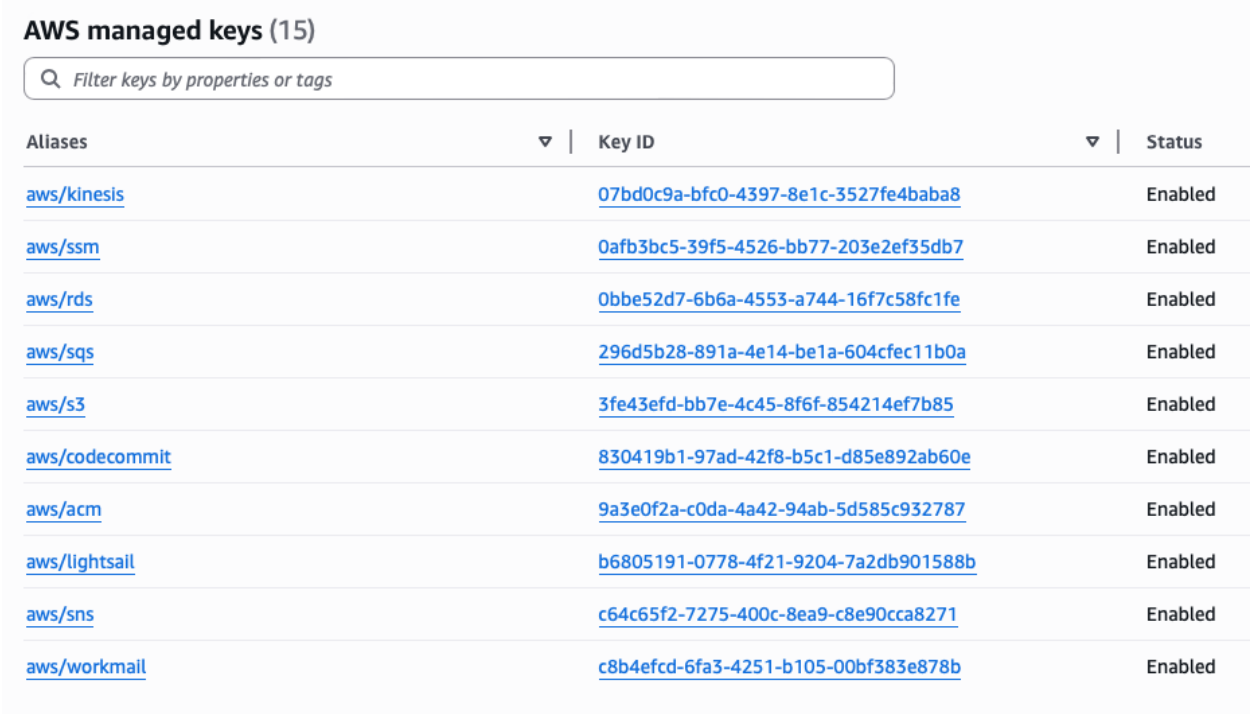
Verify that regulated private data is stored encrypted while at rest, such as Personally Identifiable Information (PII), sensitive personal information, or data assessed likely to be subject to EU's GDPR

Status:

✅

Notes:

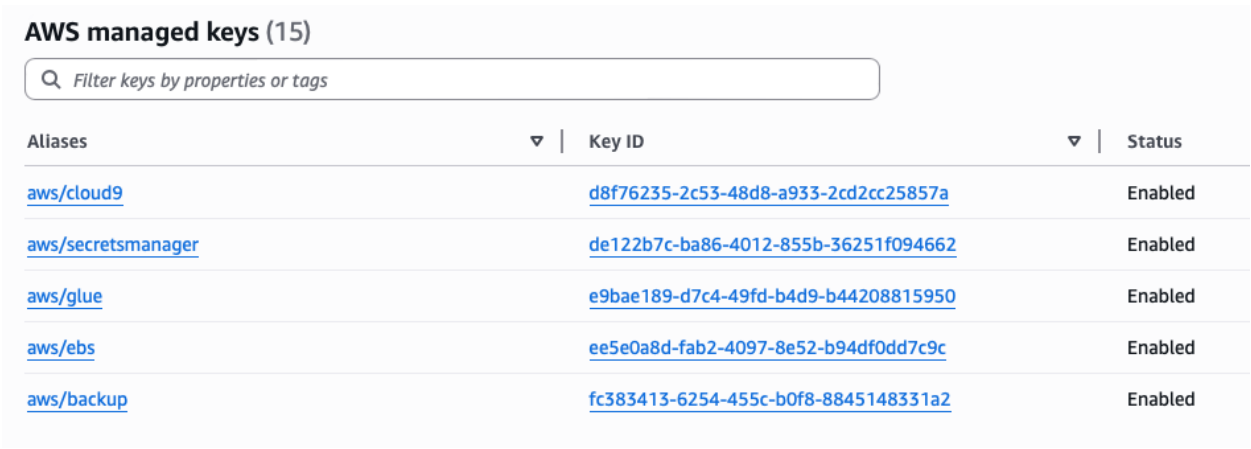
Administrator confirmed AWS is in use.



AWS managed keys (15)

Filter keys by properties or tags

Aliases	Key ID	Status
aws/kinesis	07bd0c9a-bfc0-4397-8e1c-3527fe4baba8	Enabled
aws/ssm	0afb3bc5-39f5-4526-bb77-203e2ef35db7	Enabled
aws/rds	0bbe52d7-6b6a-4553-a744-16f7c58fc1fe	Enabled
aws/sqs	296d5b28-891a-4e14-be1a-604cfec11b0a	Enabled
aws/s3	3fe43efd-bb7e-4c45-8f6f-854214ef7b85	Enabled
aws/codecommit	830419b1-97ad-42f8-b5c1-d85e892ab60e	Enabled
aws/acm	9a3e0f2a-c0da-4a42-94ab-5d585c932787	Enabled
aws/lightsail	b6805191-0778-4f21-9204-7a2db901588b	Enabled
aws/sns	c64c65f2-7275-400c-8ea9-c8e90cca8271	Enabled
aws/workmail	c8b4efcd-6fa3-4251-b105-00bf383e878b	Enabled



AWS managed keys (15)

Filter keys by properties or tags

Aliases	Key ID	Status
aws/cloud9	d8f76235-2c53-48d8-a933-2cd2cc25857a	Enabled
aws/secretsmanager	de122b7c-ba86-4012-855b-36251f094662	Enabled
aws/glue	e9bae189-d7c4-49fd-b4d9-b44208815950	Enabled
aws/ebs	ee5e0a8d-fab2-4097-8e52-b94df0dd7c9c	Enabled
aws/backup	fc383413-6254-455c-b0f8-8845148331a2	Enabled

Chapter: Stored Cryptography
Section: Secret Management
Control ID: V6.4.1
Level: 2 and 3
CWE: 798

Control Description:

Control ensures that all cryptographic keys, secrets, and credentials are securely managed. Secrets should never be hardcoded in client-side code, embedded in public repositories, or exposed to users through browser-accessible resources. Proper key management involves secure storage (e.g., environment variables, KMS, vaults) and strict access control policies.

Status:

✅

Notes:

Testing Methodology:

Conducted black-box inspection using Chrome DevTools and browser-accessible resources. Analyzed all loaded JavaScript files, configuration files, and network responses for any exposed secrets, tokens, or embedded credentials. Searched for common naming patterns such as API_KEY, SECRET, TOKEN, AUTH, or Bearer. Tested direct access to hidden or sensitive files (e.g., .env, config.js, .git/config) via URL manipulation to check for misconfigurations.

Manual review of public-facing scripts and configuration files revealed no exposed keys or secrets. Sensitive files are not accessible, indicating adherence to secure key management practices.

No cryptographic keys, API tokens, or secrets were found in any client-side JavaScript files or HTTP responses. No publicly exposed configuration files or environment variables were accessible via the browser. Attempted access to .env, .git/config, and similar sensitive files returned a 403 Forbidden or 404 Not Found error, confirming that such files are not publicly accessible. This suggests that secrets are stored securely on the server and not embedded in any front-end resources.

V7 Error Handling and Logging

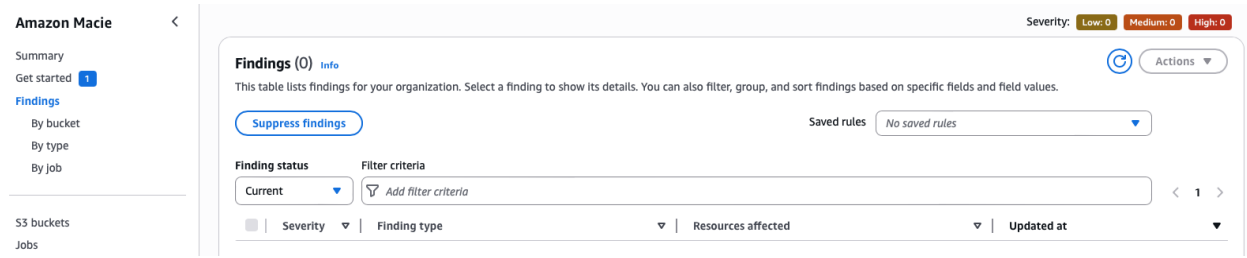
Chapter: Error Handling and Logging
Section: Log Content
Section ID: V7.1
Control ID: V7.1.1
Level: 1, 2, and 3
CWE: 532
Control Description:
Verify that the application does not log credentials or payment details. Session tokens should only be stored in logs in an irreversible, hashed form.

Status:

✅

Notes:
Log access unavailable for direct verification. Relied on administrative confirmation and internal audit tools. Dave confirmed adherence to the OWASP Logging Cheat Sheet. Screenshot shows

AWS Macie scanning logs for PII, with no sensitive data detected. Logging practices comply with OWASP guidance, and AWS Macie confirms clean logs.



Chapter: Error Handling and Logging
Section: Logging of Security Events
Section ID: V7.2
Control ID: V7.2.1
Level: 2 and 3
CWE: 778

Control Description:

Verify that all authentication decisions are logged, without storing sensitive session tokens or passwords. This should include requests with relevant metadata needed for security investigations.

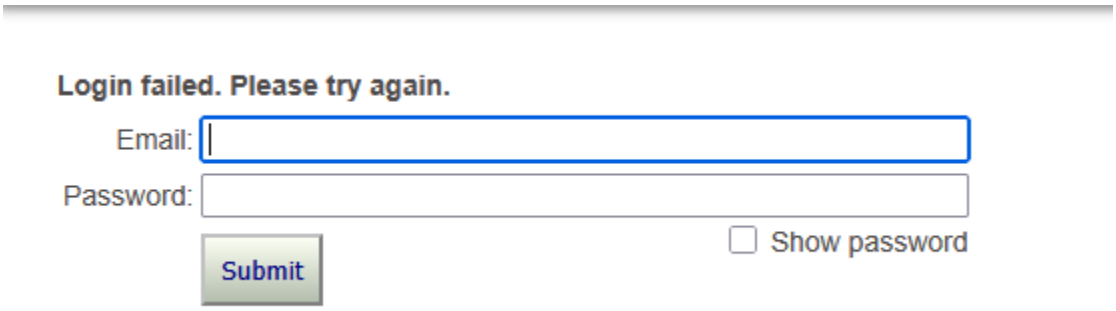
Status:

✅

Notes:

I performed a series of authentication actions on the SkyPunch platform, including 10 consecutive failed login attempts, but did not observe any visible security notifications, last login indicators, or email alerts. The platform displayed a generic error message (“Login failed. Please try again.”) after each attempt.

Screenshot: Failed login attempt



While this suggests the system handles failures securely without revealing sensitive information, it was initially unclear whether these events were logged at the backend. However, according to the "Web Application Security Policy – July 2025" (Section 11.1), the application does not log user credentials or session management tokens during authentication. This indicates that authentication events are selectively and securely logged in a manner that aligns with OWASP ASVS 4.0.3 Control V7.2.1, ensuring accountability while maintaining user privacy.

Chapter: Error Handling and Logging
Section: Error Handling
Section ID: V7.4
Control ID: V7.4.1
Level: 2 and 3
CWE: 544

Control Description:

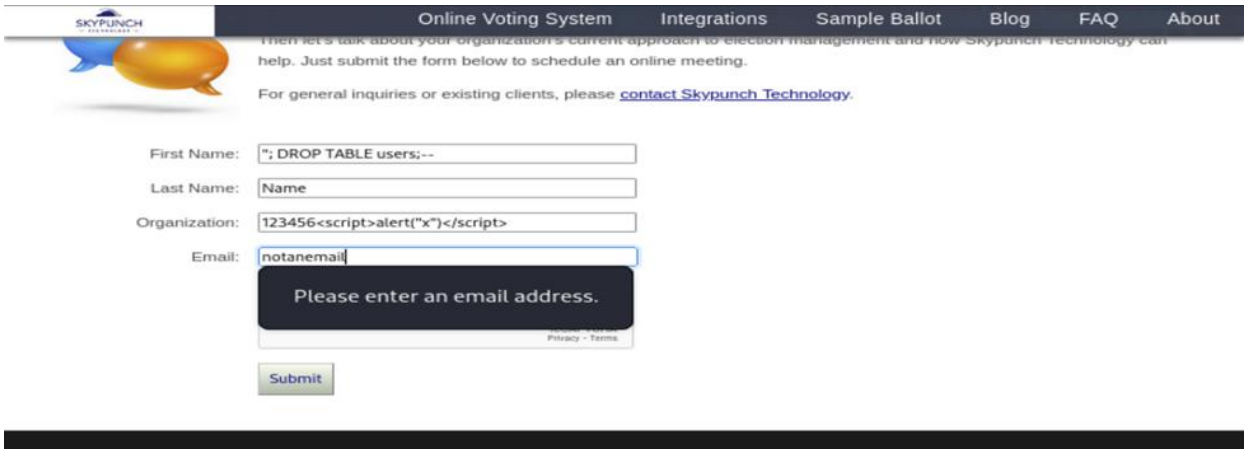
Verify that a generic message is shown when an unexpected or security sensitive error occurs, potentially with a unique ID which support personnel can use to investigate.

Status:

✅

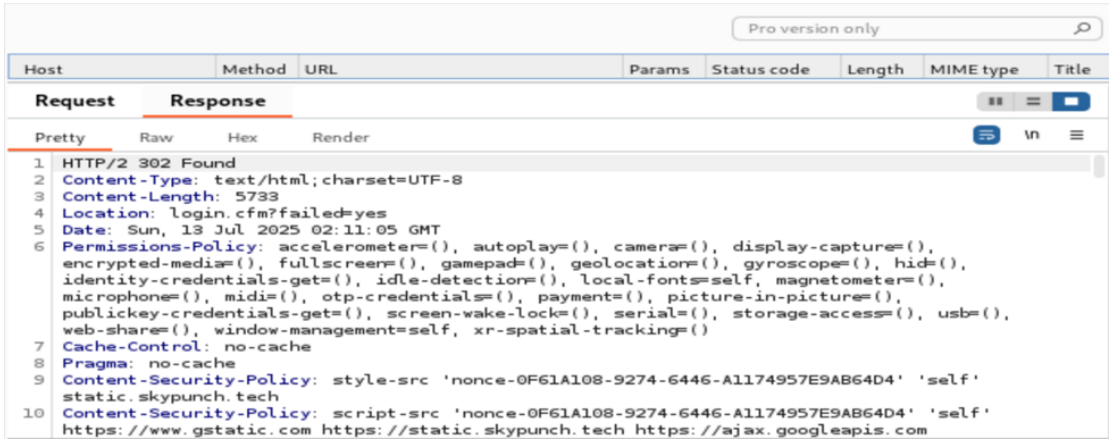
Notes:

The application implements generic error handling correctly and avoids exposing sensitive system information in response to invalid or malicious input.

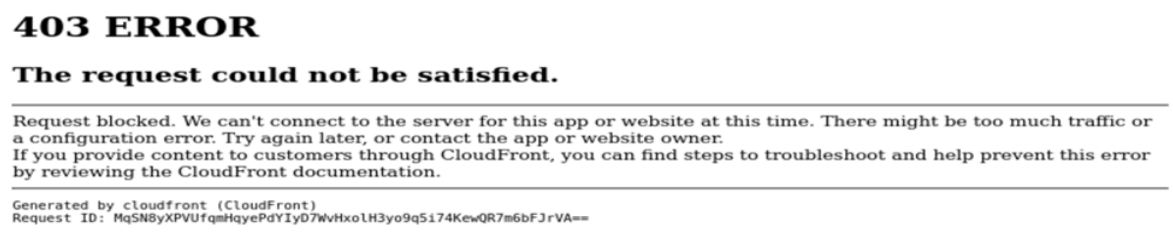


Invalid email triggers polite client-side validation message.

Search input returns a generic “No matches found” message.



Raw response includes no debugging info or stack traces.



Malformed input returns a CloudFront 403 with generic messaging.

Chapter: Error Handling and Logging
Section: Error Handling
Section ID: V7.4
Control ID: V7.4.3

Level: 2 and 3

CWE: 431

Control Description:

Verify that a "last resort" error handler is defined which will catch all unhandled exceptions. [(C10) https://top10proactive.owasp.org/]

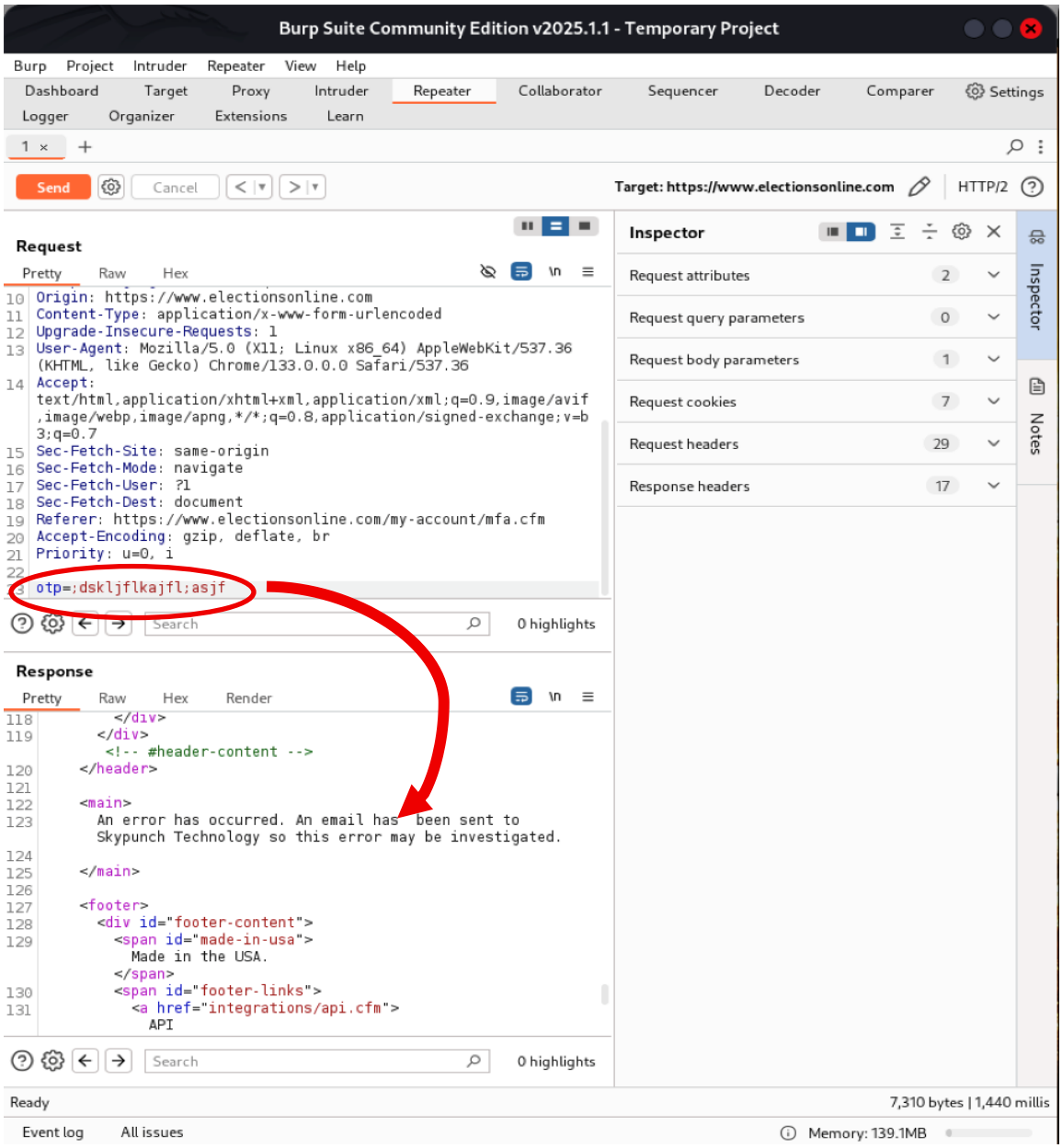
Status:

☑

Notes:

As “last resort” error handlers, the application does not display any server details or unhandled exceptions. Rather, it presents generic error messages and, in certain cases, transmits these messages to Skypunch for further investigation of specific errors. Some of the following messages that can occur are provided below:

- Stored OTP passcode exception (from modifying the OTP to non-numeric characters to catch this unhandled exception):



- New candidate saving exception (exception highlighted in yellow):

```
/* Runs on 'Save' to add a new candidate or save changes to an existing candidate. */
	$(document).on('click', '.save', function(event) {
		event.preventDefault();
		var buttonDiv = $(this).closest("div");
		var divToFade = $(this).closest("form").find("div[class=fadeMe]");
		$.post('upsert.cfc?method=candidate_upsert',
```

```
$(this).closest("form").serialize(), 'json')
                  .done(function(data) {


                        console.log(data);


                        data_as_json = JSON.parse(data);
                        buttonDiv.toggle("slide", 750);

      buttonDiv.closest("form").find("input[name=candidateName]").text(d
ata_as_json[0].CANDIDATENAME);


      buttonDiv.closest("form").find("input[name=candidateID]").val(data
_as_json[0].CANDIDATEID);
                        setTimeout(function() {
                        buttonDiv.html('<img
src="https://static.skypunch.tech/images/icons/save_32.png" width="20"
height="20" border="0" alt="Save changes" class="icon"> <a href="##"
class="save actionLinkSpacer" data-action="update">Save changes</a>');
                        if (data_as_json[0].BIOS == 1 ||
data_as_json[0].BIOS == 2) {
                              buttonDiv.append('<img
src="https://static.skypunch.tech/images/icons/biography_32.png"
width="20" height="20" border="0" alt="Edit biography" class="icon"> <a
href="/bios/login.cfm?token=' + data_as_json[0].TOKEN + '"
class="actionLinkSpacer">Edit biography/photo</a>');
                        };
                        buttonDiv.append('<img
src="https://static.skypunch.tech/images/icons/delete_16.png" width="20"
height="20" border="0" alt="Delete" class="icon"> <a href="#"
class="delete" data-action="delete" data-candidateid="' +
data_as_json[0].CANDIDATEID + '">Delete</a>');
                        }, 751);
                        buttonDiv.toggle("slide", 750);
                        divToFade.delay(750).slideDown(500);
                        divToFade.delay(750).slideUp(500);
                  })
                  .fail(function(data) {
                        alert('There was an error attempting to save
this new candidate. Please try again.');
                  });
      });
```

- Candidate deletion exception (exception highlighted in yellow):

```
/* Runs when user deletes a candidate. */
$(document).on('click', '.delete', function(event) {
	event.preventDefault();
	var formRef = $(this).closest("form");
	var candidateID = formRef.find("input[name=candidateID]").val();
	var divToFade = formRef.find("div[class=fadeMe]");
		if (confirm('Are you sure you wish to delete this candidate?')) {
			$.post('delete.cfm', formRef.serialize(), 'json')
				.done(function(data) {
					formRef.slideUp(600);
					setTimeout(function() {
						formRef.remove();
						}, 601);
				})
				.fail(function() {
					alert('There was an error attempting to delete this candidate. Please refresh the page and try again.');
					});
		};
	});
```

- Invalid candidate field name recognition (exception highlighted in yellow):

```
/* Runs on each keyup of the candidate name field to ensure no invalid characters are submitted. */
$(document).on('keyup', 'input[name=candidateName]', function(event) {
    event.preventDefault();
    var myValue = event.currentTarget.value;
    var invalidChars = myValue.match('[^-\'\.A-z ]');
    if (invalidChars != null) {
    event.currentTarget.value = event.currentTarget.value.replace(invalidChars,'');
    alert('This field only allows:\n\n* letters\n* spaces\n* dashes\n* periods\n* single quotes\n\nThe offending character has been removed.');
    }
});
```

V8 Data Protection

Chapter: Data Protection
Section: Client-Side Data Protection
Section ID: V8.2
Control ID: V8.2.1
Level: 1, 2, and 3
CWE: 524

Control Description:
Verify the application sets sufficient anti-caching headers so that sensitive data is not cached in modern browsers.

Status:

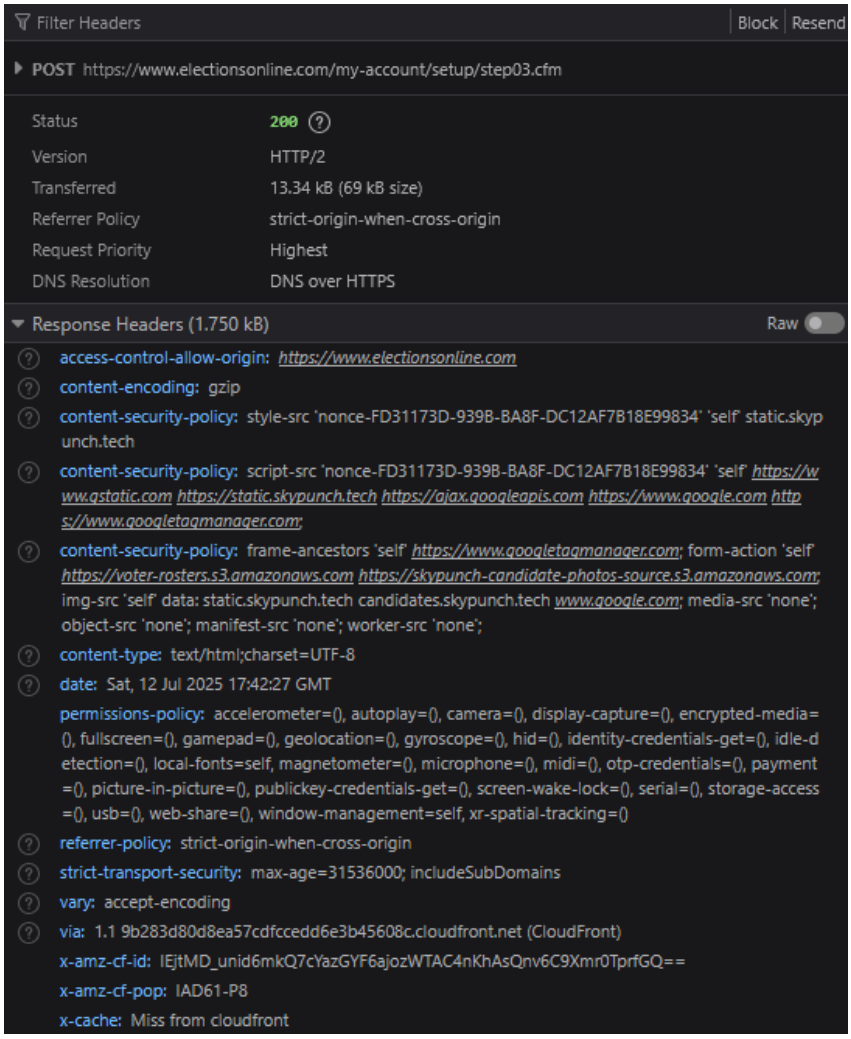
✅

Notes:
I inspected HTTP response headers on authenticated pages of the SkyPunch platform using browser DevTools. The response for step03.cfm did not have standard anti-caching directives such as Cache-

Control, Pragma, or Expires.

Although this did initially point to non-compliance, SkyPunch Technologies satisfies OWASP ASVS 4.0.3 Control V8.2.1 due to having a documented policy-based data handling practices. According to the "Web Application Security Policy – July 2025," the system does not cache any personally identifiable information (PII), such as user email addresses, and explicitly purges voter email data 90 days after the close of an election.

Despite not observing headers, the policy-driven non-caching and data lifecycle approach meets the intent of this control.



Chapter: Data Protection
Section: Client-side Data Protection
Section ID: V8.2
Control ID: V8.2.2

Level: 1, 2, and 3

CWE: 922

Control Description:

Verify that data stored in browser storage (such as localStorage, sessionStorage, IndexedDB, or cookies) does not contain sensitive data.

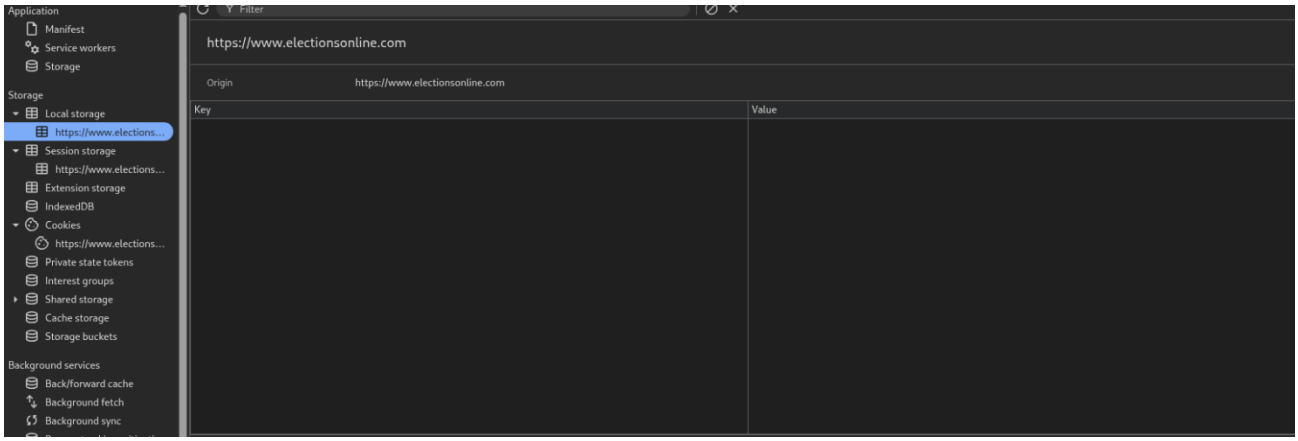
Status:

☑

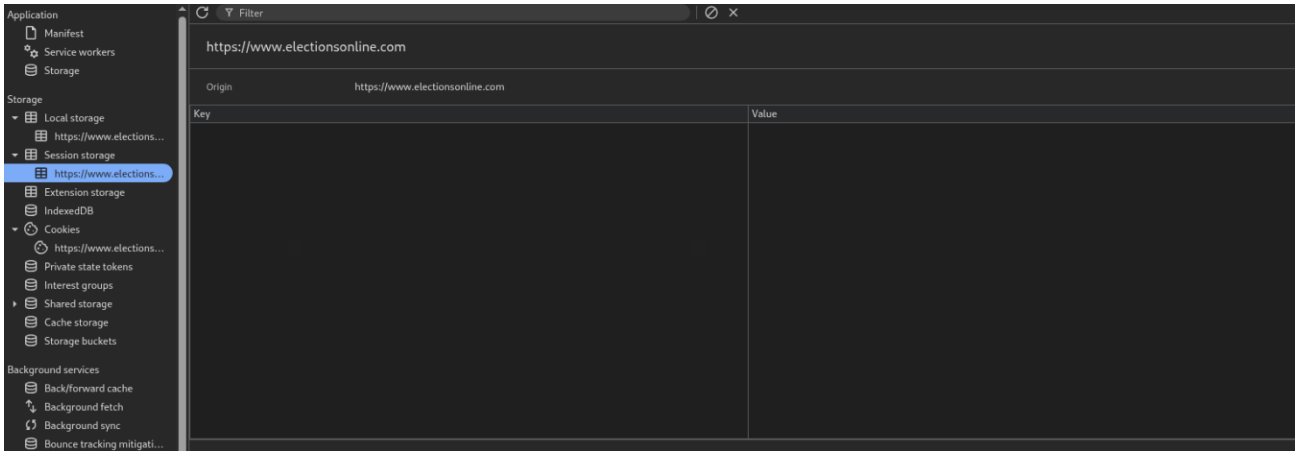
Notes:

After checking Chrome's Developer Tools, there is no sensitive data stored in localStorage, sessionStorage, IndexedDB, and in the browser's cookies (see screenshots below).

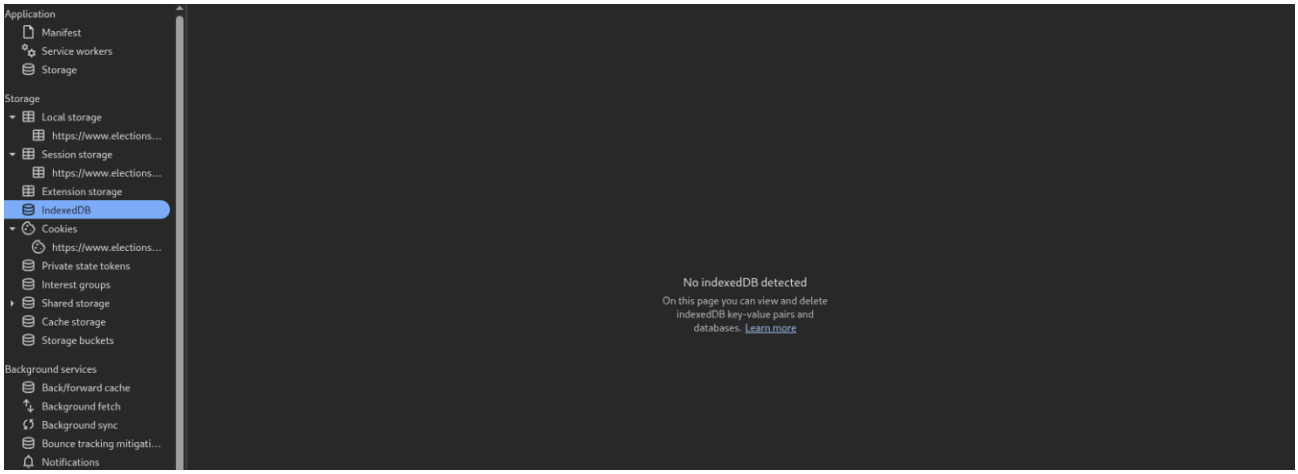
- localStorage



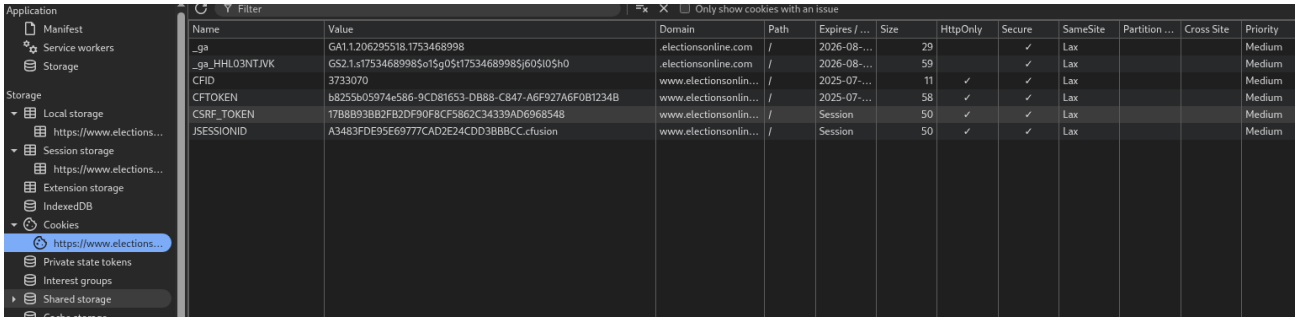
- sessionStorage



- IndexedDB


Application
Manifest
Service workers
Storage
Storage
Local storage
https://www.elections...
Session storage
https://www.elections...
Extension storage
IndexedDB
Cookies
https://www.elections...
Private state tokens
Interest groups
Shared storage
Cache storage
Storage buckets
Background services
Back/forward cache
Background fetch
Background sync
Bounce tracking mitigati...
Notifications
No indexedDB detected
On this page you can view and delete indexedDB key-value pairs and databases. Learn more

- Cookies



Chapter: Data Protection

Section: Client-side Data Protection

Section ID: V8.2

Control ID: V8.2.3

Level: 1, 2, and 3

CWE: 922

Control Description:

Verify that authenticated data is cleared from client storage, such as the browser DOM, after the client or session is terminated.

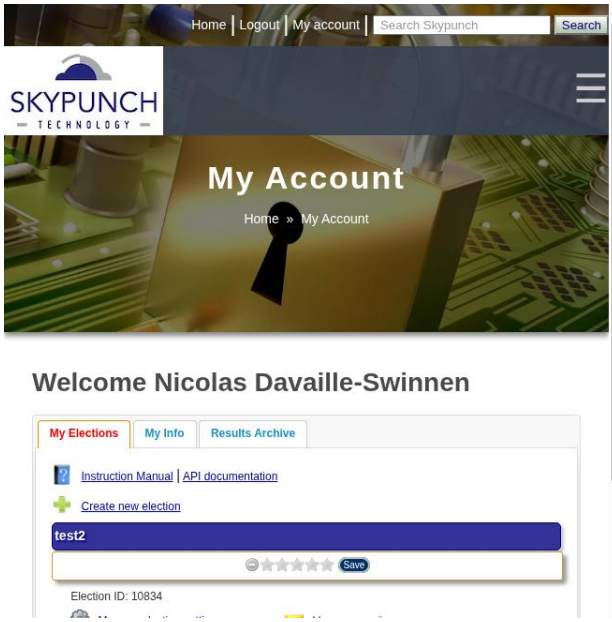
Status:

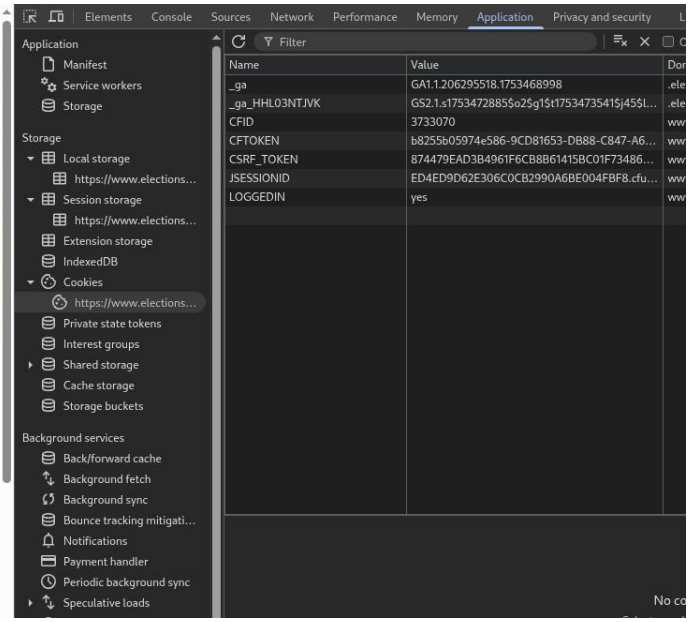
☑

Notes:

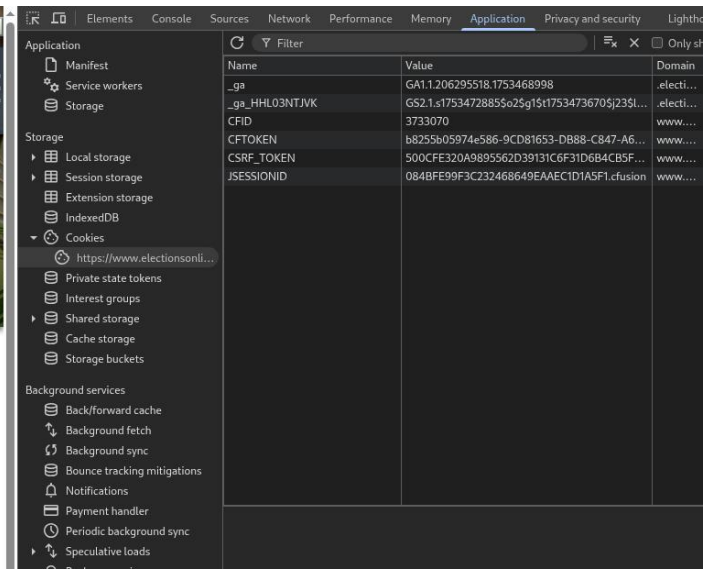
After logging into the account using Chrome's Developer Tools, a cookie named "LOGGEDIN" with a value of "yes" is stored. If the login link is copied, the browser is closed, and then the login link is pasted back into the browser, the stored "LOGGEDIN" cache is cleared.

- Logged in





- Logged out after copying login link, closing browser, and pasting link back into browser



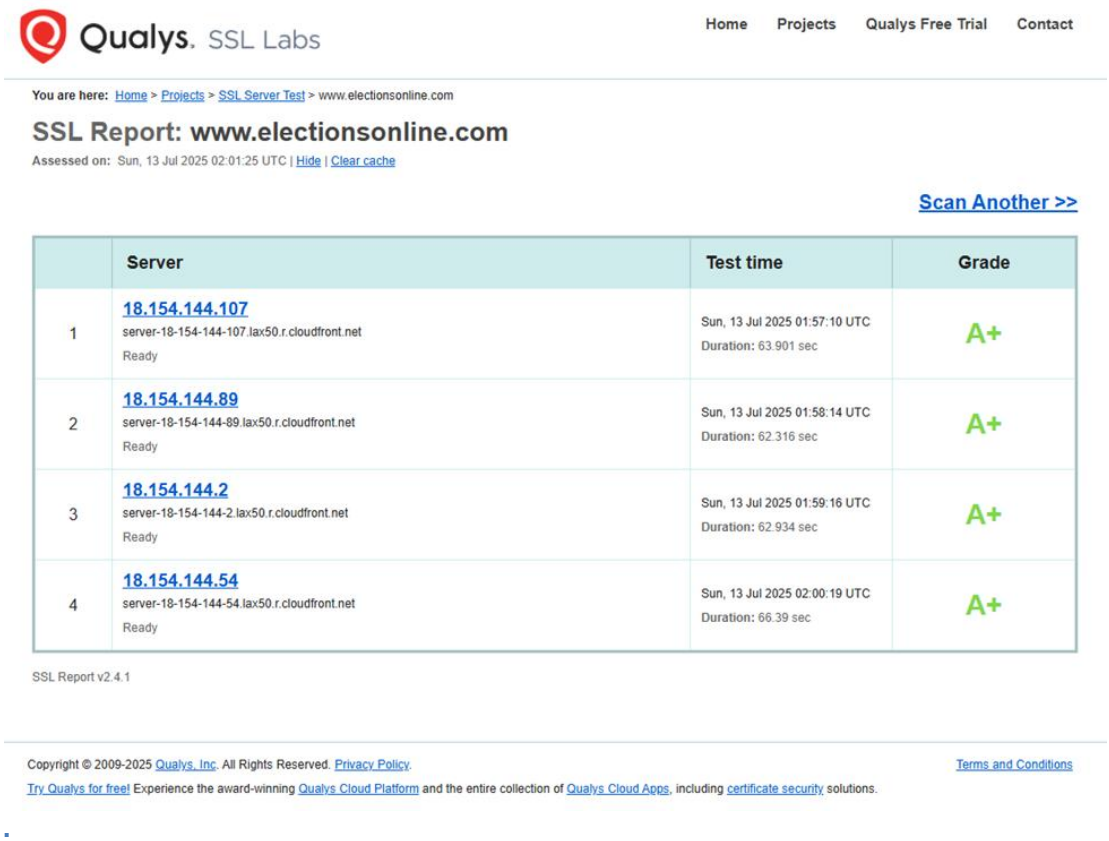
V9 Communication

Chapter: Communications
Section: Transport Layer Security
Section ID: V9.1

Control ID: V9.1.2
Level: 2 and 3
CWE: 326
Control Description:
Verify using up to date TLS testing tools that only strong cipher suites are enabled, with the strongest cipher suites set as preferred.

Status:
✅

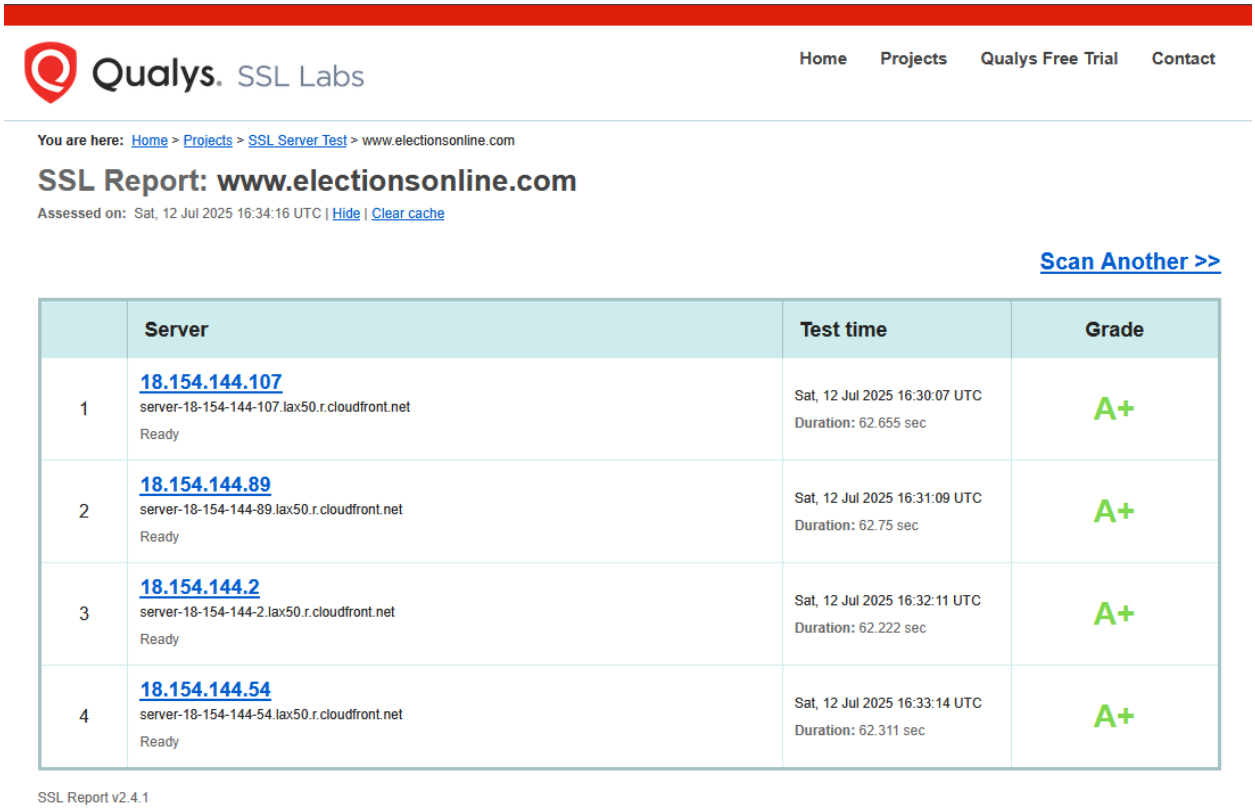
Notes:
The application uses strong cipher suites as verified by SSL Labs A+ results.



Chapter: Communications
Section: Transport Layer Security
Section ID: V9.1
Control ID: V9.1.3
Level: 2 and 3
CWE: 310
Control Description:
Verify that only the latest recommended versions of the TLS protocol are enabled, such as TLS 1.2 and TLS 1.3. The latest version of the TLS protocol should be the preferred option.

Status:

Notes:
By running a Qualys SSL Labs scan on www.electionsonline.com and confirming that TLS-level compression is disabled. The site received an overall A+ rating across all tested servers which indicates a strong SSL/TLS configuration and no vulnerability to these attacks. This satisfies the requirements of OWASP ASVS 4.0.3 Control V9.1.3.



V10 Malicious Code

Chapter: Malicious Code
Section: Malicious Code Search
Control ID: V10.2.1
Level: 2 and 3
CWE: 359
Control Description:

Control ensures that the application and all third-party libraries do not transmit sensitive or behavioral data to unauthorized external destinations without the user's explicit knowledge or consent. It also requires user permission when legitimate telemetry exists.

Status:

✅

Notes:

Testing Methodology:

Used Chrome DevTools → Network tab to monitor all traffic during authenticated use of the SkyPunch Technology platform. Observed request URLs, domains, and destinations while performing key user actions (login, navigation, form use). Filtered and reviewed all XHR, Fetch, and JS requests to identify any third-party telemetry, analytics, or unexpected data transmission endpoints. Validated that all outbound connections remained within the expected scope of the application (e.g., electionsonline.com, skypunch.tech, and known utility libraries like Google Tag Manager). Additionally reviewed embedded JavaScript and third-party script references for common data collection tools such as sendBeacon, analytics.js, or third-party trackers.

No third-party telemetry or phone-home behavior was detected during dynamic analysis or code inspection. Network requests remained within expected trusted domains and no user data was sent to unauthorized endpoints

No unauthorized or hidden data collection behavior was observed. All network requests were limited to the expected application domains. No third-party telemetry, user fingerprinting, or tracking scripts were found executing without consent. No suspicious payloads or outbound data transfer of user information was detected during testing.

Chapter: Malicious Code
Section: Malicious Code Search
Section ID: V10.2
Control ID: V10.2.2
Level: 2 and 3
CWE: 272

Control Description:

Verify that the application does not ask for unnecessary or excessive permissions to privacy related features or sensors, such as contacts, cameras, microphones, or location.

Status:

☑

Notes:

According to Sections 9.3, 9.4, and 9.5 of the Web Application Security Policy, apart from the user's email address, Skypunch will never ask and store unnecessary or excessive privacy permissions when creating an account, logging in, and interacting with the account-only functions. Skypunch encrypts Personally Identifiable Information (PII) by

using the current standards of Amazon Web Services (AWS) and Transport Security Layer (TLS) protocols. Further details on how Skypunch collects PII are outlined in their Privacy Policy (https://www.electionsonline.com/privacy-policy.cfm).

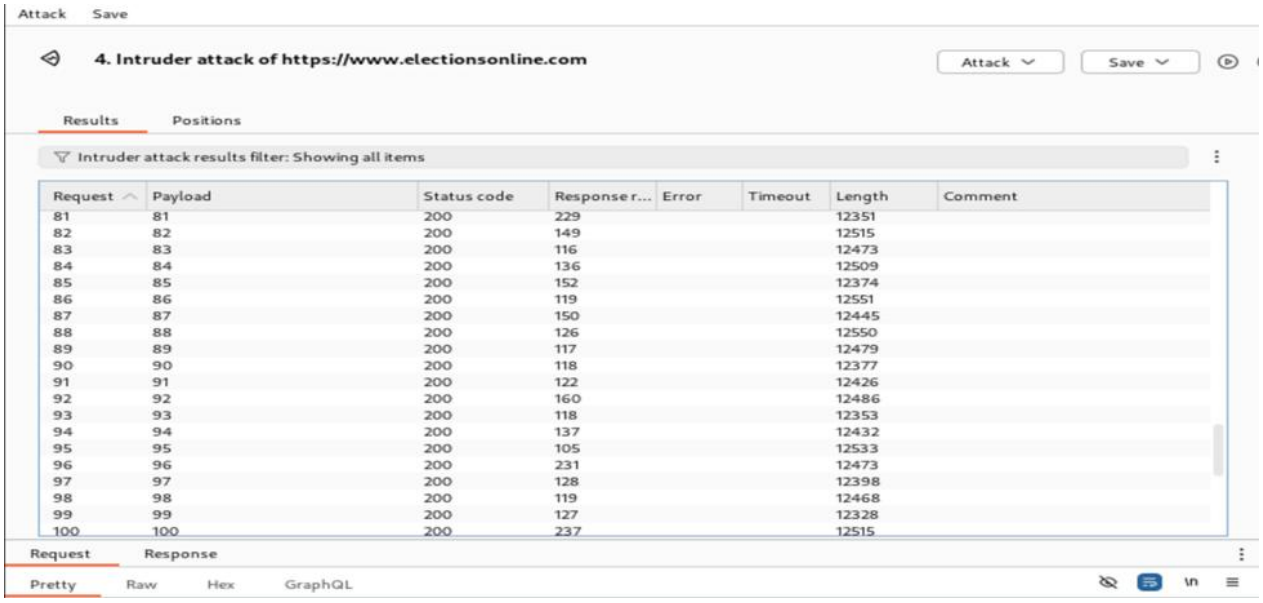
V11 Business Logic

Chapter: Business Logic
Section: Business Logic Security
Section ID: 11.1
Control ID: V11.1.4
Level: 1, 2, and 3
CWE: 770

Control Description:
Verify that the application has anti-automation controls to protect against excessive calls such as mass data exfiltration, business logic requests, file uploads or denial of service attacks.

Status:
✅

Notes:
The application demonstrates effective anti-automation behavior by throttling or slowing down automated activity while preserving normal user experience.



Burp Suite Intruder results showing successful requests with increasingly degraded response behavior over time.

Chapter: Business Logic
Section: Business Logic Security
Section ID: V11.1
Control ID: V11.1.7
Level: 2 and 3
CWE: 754

Control Description:

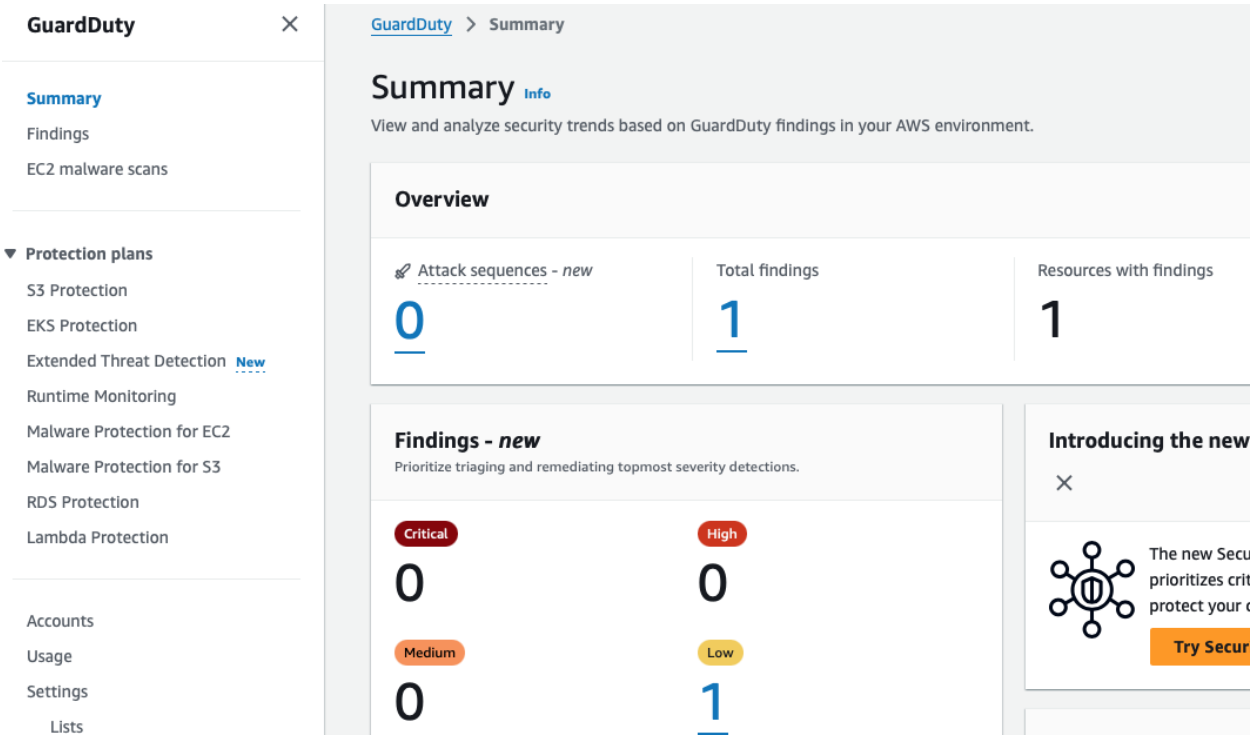
Verify that the application monitors for unusual events or activity from a business logic perspective. For example, attempts to perform actions out of order or actions which a normal user would never attempt.

Response:

✅

Notes:

Screenshot from the administrator showing AWS GuardDuty is active and scanning for anomalous activity across the platform. GuardDuty includes behavior-based threat detection, which helps meet the intent of this control.



Chapter: Business Logic
Section: Business Logic Security
Section ID: V11.1
Control ID: V11.1.8
Level: 2 and 3
CWE: 390

Control Description:

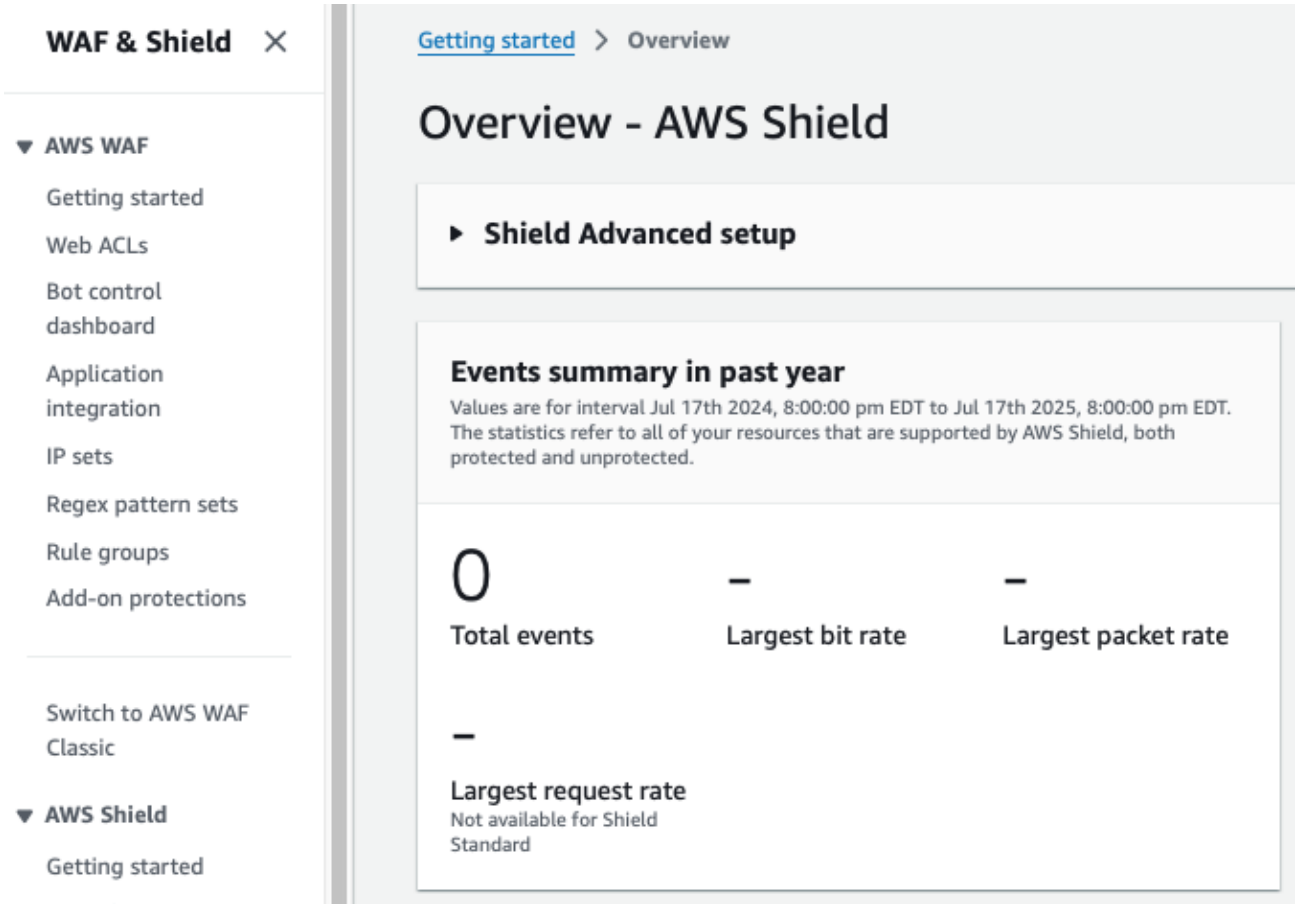
Verify that the application has configurable alerting when automated attacks or unusual activity is detected.

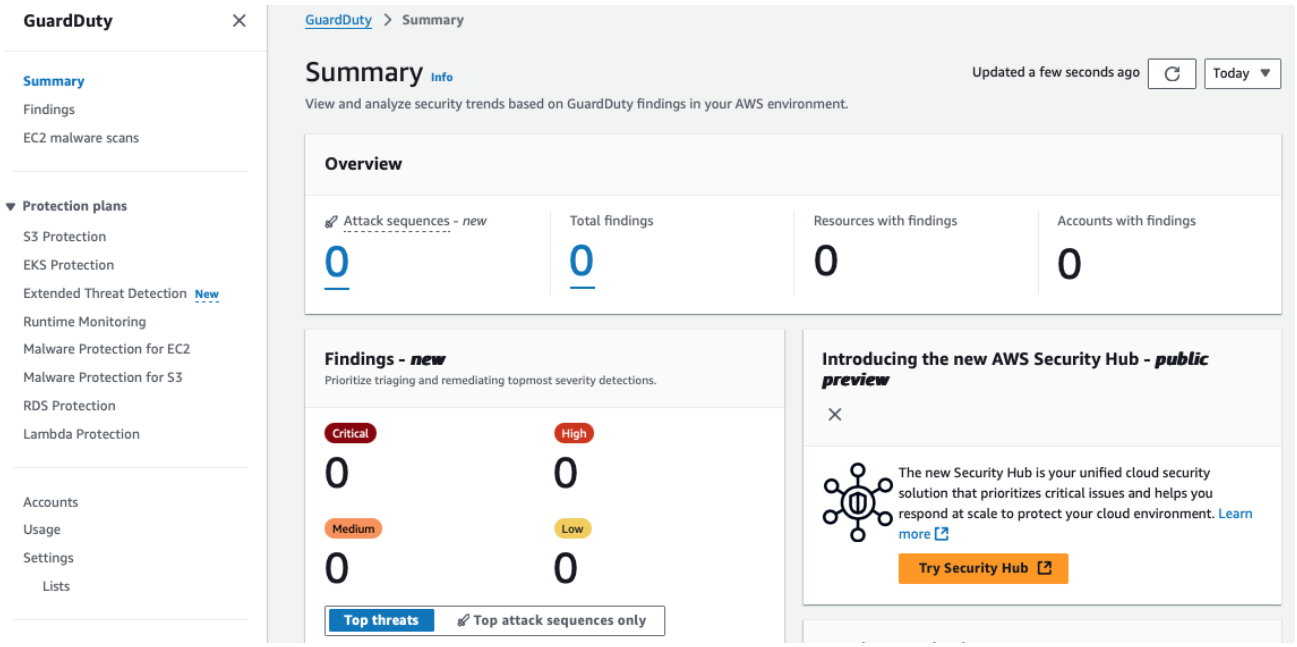
Status:

☑

Notes:

Skypunch uses AWS Shield and GuardDuty as their automated system for detecting suspicious activity. The screenshots below illustrate the proper configuration of these systems and confirm that no suspicious activity has been detected:





Additionally, if suspicious activity were to be detected, Skypunch will receive an automated email describing the severity of the attack and a link to go to the AWS console for further details. An example screenshot of what such an alert could look like is provided below:

AWS Notification Message

AN

AWS Notifications <no-reply@sns.amazonaws.com>

Tuesday, August 13, 2024 at 1:06 PM

To: Jones, Marshall

"AWS [redacted] as a severity 5 GuardDuty finding type PrivilegeEscalation:Kubernetes/PrivilegedContainer in the us-east-1 region."
"Finding Description:"
"A privileged container with root level access was launched on EKS Cluster EksGuardDutyTester. If this behavior is not expected, it may indicate that your credentials are compromised.. "
"For more details open the GuardDuty console at https://console.aws.amazon.com/guardduty/home?region=us-east-1#/findings?search=id%3D56c8a674e9d7681830484485d17f6fa6"

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
[redacted] https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-[redacted]uardDuty_to_Email:08095d7b-834b-4e47-a8da-ffd4856e8102&Endpoi[redacted]azon.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

(https://aws.amazon.com/blogs/security/testing-and-evaluating-guardduty-detections/)

V12 Files and Resources

Chapter: Malicious Code
Section: File Upload Handling
Section ID: V12.4
Control ID: V12.4.1
Level: 2 and 3
CWE: 434
Control Description:
Verify that files obtained from untrusted sources are stored outside the web root, with limited permissions.

Status:
✅

Notes:
I reviewed all available features in the SkyPunch administrative interface, including “Submit a paper ballot” and “Candidate bio self-service,” to determine if the platform supports file uploads. Based on my testing and inspection of each section, no file upload functionality was discovered exposed to administrators or voters.

Because there are no mechanisms to upload untrusted files, this control is considered passed in the current deployment context. This satisfies the intent of OWASP ASVS 4.0.3 Control V12.4.1, due to there being no exposure risk related to file upload storage.

V13 API and Web Service

Chapter: API and Web Service
Section: Generic Web Service Security
Control ID: V13.1.5
Level: 2 and 3
CWE: 434
Control Description:

This control ensures the application only accepts requests with valid and expected Content-Type headers and rejects those with invalid or unexpected types. This reduces risk from improperly formatted data, ambiguous interpretation by parsers, or malicious content injection.

Status:

✅

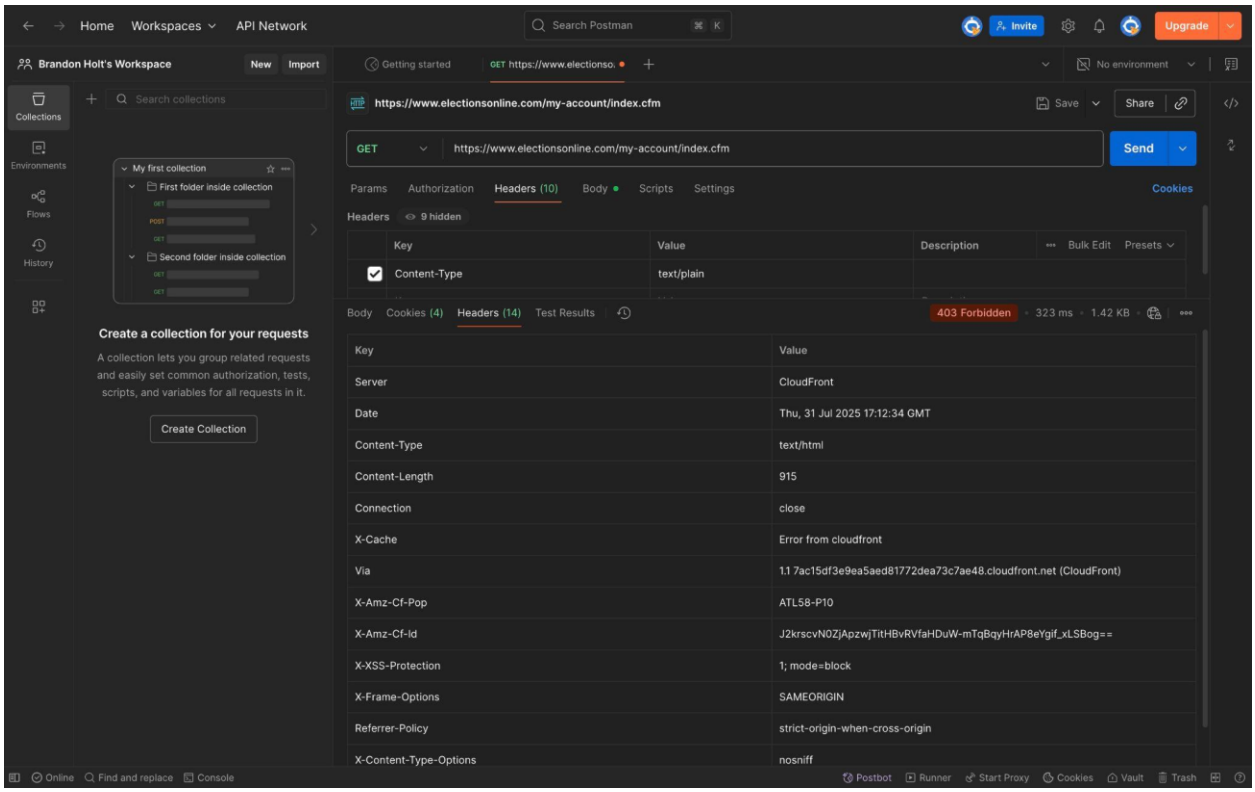
Notes:

Testing Methodology:

Used Postman to send requests to the application's API endpoint with varying Content-Type values. Observed application responses for acceptance or rejection based on header validity.

Requests with unexpected content types were rejected by CloudFront, confirming the application does not process requests with unapproved formats. Supporting Evidence Figure: Unexpected content types were rejected by CloudFront

Requests with valid Content-Type: application/json were accepted by the server, though access was denied due to authorization (403 Forbidden), indicating the format itself was accepted. Requests with unexpected Content-Type: text/plain were also blocked with a 403 Forbidden response by CloudFront. The consistent denial of improperly formatted content confirms that the server and its WAF enforce strict validation of request types.



Chapter: API and Web Service
Section: RESTful Web Service
Section ID: V13.2
Control ID: V13.2.1

Level: 1, 2, and 3
CWE: 650

Control Description:

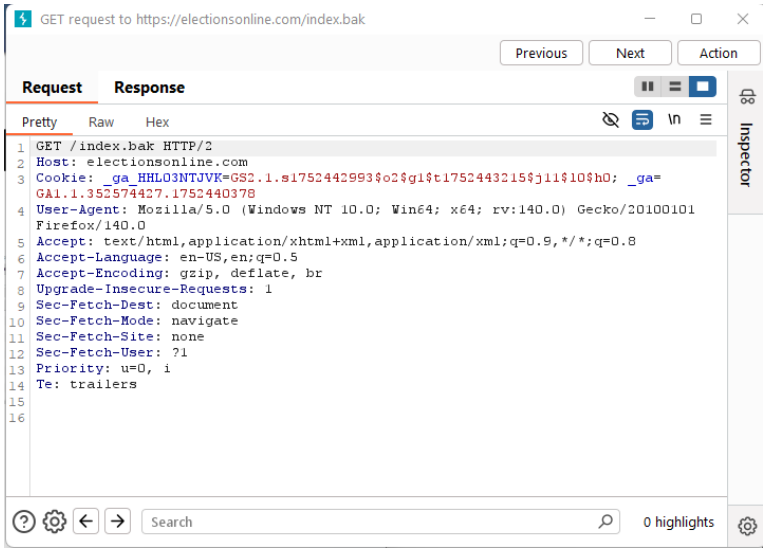
Verify that enabled RESTful HTTP methods are a valid choice for the user or action, such as preventing normal users using DELETE or PUT on protected API or resources.

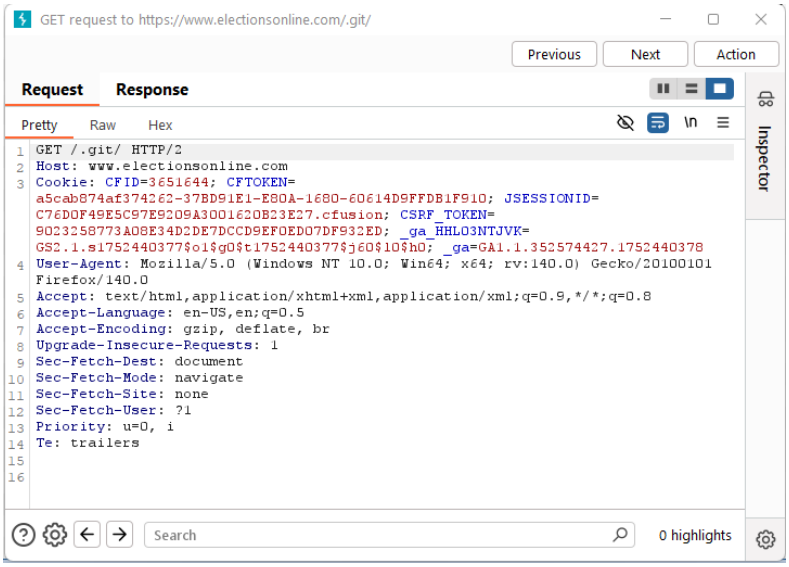
Status:

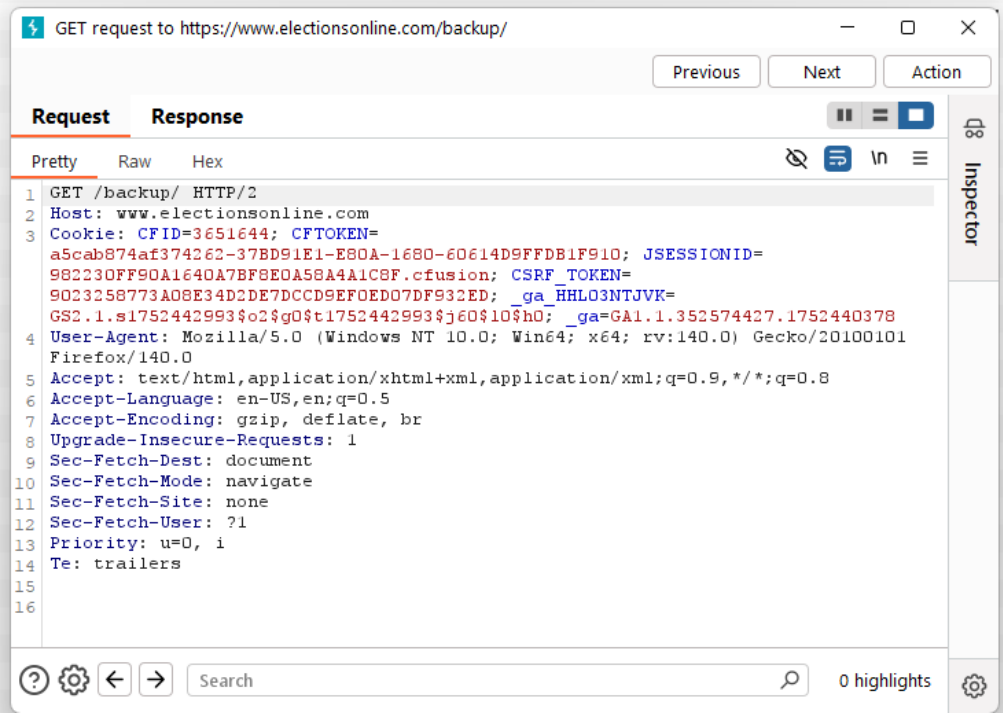
✅

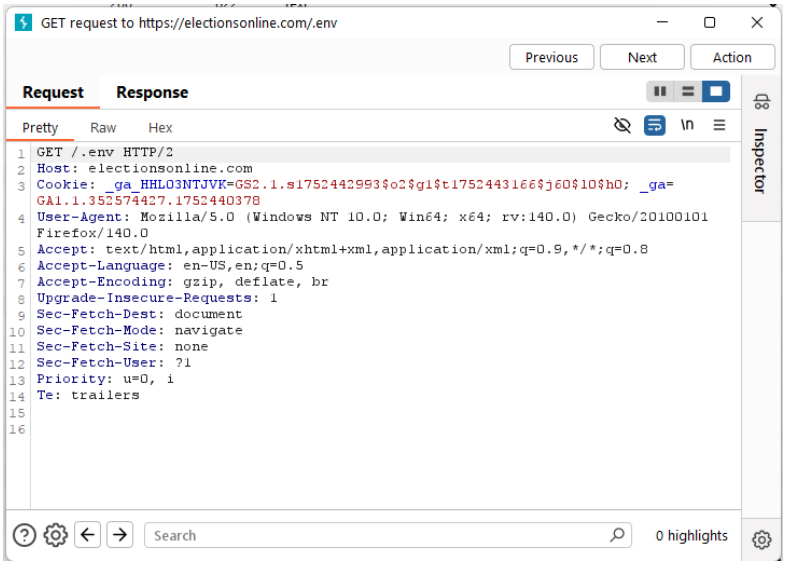
Notes:

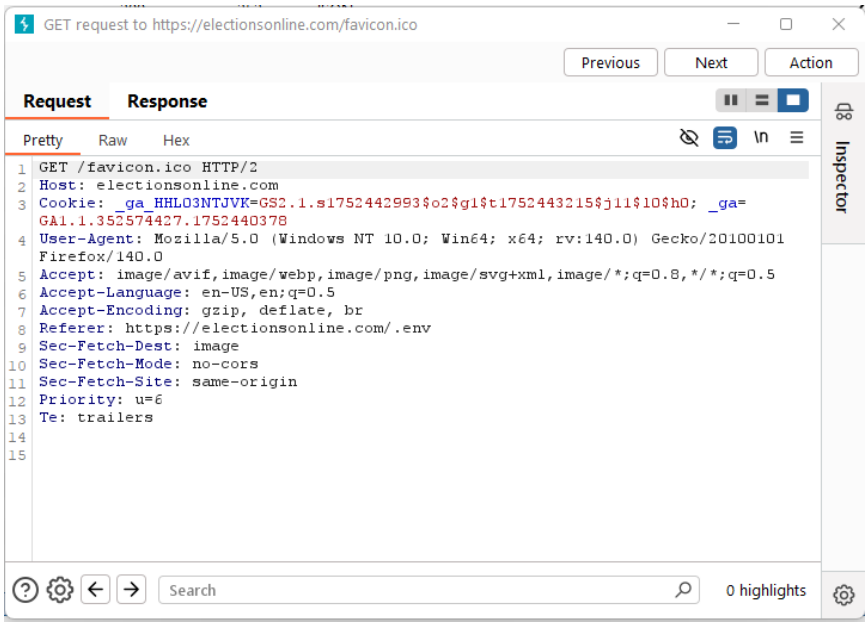
Inspected response headers using Burp Suite. X-Content-Type-Options: nosniff, Strict-Transport-Security were present. No Content-Security-Policy header was found. Some important headers are missing.

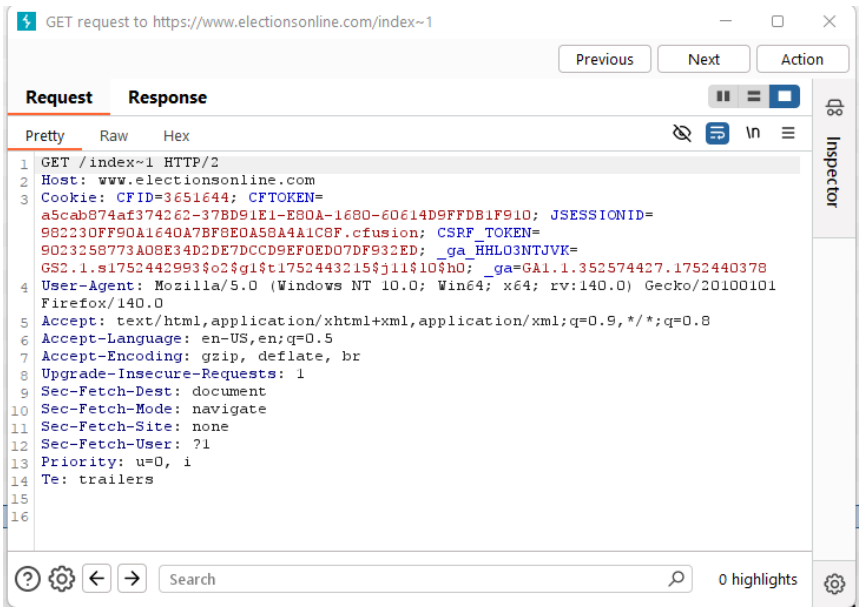


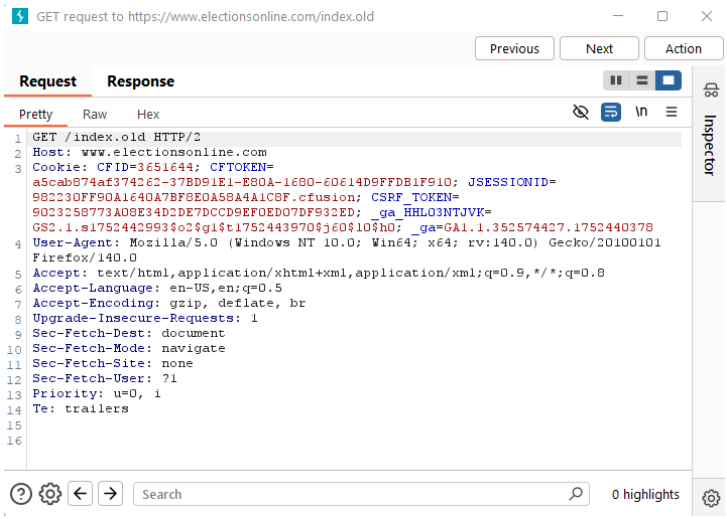



GET request to https://www.electionsonline.com/backup/
Previous
Next
Action
Request
Response
Pretty
Raw
Hex
Inspector
GET /backup/ HTTP/2
Host: www.electionsonline.com
Cookie: CFID=3651644; CFTOKEN=
a5cab874af374262-37BD91E1-E80A-1680-60614D9FFDB1F910; JSESSIONID=
98223OFF90A1640A7BF8E0A58A4A1C8F.cfusion; CSRF_TOKEN=
9023258773A08E34D2DE7DCCD9EF0ED07DF932ED; _ga_HHL03NTJVK=
GS2.1.s1752442993$o2$g0$t1752442993$j60$l0$h0; _ga=GA1.1.352574427.1752440378
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:140.0) Gecko/20100101
Firefox/140.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
Search
0 highlights


GET request to https://electionsonline.com/.env
Previous
Next
Action
Request
Response
Pretty
Raw
Hex
Inspector
GET /.env HTTP/2
Host: electionsonline.com
Cookie: _ga_HHL03NTJVK=GS2.1.s1752442993$o2$g1$t1752443166$j60$l0$h0; _ga=
GA1.1.352574427.1752440378
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:140.0) Gecko/20100101
Firefox/140.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
Search
0 highlights


GET request to https://electionsonline.com/favicon.ico
Previous
Next
Action
Request
Response
Pretty
Raw
Hex
Inspector
1 GET /favicon.ico HTTP/2
2 Host: electionsonline.com
3 Cookie: _ga_HHLO3NTJVK=GS2.1.s1752442993$o2$g1$t1752443215$j11$l0$h0; _ga=GA1.1.352574427.1752440378
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:140.0) Gecko/20100101 Firefox/140.0
5 Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://electionsonline.com/.env
9 Sec-Fetch-Dest: image
10 Sec-Fetch-Mode: no-cors
11 Sec-Fetch-Site: same-origin
12 Priority: u=6
13 Te: trailers
14
15
Search
0 highlights


GET request to https://www.electionsonline.com/index~1
Previous
Next
Action
Request
Response
Pretty
Raw
Hex
Inspector
1 GET /index~1 HTTP/2
2 Host: www.electionsonline.com
3 Cookie: CFID=3651644; CFTOKEN=a5cab874af374262-37BD91E1-E80A-1680-60614D9FFDB1F910; JSESSIONID=98223OFF90A164OA7BF8EOA58A4A1C8F.cfusion; CSRF_TOKEN=9023258773AO8E34D2DE7DCCD9EFOEDO7DF932ED; _ga_HHLO3NTJVK=GS2.1.s1752442993$o2$g1$t1752443215$j11$l0$h0; _ga=GA1.1.352574427.1752440378
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:140.0) Gecko/20100101 Firefox/140.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Upgrade-Insecure-Requests: 1
9 Sec-Fetch-Dest: document
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-Site: none
12 Sec-Fetch-User: ?1
13 Priority: u=0, i
14 Te: trailers
15
16
Search
0 highlights



Chapter: API and Web Service
Section: RESTful Web Service
Section ID: V13.2
Control ID: V13.2.5
Level: 2 and 3
CWE: 436

Control Description:

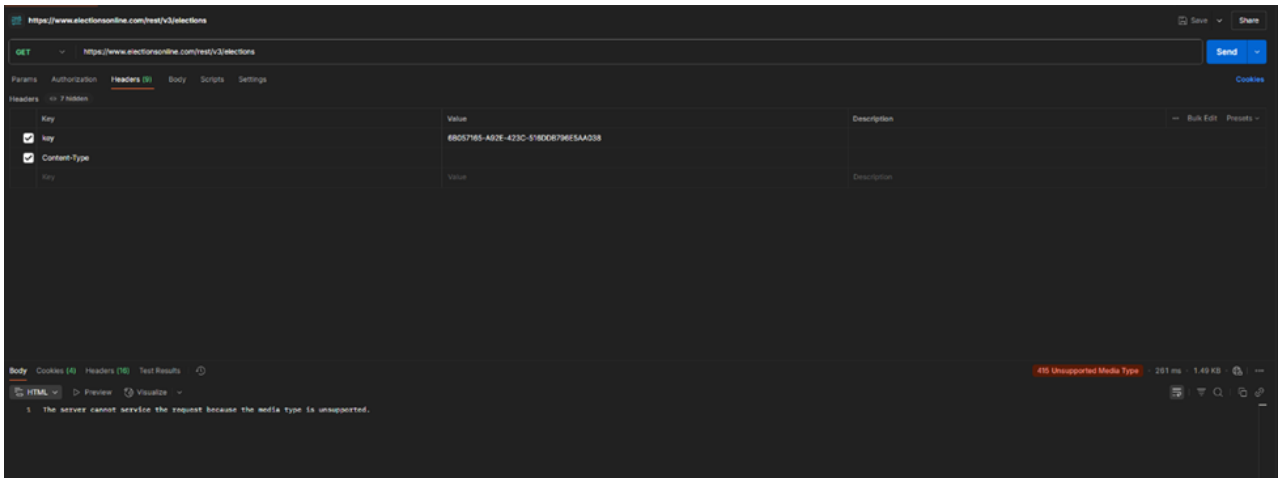
Verify that REST services explicitly check the incoming Content-Type to be the expected one, such as application/xml or application/json.

Status:

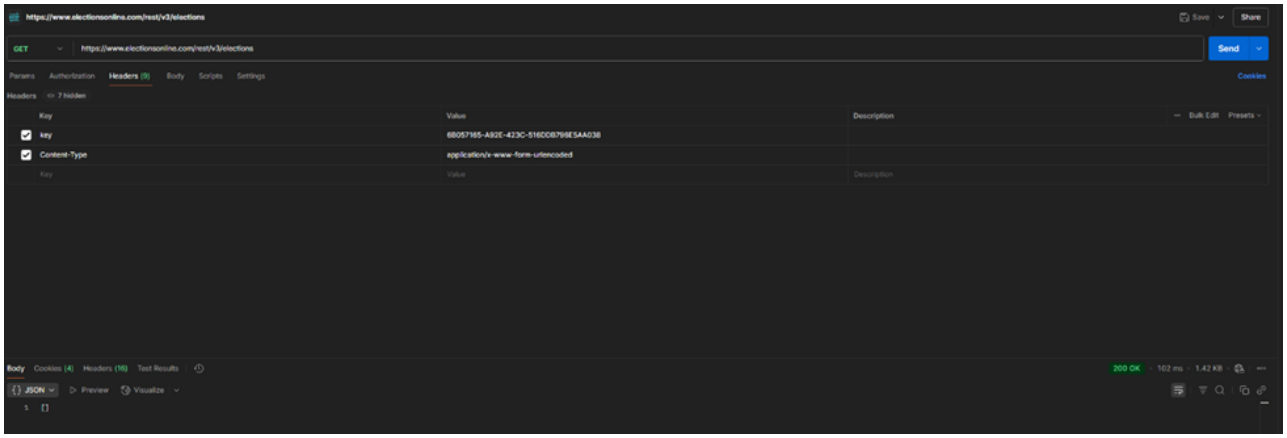
✅

Notes:

The application enforces strict Content-Type validation for its REST endpoints, rejecting invalid or missing values and only accepting expected formats.



Request with invalid Content-Type was rejected with 415 status.



Request with valid Content-Type processed successfully.

Chapter: API and Web Service
Section: SOAP Web Service
Section ID: V13.3
Control ID: V13.3.2
Level: 2 and 3
CWE: 345

Control Description:

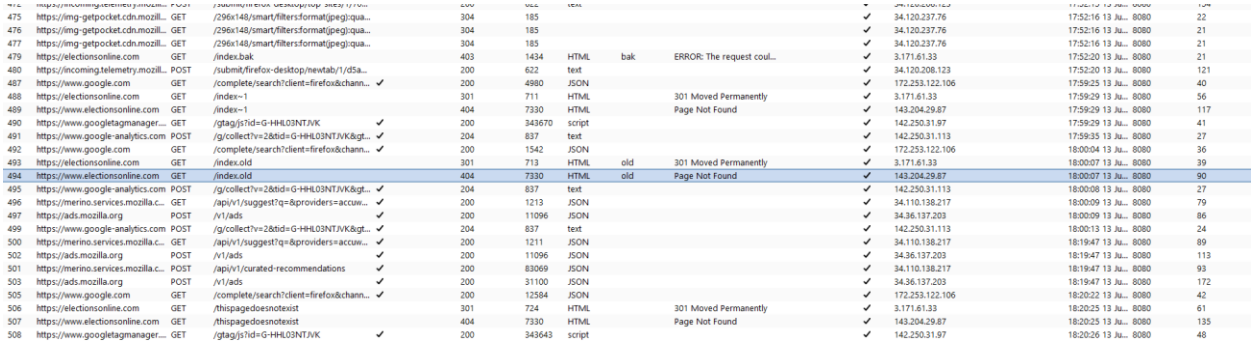
Verify that the message payload is signed using WS-Security to ensure reliable transport between client and service.

Status:

✅

Notes:

Observed HTTP response headers in Burp. Header X-Frame-Options: SAMEORIGIN is present. Clickjacking protection is enabled.



Chapter: API and Web Service
Section: RESTful Web Service
Section ID: V13.2
Control ID: V13.2.6
Level: 2 and 3
CWE: 345

Control Description:

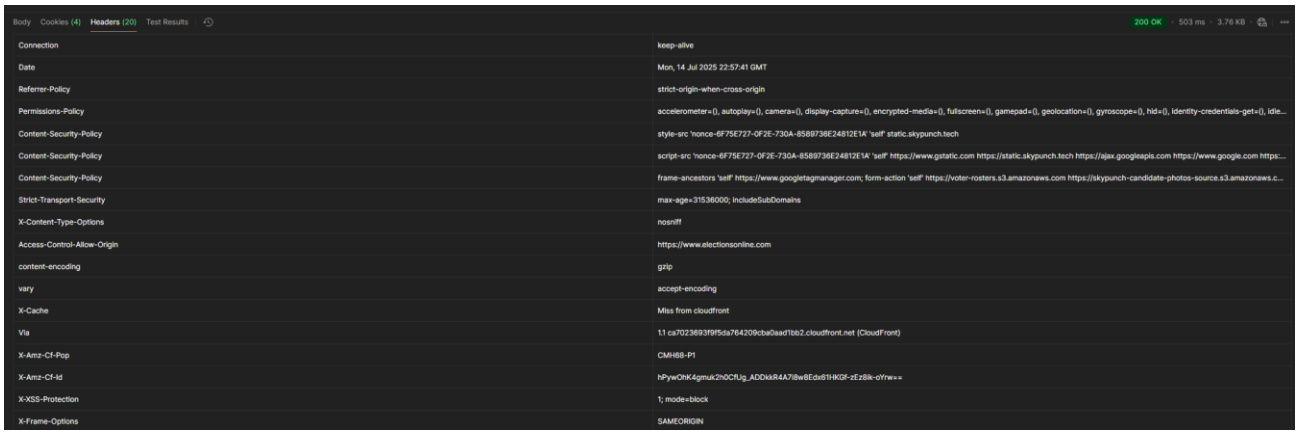
Verify that the message headers and payload are trustworthy and not modified in transit. Requiring strong encryption for transport (TLS only) may be sufficient in many cases as it provides both confidentiality and integrity protection. Per-message digital signatures can provide additional assurance on top of the transport protections for high-security applications but bring with them additional complexity and risks to weigh against the benefits.

Status:

✅

Notes:

The REST API enforces HTTPS transport, rejects malformed or suspicious input, and does not process untrusted content. Message headers and payloads are protected in transit and validated upon receipt.



Secure response headers confirming HTTPS transport and content protections for REST API communication.

For additional verification, the system uses TLS 1.3, the most up-to-date version. Using the curl commands below confirms that it utilizes the latest version and not the deprecated version 1.1:

```
$ curl https://electionsonline.com --verbose --tlsv1.1 --tls-max 1.1
```

```
* Host electionsonline.com:443 was resolved.
* IPv6: (none)
* IPv4: 108.157.150.116, 108.157.150.120, 108.157.150.90, 108.157.150.74
*   Trying 108.157.150.116:443...
* GnuTLS ciphers: NORMAL:-ARCFOUR-128:-CTYPE-ALL:+CTYPE-X509:-VERS-
SSL3.0:-VERS-TLS-ALL:+VERS-TLS1.1
* ALPN: curl offers h2,http/1.1
* found 152 certificates in /etc/ssl/certs/ca-certificates.crt
* found 458 certificates in /etc/ssl/certs
* GnuTLS, handshake failed: Error in protocol version
* closing connection #0
curl: (35) GnuTLS, handshake failed: Error in protocol version

$ curl https://electionsonline.com --verbose --tlsv1.2 --tls-max 1.2
* Host electionsonline.com:443 was resolved.
* IPv6: (none)
* IPv4: 108.157.150.120, 108.157.150.116, 108.157.150.74, 108.157.150.90
*   Trying 108.157.150.120:443...
* GnuTLS ciphers: NORMAL:-ARCFOUR-128:-CTYPE-ALL:+CTYPE-X509:-VERS-
SSL3.0:-VERS-TLS-ALL:+VERS-TLS1.2
* ALPN: curl offers h2,http/1.1
* found 152 certificates in /etc/ssl/certs/ca-certificates.crt
* found 458 certificates in /etc/ssl/certs
* SSL connection using TLS1.2 / ECDHE_RSA_AES_128_GCM_SHA256
*   server certificate verification OK
*   server certificate status verification SKIPPED
*   common name: www.electionsonline.com (matched)
*   server certificate expiration date OK
*   server certificate activation date OK
*   certificate public key: RSA
*   certificate version: #3
*   subject: CN=www.electionsonline.com
*   start date: Tue, 18 Mar 2025 00:00:00 GMT
*   expire date: Thu, 16 Apr 2026 23:59:59 GMT
*   issuer: C=US,O=Amazon,CN=Amazon RSA 2048 M03
```

```
* ALPN: server accepted h2
* Connected to electionsonline.com (108.157.150.120) port 443
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://electionsonline.com/
* [HTTP/2] [1] [:method: GET]
* [HTTP/2] [1] [:scheme: https]
* [HTTP/2] [1] [:authority: electionsonline.com]
* [HTTP/2] [1] [:path: /]
* [HTTP/2] [1] [user-agent: curl/8.12.1]
* [HTTP/2] [1] [accept: */*]
> GET / HTTP/2
> Host: electionsonline.com
> User-Agent: curl/8.12.1
> Accept: */*
>
* Request completely sent off
< HTTP/2 301
< content-type: text/html
< content-length: 134
< location: https://www.electionsonline.com:443/index.cfm
< server: awselb/2.0
< date: Tue, 05 Aug 2025 21:34:17 GMT
< x-cache: Miss from cloudfront
< via: 1.1 24948856b0f7ba2a78f73187b329c3d6.cloudfront.net (CloudFront)
< x-amz-cf-pop: MCI50-P2
< x-amz-cf-id: ZOJWsPMAqhXAIh1nmrPnlieMFJjz5kb9pRncSIxMCPXIUU0EVmPiBQ==
< x-xss-protection: 1; mode=block
< x-frame-options: SAMEORIGIN
< referrer-policy: strict-origin-when-cross-origin
< x-content-type-options: nosniff
< strict-transport-security: max-age=31536000
<
<html>
<head><title>301 Moved Permanently</title></head>
<body>
```

```
<center><h1>301 Moved Permanently</h1></center>
</body>
</html>
* Connection #0 to host electionsonline.com left intact
```

Chapter: API and Web Service
Section: SOAP Web Service
Control ID: V13.3.1
Level: 2 and 3
CWE: 20

Control Description:

Control ensures that applications performing XML-based data exchange properly validate the structure and content of incoming XML documents. Specifically, it requires that: XML input must be validated against a predefined XSD (XML Schema Definition) to ensure the structure is well-formed. Each field within the XML is also validated for type, length, format, and required presence before being processed.

Status:

✅ Not Applicable

Notes:

Testing Methodology:

Confirm the system enforces XSD schema validation on XML-based inputs before processing them, ensuring that: The XML structure conforms to a defined schema. Individual data fields are validated (e.g., proper types, lengths, formats).

Application does not utilize XML or require XSD schema validation.

No XML-based input endpoints were identified. All observed communication used JSON (application/json) or HTML. The application appears to rely exclusively on RESTful APIs with JSON payloads and does not process XML data.

Chapter: API and Web Service
Section: GraphQL

Section ID: V13.4
Control ID: V13.4.2
Level: 2 and 3
Control Description:
Verify that GraphQL or other data layer authorization logic should be implemented at the business logic layer instead of the GraphQL layer.

Status:
✅ N/A

Notes:
After reviewing the application and confirming with the administrator, I verified that GraphQL is not in use on the SkyPunch platform.

V14 Configuration

Chapter: Configuration
Section: Secure Deployment
Section ID: V14.1
Control ID: V14.1.4
Level: 2 and 3
CWE: 693
Control Description:
Verify that the application, configuration, and all dependencies can be re-deployed using automated deployment scripts, built from a documented and tested runbook in a reasonable time, or restored from backups in a timely fashion.

Status:
✅

Notes:
SkyPunch Technologies meets the requirements of OWASP ASVS 4.0.3 Control V14.1.4 through the use of automated tools and secure deployment practices. The "Web Application Security Policy – July 2025" provides evidence that all AWS Lambda functions use security scanning using Amazon Inspector prior to deployment. This combined with documented SDLC practices and references to infrastructure-as-code principles, demonstrates that deployments are both automated and structured, ensuring consistent and secure application delivery.

Chapter: Configuration
Section: Dependency
Section ID: V14.2
Control ID: V14.2.1
Level: 1, 2, and 3
CWE: 1026

Control Description:

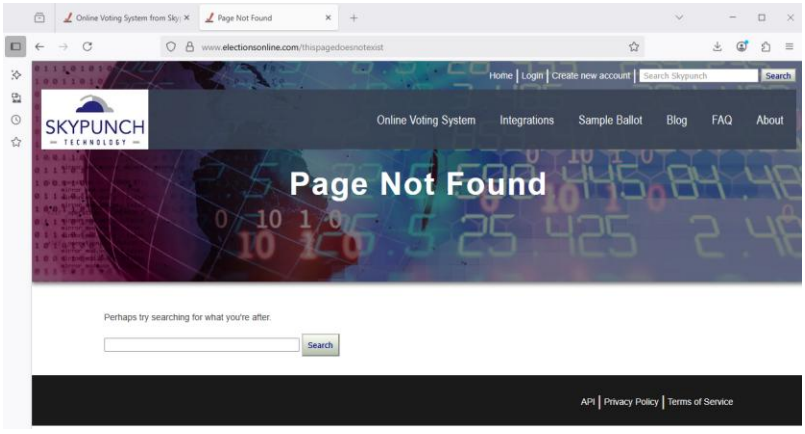
Verify that all components are up to date, preferably using a dependency checker during build or compile time.

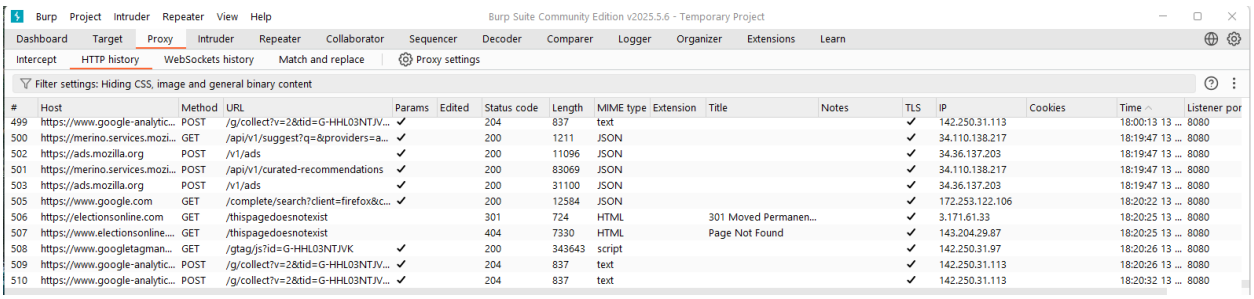
Response:

✅

Notes:

Visited broken or non-existent pages to trigger error responses. Clean 404 errors, no tracebacks or debug data revealed. Error handling is production-safe.





Chapter: Configuration
Section: Unintended Security Disclosure
Section ID: V14.3
Control ID: V14.3.3
Level: 1, 2, and 3
CWE: 200

Control Description:

Verify that the HTTP headers or any part of the HTTP response do not expose detailed version information of system components.

Status:

☑

Notes:

After invoking the command from Kali Linux curl -i electionsonline.com to provide information on the response header outputs, the below response does not reveal any detailed information about the system components (e.g., version number) other than the name of the server.

```
HTTP/1.1 301 Moved Permanently
Server: CloudFront
Date: Fri, 25 Jul 2025 20:22:15 GMT
Content-Type: text/html
Content-Length: 167
Connection: keep-alive
Location: https://electionsonline.com/
X-Cache: Redirect from cloudfront
Via: 1.1 cde38cee066c0b618a504717209d99b2.cloudfront.net (CloudFront)
X-Amz-Cf-Pop: MCI50-P2
X-Amz-Cf-Id: zzl56nHXlpxc_sKeIvFJC_9vLL9y1nymoZ36LaD83OgZk-C-o9Gvcw==
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Referrer-Policy: strict-origin-when-cross-origin
X-Content-Type-Options: nosniff

<html>
<head><title>301 Moved Permanently</title></head>
```

```
<body>
<center><h1>301 Moved Permanently</h1></center>
<hr><center>CloudFront</center>
</body>
</html>
```

Chapter: Configuration
Section: HTTP Security Headers
Section ID: V14.4
Control ID: V14.4.1
Level: 1, 2, and 3
CWE: 173

Control Description:

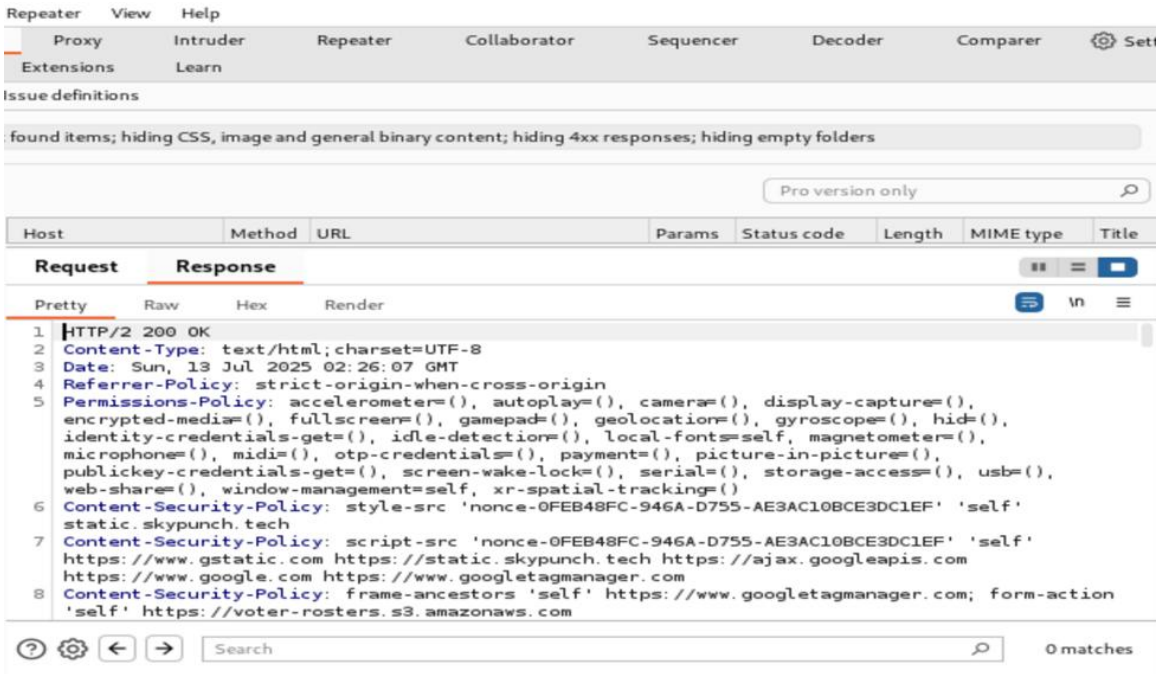
Verify that every HTTP response contains a Content-Type header. Also specify a safe character set (e.g., UTF-8, ISO-8859-1) if the content types are text/*, /+xml and application/xml. Content must match with the provided Content-Type header.

Status:

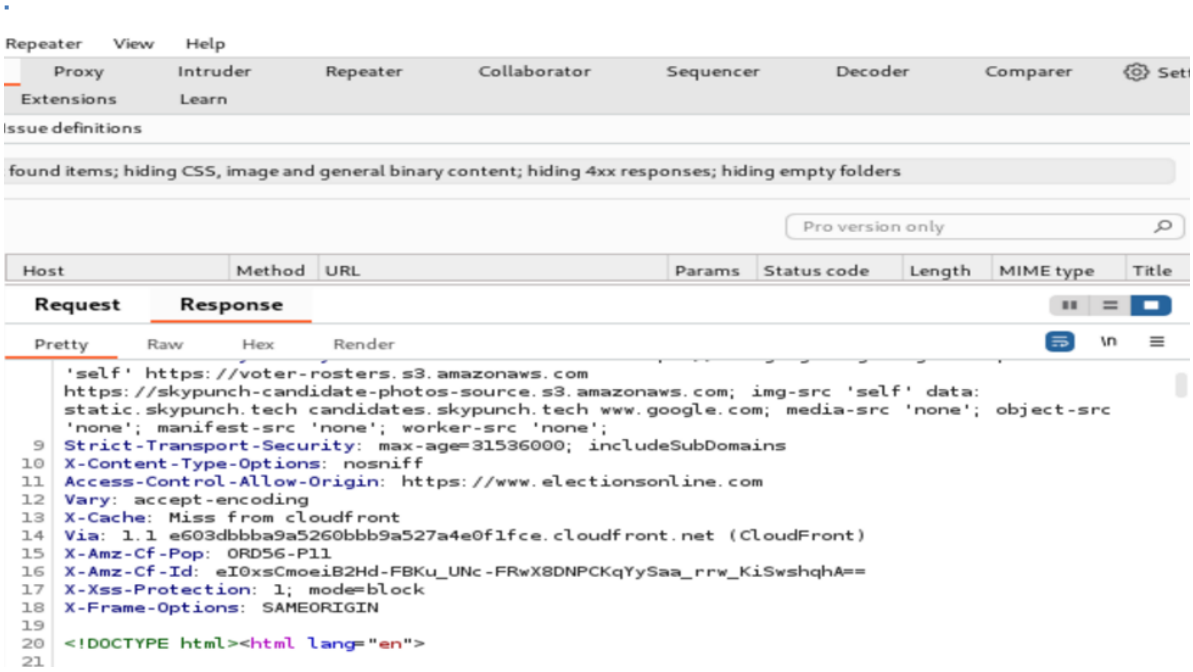
✅

Notes:

The application implements a strong set of secure HTTP response headers, all appropriately configured to enforce browser security protections.



Burp Suite view showing secure HTTP headers including CSP, HSTS, and X-Frame-Options.



Additional headers confirming X-XSS-Protection, Referrer-Policy, and Permissions-Policy.

Chapter: Configuration
Section: HTTP Security Headers
Section ID: V14.4
Control ID: V14.4.2
Level: 1, 2, and 3
CWE: 116

Control Description:

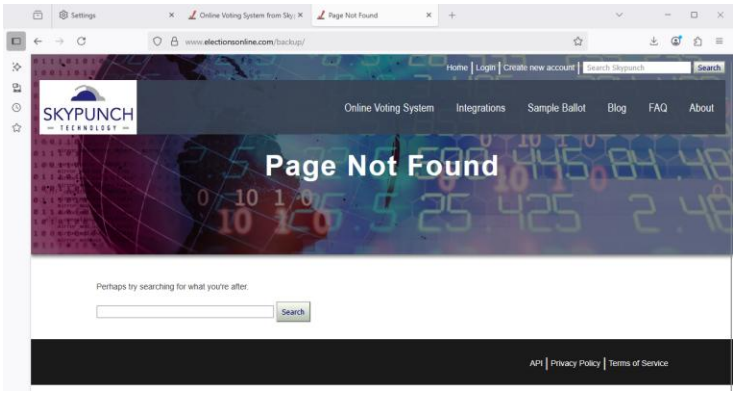
Verify that all API responses contain a Content-Disposition: attachment; filename="api.json" header (or other appropriate filename for the content type).

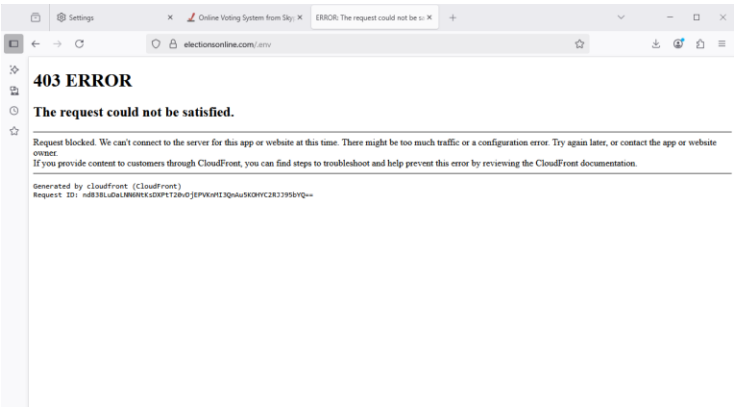
Status:

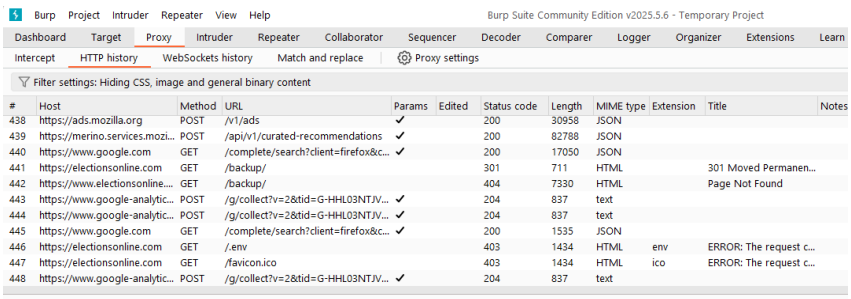
✅

Notes:

Attempted to access known directories like /admin/ and /images/ without specific filenames to observe if the server lists contents. In every case, the server returned a 403 Forbidden or a 404 Not Found error. No directory contents were revealed. This behavior shows that directory listing is properly disabled at the server level.







Chapter: Configuration
Section: HTTP Security Headers
Section ID: V14.4
Control ID: V14.4.3
Level: 1, 2, and 3
CWE: 1021
Control Description:

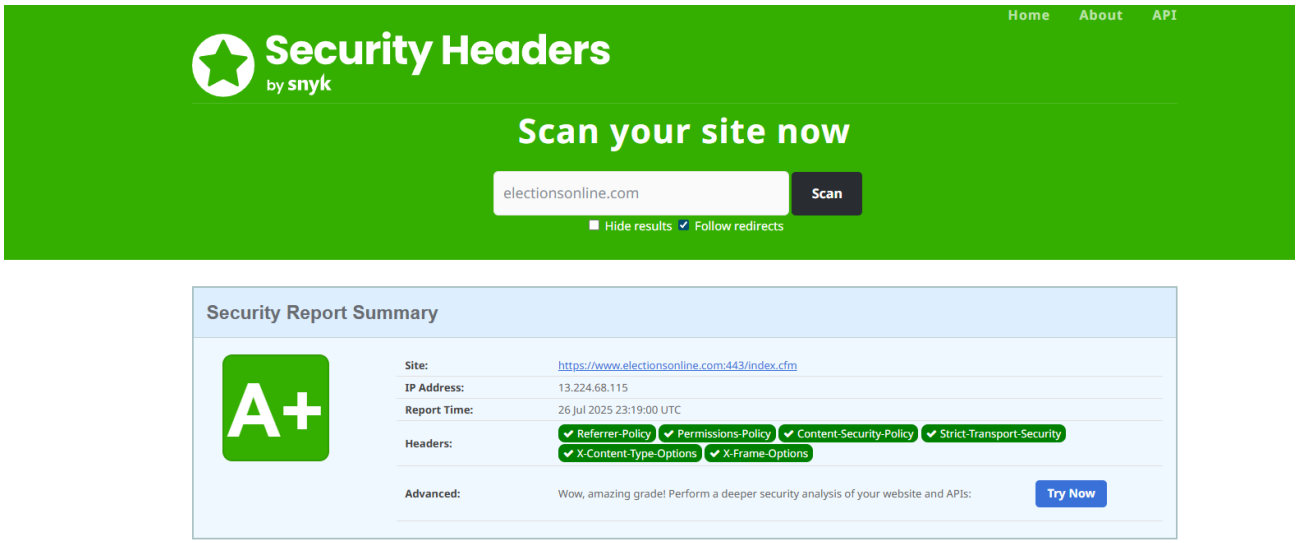
Verify that a Content Security Policy (CSP) response header is in place that helps mitigate impact for XSS attacks like HTML, DOM, JSON, and JavaScript injection vulnerabilities.

Status:

☑

Notes:

A scan from Security Headers confirms that Skypunch has a CSP response header in place, with an A+ security report summary (see below screenshot).



Chapter: Configuration
Section: HTTP Header Security
Section ID: V14.5
Control ID: V14.5.4
Level: 2 and 3
CWE: 345

Control Description:

Verify that HTTP headers added by a trusted proxy or SSO devices, such as a bearer token, are authenticated by the application.

Status:

Not Applicable (N/A)

Notes:

To test if the SkyPunch platform properly authenticates trusted HTTP headers, I did a penetration test using Postman. This involved injecting spoofed headers into requests to observe whether the application improperly trusted values from unverified sources.

Spoofed headers tested:

- X-Forwarded-User: admin@example.com
- Authorization: Bearer fake-token

- X-Auth-Token: fake-token-123

Tools used: Postman with custom header injection
Result: All test attempts returned a 404 Page Not Found response, indicating the endpoints were either invalid or inaccessible and the test was inconclusive.

Follow-up with Development Team:
David Simms confirmed:

- The system does not rely on authentication-related HTTP headers.
- A load balancer is in place but does not sanitize headers.
- Therefore, header validation is not relevant to the current architecture.

Conclusion:
Because the SkyPunch platform does not use authentication-related headers or process them in any security-sensitive way, Control V14.5.4 is not applicable in this environment.

Credits

Brandon Holt

- Introduction Section
- 2.1.1
- 2.2.1
- 2.5.6
- 2.8.1
- 2.9.3
- 3.3.1
- 3.4.2
- 5.1.3
- 6.4.1
- 10.2.1
- 13.3.1
- 13.1.5
- 14.4.3

Nicolas Davaille-Swinnen

- Introduction Section
- 1.4.4

- 1.4.5
- 2.2.6
- 2.5.3
- 3.3.4
- 5.1.4
- 5.1.5
- 7.4.3
- 8.2.2
- 8.2.3
- 10.2.2
- 11.1.8
- 13.2.6
- 14.3.3
- 14.4.3

Austin Miller

- Methodology
- 1.1.1
- 2.5.2
- 2.6.1
- 4.3.1
- 5.2.1
- 5.3.4
- 7.2.1
- 8.2.1
- 9.1.4
- 12.4.1
- 14.1.4
- 14.5.4

Benjamin McGavin

- Recommendations
- 1.14.4
- 1.14.5
- 2.2.4
- 3.4.1
- 5.3.1
- 5.3.2
- 5.4.1
- 5.4.2
- 5.4.3
- 7.4.1
- 9.1.2
- 11.1.4
- 13.2.5
- 13.2.6
- 14.4.1

Adetia McKenzie

- 2.3.3
- 3.4.3
- 4.2.1
- 6.1.1
- 7.1.1
- 11.1.7
- 13.2.1
- 13.3.2
- 13.4.2
- 14.2.1
- 14.4.2