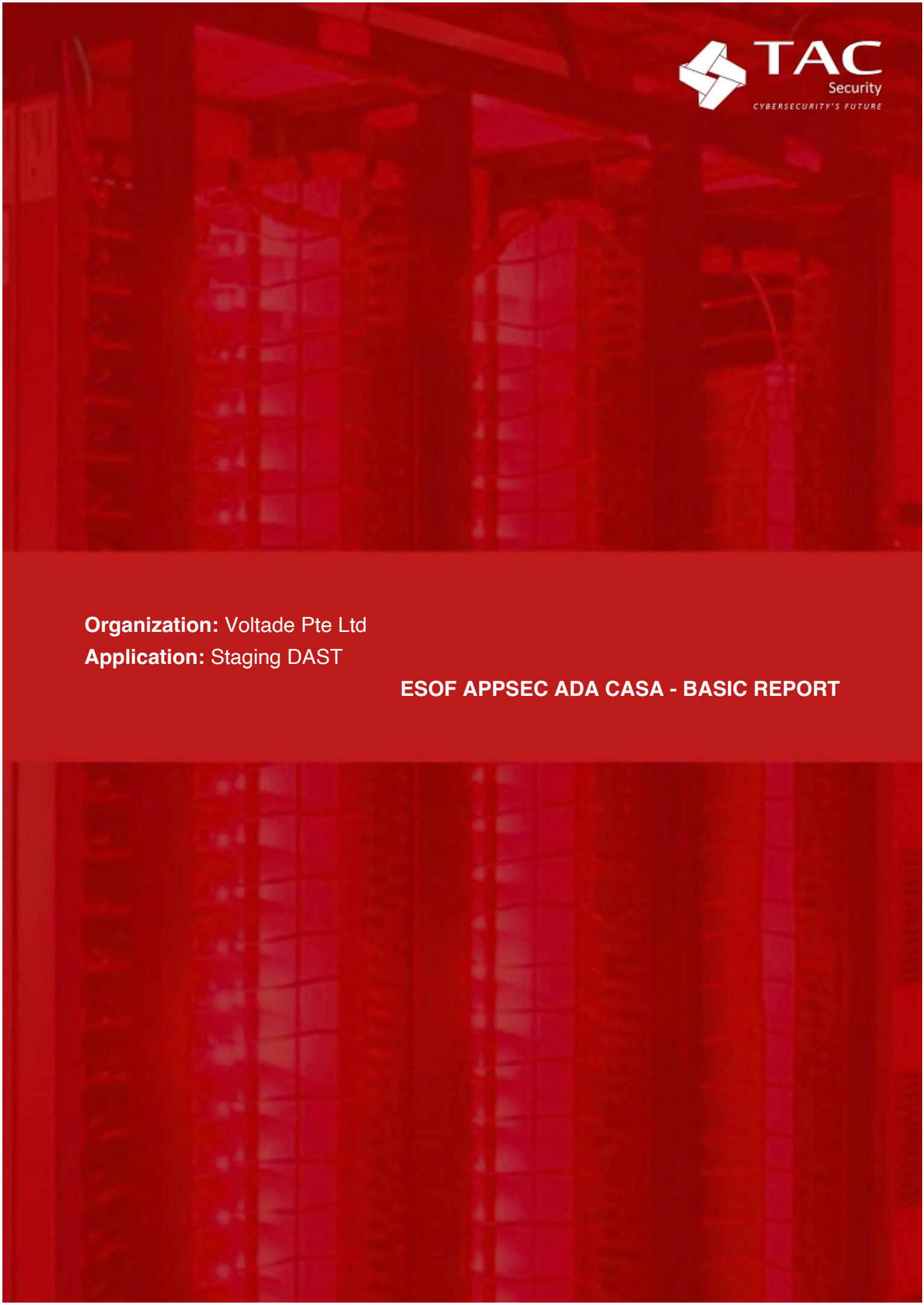


Organization: Voltade Pte Ltd

Application: Staging DAST

ESOF APPSEC ADA CASA - BASIC REPORT

Security is in Our DNA

1st Largest Auditor

Market share (almost 100%) of UPI assessment.

2nd Largest Financial Institution

Of countries application assessor.

3rd Largest telecom Company's

End to end security assessor.

Fortune 500

Oil and Gas company is protected by us.

10 Billion Transactions

Assessed on more than 200+ banking applications.

Top Fortune 500

Companies vulnerabilities have been managed by ESOF AppSec ADA.

Executive Summary

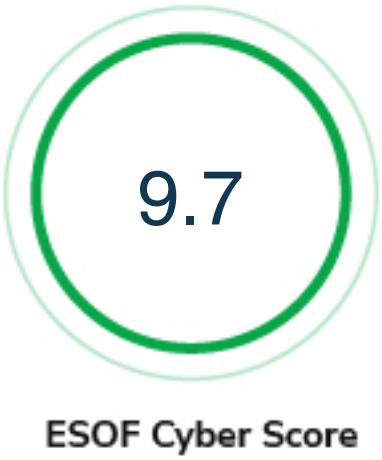
CASA has built upon the industry-recognized standards of the OWASP's Application Security Verification Standard (ASVS) to provide a consistent set of requirements to harden security for any application. Further, CASA provides a uniform way to perform trusted assurance assessments of these requirements when such assessments are required for applications with potential access to sensitive data.

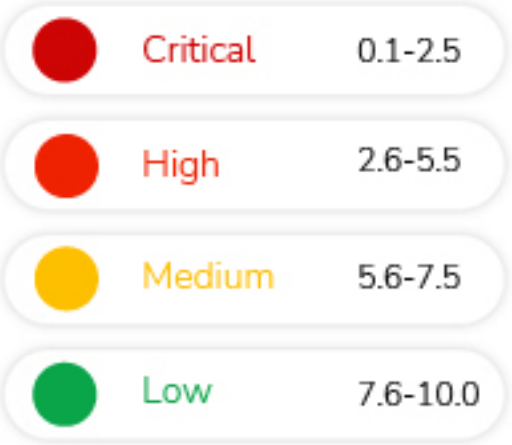
There is no "one size fits all solution" when it comes to evaluating application risk to securing user data. The CASA assessment acknowledges this reality and is adapted with a risk-based, multi-tier assessment approach to evaluate application risk based on user, scope, and other application specific items.

Risk Classification

Critical Risk	These vulnerabilities can allow attackers to take complete control of your web applications and web servers. In exploiting this type of vulnerability, attackers could carry out a range of malicious acts.
High Risk	A High severity vulnerability, which means that on exploiting such vulnerabilities, attackers, can view information about your system that helps them find or exploit other vulnerabilities that enable them to access sensitive user and administrator information.
Medium Risk	Potential weakness in controls, which could develop into an exposure. Or Issues that represent areas of concern and may impact controls. They should be addressed reasonably promptly.
Low Risk	Potential weaknesses in controls, which in combination with other weaknesses can develop into exposure. Suggested improvements not immediately/directly affecting controls.
Info Risk	Weaknesses mentioned under these sections are informational and are best practices. Either these weaknesses cannot be exploited directly or are very difficult to exploit due to multiple constrains.

ESOF AppSec ADA Cyber Score Classification





Target

- **Title:** Staging DAST
- **Source:** https://envoy-crm.voltade.sg

Testing Details

Start Date	May 5, 2025 13:21:56
Finish Date	May 16, 2025 09:10:13
Revalidation Date	May 16, 2025

Observation Summary

Sr. No.	Vulnerabilities	Severity	Status
1	Absence of Anti-CSRF Tokens	Low	Patched
2	Proxy Disclosure	Low	Patched
3	Permissions Policy Header Not Set	Info	Patched
4	Cookie without SameSite Attribute	Info	Patched
5	Session Management Response Identified	Info	Patched
6	Re-examine Cache-control Directives	Info	Patched
7	Non-Storable Content	Info	Patched
8	Cookie Slack Detector	Info	Patched
9	Strict-Transport-Security Header Not Set	Info	Patched
10	User Agent Fuzzer	Info	Patched

Note: All the vulnerabilities need to be patched to pass the CASA Tier 2 & Tier 3 assessments.

1 Absence of Anti-CSRF Tokens

Severity Level: Low

CWEID: 352

Status: Patched

Description:

No Anti-CSRF tokens were found in a HTML submission form.

A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.

CSRF attacks are effective in a number of situations, including:

* The victim has an active session on the target site.

* The victim is authenticated via HTTP auth on the target site.

* The victim is on the same local network as the target site.

CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.

Vulnerable URL

https://voltade.cloudflareaccess.com

Remedies:
Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
For example, use anti-CSRF packages such as the OWASP CSRFGuard.
Phase: Implementation
Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script.
Phase: Architecture and Design
Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330).
Note that this can be bypassed using XSS.
Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation.
Note that this can be bypassed using XSS.
Use the ESAPI Session Management control.
This control includes a component for CSRF.
Do not use the GET method for any request that triggers a state change.
Phase: Implementation
Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.

Evidence Url:

https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT

<form class="AuthFormLogin" action='https://voltade.cloudflareaccess.com/cdn-cgi/access/verify-code/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOTczZGUxOWYwIiwiYWxnIjoiUlMyNTYiLCJ0eXAiOiJKV1QifQ.eyJzZXJ2aWNlX3Rva2VuX3N0YXR1cyI6ZmFsc2UsImlhdCI6MTc0NjUxMDAwMywic2VydmljZV90b2tlbl9pZCI6IiIsImF1ZCI6IjllMWY1YzQ2ODY0YzMzMjYxMTZlZWEzZWQ4ZGQ2ZDdiNjQ0OTc1ODc4ZjZjMTE2NTA5MmU4ZDk3NTUxZTA2YmMiLCJob3N0bmFtZSI6ImVudm95LWNybS52b2x0YWRlLnNnIiwiYXBwX3Nlc3Npb25faGFzaCI6IjRmNzUyOGFkOGI2NTc3ZjUyN2E2NzcyYWJkMjBhNTFjZTQ1YWY0MDJjYjhiMzhlNjk5ODdiZjA2NjM3OGFhZWUiLCJuYmYiOjE3NDY1MTAwMDMsImlzX3dhcnAiOmZhbHNlLCJpc19nYXRld2F5IjpmYWxzZSwidHlwZSI6Im1ldGEiLCJyZWRpcmVjdF91cmwiOiJcLyIsIm10bHNfYXV0aCI6eyJjZXJ0X2lzc3Vlcl9za2kiOiIiLCJjZXJ0X3ByZXNlbnRlZCI6ZmFsc2UsImNlcnRfc2VyaWFsIjoiIiwiY2VydF9pc3N1ZXJfZG4iOiIiLCJhdXRoX3N0YXR1cyI6Ik5PTkUifSwiYXV0 aF9zdGF0dXMiOiJOT05FIn0.dbjoMmLWpJBLKzG2s3PS7-ENifh5xpCdeDrfHQmYN18j40Wc0w6p3GjSaXqp3haWrVikEXk ICIlo4BiFYRSXZSNlAI-0G90AF4rLg_z1mE6ImA5Qn7_tPIwt0mZ2_GP5tdIZFBBWgkMi4ZZ9QixHMwmj2vr3Fb4Dieutgc O6iQhATKqAwa__JuPXOAFjsbSh3Mzh6XBxbXHj75t_rfHC1N5zhKQ4kpvsI4VlPP1bFgwSyI_DCaC9r2wlyAIWPVyG_feuy 43GHHn1vSmnSZGVxNqu-OyhcZBmIghi66xtmUTdKJxfv_qlgSR2LSZPlHIx18WhddonXKXzntNndmQGkQ' method="post" id="totp-form">

Evidence Response:

No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "client_id" "connector_id" "connector_type" "email" "redirect_url"].

https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT

<form class="AuthFormLogin" action='https://voltade.cloudflareaccess.com/cdn-cgi/access/verify-code/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOTczZGUxOWYwIiwiYWxnIjoiUlMyNTYiLCJ0eXAiOiJKV1QifQ.eyJzZXJ2aWNlX3Rva2VuX3N0YXR1cyI6ZmFsc2UsImlhdCI6MTc0NzM4NTcwMSwic2VydmljZV90b2tlbl9pZCI6IiIsImF1ZCI6IjllMWY1YzQ2ODY0YzMzMjYxMTZlZWEzZWQ4ZGQ2ZDdiNjQ0OTc1ODc4ZjZjMTE2NTA5MmU4ZDk3NTUxZTA2YmMiLCJob3N0bmFtZSI6ImVudm95LWNybS52b2x0YWRlLnNnIiwiYXBwX3Nlc3Npb25faGFzaCI6IjUyNWQ1MDAwMzFlNzliZDQ5N2NlMzkzM2M1YjI4ZGRjZjUxOWU3MjliODljODE0YmYzNjU1MGZlODh

kOGIzNGQiLCJuYmYiOjE3NDczODU3MDEsImlzX3dhcnAiOmZhbHNlLCJpc19nYXRld2F5IjpmYWxzZSwidHlwZSI6Im11dG
EiLCJyZWRpcmVjdF91cmwiOiJcLyIsIm10bHNfYXV0aCI6eyJjZXJ0X2lzc3Vlcl9za2kiOiIiLCJjZXJ0X3ByZXNlbnRlZ
CI6ZmFsc2UsImNlcnRfc2VyaWFsIjoiIiwiY2VydF9pc3N1ZXJfZG4iOiIiLCJhdXRoX3N0YXR1cyI6Ik5PTkUifSwiYXV0 aF9zdGF0dXMiOiJOT05FIn0.HV2p6P0C8h5UP-JsE8-
OWDIBk3pWxdnk8kCpxkm0R4_k1ysbZTrXJO5JJCUWoSXFUwtptOM g57m8CZcz8Ns1pH-
sQbM7STzTxMmhlvH1lXVN7pgarRNKU8y_V-50KQ50UKuD-rAhwi5i8uLFZmrljcyMfidue1A3JBEhnt
LsDmNbrz9dIWo_uitkOaKhDpVCWTFpChbbRvQVYDN0HfxnRmqBBf3xbzn_Ij_tzTpiQlRRO3rucvHJoDCtdj6c4AOQVTj
ya kjFt0HRk2dNDZkN9U1VVpBMAyZeSjFX5gR-oE7g_Hcu_hF_pmpMA2jaJU2Ip6aJewDiTkhZ2j2hUc2Anw'
method="post" id="totp-form">
No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken,
csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf,
_csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form:
[Form 1: "client_id" "connector_id" "connector_type" "email" "redirect_url"].

Reference:
https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
https://cwe.mitre.org/data/definitions/352.html

2 Proxy Disclosure

Severity Level: Low

CWEID: 200

Status: Patched

Description:

1 proxy server(s) were detected or fingerprinted. This information helps a potential attacker to determine

- A list of targets for an attack against the application.

- Potential vulnerabilities on the proxy servers that service the application.

- The presence or absence of any proxy-based components that might cause attacks against the application to be detected, prevented, or mitigated.

Vulnerable URL

https://envoy-crm.voltade.sg

Remedies:

Disable the 'TRACE' method on the proxy servers, as well as the origin web/application server.
Disable the 'OPTIONS' method on the proxy servers, as well as the origin web/application server, if it is not required for other purposes, such as 'CORS' (Cross Origin Resource Sharing).
Configure the web and application servers with custom error pages, to prevent 'fingerprintable' product-specific error pages being leaked to the user in the event of HTTP errors, such as 'TRACK' requests for non-existent pages.
Configure all proxies, application servers, and web servers to prevent disclosure of the technology and version information in the 'Server' and 'X-Powered-By' HTTP response headers.

Evidence Url:

```
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT
username
```

Evidence Response:

```
The following pattern was used: \bUSERNAME\b and was detected in the element starting with: "<script>!function(e){"use strict";function t(t){return!!t&&("Symbol"in e&&"iterator"in e.Symbol&&"function"==typeof t[Symbol.ite", see evidence field for the suspicious comment/snippet.
https://envoy-crm.voltade.sg
Using the TRACE, OPTIONS, and TRACK methods, the following proxy servers have been identified between ZAP and the application/web server: - cloudflare The following web/application server has been identified: - cloudflare
https://envoy-crm.voltade.sg/
Using the TRACE, OPTIONS, and TRACK methods, the following proxy servers have been identified between ZAP and the application/web server: - cloudflare The following web/application server has been identified: - cloudflare
```

https://envoy-crm.voltade.sg/robots.txt
Using the TRACE, OPTIONS, and TRACK methods, the following proxy servers have been identified between ZAP and the application/web server: - cloudflare The following web/application server has been identified: - cloudflare
https://envoy-crm.voltade.sg/sitemap.xml
Using the TRACE, OPTIONS, and TRACK methods, the following proxy servers have been identified between ZAP and the application/web server: - cloudflare The following web/application server has been identified: - cloudflare
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT
username
The following pattern was used: \bUSERNAME\b and was detected in the element starting with: "<script>!function(e){"use strict";function t(t){return!!t&&("Symbol"in e&&"iterator"in e.Symbol&&"function"==typeof t[Symbol.ite", see evidence field for the suspicious comment/snippet.
https://envoy-crm.voltade.sg
Using the TRACE, OPTIONS, and TRACK methods, the following proxy servers have been identified between ZAP and the application/web server: - cloudflare The following web/application server has been identified: - cloudflare
https://envoy-crm.voltade.sg/
Using the TRACE, OPTIONS, and TRACK methods, the following proxy servers have been identified between ZAP and the application/web server: - cloudflare The following web/application server has been identified: - cloudflare
https://envoy-crm.voltade.sg/robots.txt
Using the TRACE, OPTIONS, and TRACK methods, the following proxy servers have been identified between ZAP and the application/web server: - cloudflare The following web/application server has been identified: - cloudflare
https://envoy-crm.voltade.sg/sitemap.xml
Using the TRACE, OPTIONS, and TRACK methods, the following proxy servers have been identified between ZAP and the application/web server: - cloudflare The following web/application server has been identified: - cloudflare

Reference:
https://tools.ietf.org/html/rfc7231#section-5.1.2

3 Permissions Policy Header Not Set

Severity Level: Info

CWEID: 693

Status: Patched

Description:

Permissions Policy Header is an added layer of security that helps to restrict from unauthorized access or usage of browser/client features by web resources. This policy ensures the user privacy by limiting or specifying the features of the browsers can be used by the web resources. Permissions Policy provides a set of standard HTTP headers that allow website owners to limit which features of browsers can be used by the page such as camera, microphone, location, full screen etc.

Vulnerable URL

https://voltade.cloudflareaccess.com

Remedies:

Ensure that your web server, application server, load balancer, etc. is configured to set the Permissions-Policy header.

Evidence Url:

```
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT
frame-ancestors 'none'; connect-src 'self' http://127.0.0.1:*; default-src https: 'unsafe-inline'
```

Evidence Response:

```
The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: script-src, style-src, img-src, frame-src, font-src, media-src, object-src, manifest-src, worker-src, form-action The directive(s): form-action are among the directives that do not fallback to default-src, missing/excluding them is the same as allowing anything.
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT
frame-ancestors 'none'; connect-src 'self' http://127.0.0.1:*; default-src https: 'unsafe-inline'
script-src includes unsafe-inline.
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT
frame-ancestors 'none'; connect-src 'self' http://127.0.0.1:*; default-src https: 'unsafe-inline'
```

style-src includes unsafe-inline.
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT
frame-ancestors 'none'; connect-src 'self' http://127.0.0.1:; default-src https: 'unsafe-inline'*
The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: script-src, style-src, img-src, frame-src, font-src, media-src, object-src, manifest-src, worker-src, form-action The directive(s): form-action are among the directives that do not fallback to default-src, missing/excluding them is the same as allowing anything.
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT
frame-ancestors 'none'; connect-src 'self' http://127.0.0.1:; default-src https: 'unsafe-inline'*
script-src includes unsafe-inline.
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT
frame-ancestors 'none'; connect-src 'self' http://127.0.0.1:; default-src https: 'unsafe-inline'*
style-src includes unsafe-inline.
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT

Reference:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Permissions-Policy
https://developer.chrome.com/blog/feature-policy/
https://scotthelme.co.uk/a-new-security-header-feature-policy/
https://w3c.github.io/webappsec-feature-policy/
https://www.smashingmagazine.com/2018/12/feature-policy/

4 Cookie without SameSite Attribute

Severity Level: Info

CWEID: 1275

Status: Patched

Description:

A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.

Vulnerable URL

https://envoy-crm.voltade.sg

Remedies:

Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.

Evidence Url:

```
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?
kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redire
ct_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY
0MTUwM2Q3NmNhMzg4N2JiM2RjOT
Set-Cookie: CF_Session
https://envoy-crm.voltade.sg
Set-Cookie: CF_AppSession
https://envoy-crm.voltade.sg/
Set-Cookie: CF_AppSession
https://envoy-crm.voltade.sg/robots.txt
Set-Cookie: CF_AppSession
https://envoy-crm.voltade.sg/sitemap.xml
Set-Cookie: CF_AppSession
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?
kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redire
ct_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY
0MTUwM2Q3NmNhMzg4N2JiM2RjOT
Set-Cookie: CF_Session
https://envoy-crm.voltade.sg
Set-Cookie: CF_AppSession
https://envoy-crm.voltade.sg/
Set-Cookie: CF_AppSession
https://envoy-crm.voltade.sg/robots.txt
Set-Cookie: CF_AppSession
https://envoy-crm.voltade.sg/sitemap.xml
Set-Cookie: CF_AppSession
```

Reference:

https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site

5 Session Management Response Identified

Severity Level: Info

CWEID: -1

Status: Patched

Description:

The given response has been identified as containing a session management token. The 'Other Info' field contains a set of header tokens that can be used in the Header Based Session Management Method. If the request is in a context which has a Session Management Method set to "Auto-Detect" then this rule will change the session management to use the tokens identified.

Vulnerable URL

https://envoy-crm.voltade.sg

Remedies:

This is an informational alert rather than a vulnerability and so there is nothing to fix.

Evidence Url:

```
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?
kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redire
ct_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY
0MTUwM2Q3NmNhMzg4N2JiM2RjOT
<a class="OrgAvatarLink" href=''> <div class="OrgAvatarLink-logo"> <img height=50
src="https:&#x2F;&#x2F;avatars.githubusercontent.com&#x2F;u&#x2F;124495972
?s&#x3D;200&amp;amp;v&#x3D;4" style="display:none;" onload="this.style.display=''"></img>
</div> <div class="OrgAvatarLink-title"> Voltade </div> </a>
```

Evidence Response:

```
Links have been found that do not have traditional href attributes, which is an indication
that this is a modern web application.
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?
kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redire
ct_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY
0MTUwM2Q3NmNhMzg4N2JiM2RjOT
nUBFeYbfIu7AxuriH
cookie:CF_Session
https://envoy-crm.voltade.sg
n35bfece43e24ef87
cookie:CF_AppSession
https://envoy-crm.voltade.sg/
n055a649a35858a63
cookie:CF_AppSession
https://envoy-crm.voltade.sg/
n0dbdbbdc8daa2ce4
cookie:CF_AppSession
https://envoy-crm.voltade.sg/robots.txt
```

```
nb1c4df9c945c9697
cookie:CF_AppSession
https://envoy-crm.voltade.sg/sitemap.xml
n152b965985c2124b
cookie:CF_AppSession
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?
kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redire
ct_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY
0MTUwM2Q3NmNhMzg4N2JiM2RjOT
<a class="OrgAvatarLink" href=''> <div class="OrgAvatarLink-logo"> <img height=50
src="https:&#x2F;&#x2F;avatars.githubusercontent.com&#x2F;u&#x2F;124495972
?s&#x3D;200&amp;amp;v&#x3D;4" style="display:none;" onload="this.style.display=''"></img>
</div> <div class="OrgAvatarLink-title"> Voltade </div> </a>
Links have been found that do not have traditional href attributes, which is an indication
that this is a modern web application.
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?
kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redire
ct_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY
0MTUwM2Q3NmNhMzg4N2JiM2RjOT
nq47Ilk4mnyk45mYS
cookie:CF_Session
https://envoy-crm.voltade.sg
n2166f97218780097
cookie:CF_AppSession
https://envoy-crm.voltade.sg/
n055cc713986e2b23
cookie:CF_AppSession
https://envoy-crm.voltade.sg/
n0b803c22e209db40
cookie:CF_AppSession
https://envoy-crm.voltade.sg/robots.txt
n0b870e3341a4512a
cookie:CF_AppSession
https://envoy-crm.voltade.sg/sitemap.xml
na2c067d179ccea74
cookie:CF_AppSession
```

Reference:
https://www.zaproxy.org/docs/desktop/addons/authentication-helper/session-mgmt-id

6 Re-examine Cache-control Directives

Severity Level: Info

CWEID: 525

Status: Patched

Description:

The cache-control header has not been set properly or is missing, allowing the browser and proxies to cache content. For static assets like css, js, or image files this might be intended, however, the resources should be reviewed to ensure that no sensitive content will be cached.

Vulnerable URL

https://voltade.cloudflareaccess.com

Remedies:

For secure content, ensure the cache-control HTTP header is set with "no-cache, no-store, must-revalidate". If an asset should be cached consider setting the directives "public, max-age, immutable".

Evidence Url:

https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT

https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT

Reference:

https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html#web-content-caching
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control
https://grayduck.mn/2021/09/13/cache-control-recommendations/

7 Non-Storable Content

Severity Level: Info

CWEID: 524

Status: Patched

Description:

The response contents are not storable by caching components such as proxy servers. If the response does not contain sensitive, personal or user-specific information, it may benefit from being stored and cached, to improve performance.

Vulnerable URL

https://envoy-crm.voltade.sg

Remedies:

The content may be marked as storable by ensuring that the following conditions are satisfied:
The request method must be understood by the cache and defined as being cacheable ("GET", "HEAD", and "POST" are currently defined as cacheable)
The response status code must be understood by the cache (one of the 1XX, 2XX, 3XX, 4XX, or 5XX response classes are generally understood)
The "no-store" cache directive must not appear in the request or response header fields
For caching by "shared" caches such as "proxy" caches, the "private" response directive must not appear in the response
For caching by "shared" caches such as "proxy" caches, the "Authorization" header field must not appear in the request, unless the response explicitly allows it (using one of the "must-revalidate", "public", or "s-maxage" Cache-Control response directives)
In addition to the conditions above, at least one of the following conditions must also be satisfied by the response:
It must contain an "Expires" header field
It must contain a "max-age" response directive
For "shared" caches such as "proxy" caches, it must contain a "s-maxage" response directive
It must contain a "Cache Control Extension" that allows it to be cached
It must have a status code that is defined as cacheable by default (200, 203, 204, 206, 300, 301, 404, 405, 410, 414, 501).

Evidence Url:

```
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT
```

Evidence Response:

```
In the absence of an explicitly specified caching lifetime directive in the response, a liberal lifetime heuristic of 1 year was assumed. This is permitted by rfc7234.
https://envoy-crm.voltade.sg
no-store
https://envoy-crm.voltade.sg/
no-store
https://envoy-crm.voltade.sg/robots.txt
```

no-store
https://envoy-crm.voltade.sg/sitemap.xml
no-store
https://voltade.cloudflareaccess.com/cdn-cgi/access/login/envoy-crm.voltade.sg?kid=9e1f5c46864c3326116eea3ed8dd6d7b644975878f6c1165092e8d97551e06bc&redirect_url=%2F&meta=eyJraWQiOiJkM2Q2NWViMjQ4OTI1MDQzNjY5ZjYxMWFmOWEyNzkzNWJlYWY0MTUwM2Q3NmNhMzg4N2JiM2RjOT
In the absence of an explicitly specified caching lifetime directive in the response, a liberal lifetime heuristic of 1 year was assumed. This is permitted by rfc7234.
https://envoy-crm.voltade.sg
no-store
https://envoy-crm.voltade.sg/
no-store
https://envoy-crm.voltade.sg/robots.txt
no-store
https://envoy-crm.voltade.sg/sitemap.xml
no-store

Reference:
https://datatracker.ietf.org/doc/html/rfc7234
https://datatracker.ietf.org/doc/html/rfc7231
https://www.w3.org/Protocols/rfc2616/rfc2616-sec13.html

8 Cookie Slack Detector

Severity Level: Info

CWEID: 205

Status: Patched

Description:

Repeated GET requests: drop a different cookie each time, followed by normal request with all cookies to stabilize session, compare responses against original baseline GET. This can reveal areas where cookie based authentication/attributes are not actually enforced.

Vulnerable URL

https://envoy-crm.voltade.sg

Evidence Url:

```
https://envoy-crm.voltade.sg
```

Evidence Response:

```
NOTE: Because of its name this cookie may be important, but dropping it appears to have no
effect: [CF_AppSession] Cookies that don't have expected effects can reveal flaws in
application logic. In the worst case, this can reveal where authentication via cookie
token(s) is not actually enforced. These cookies affected the response: These cookies did NOT
affect the response: CF_AppSession
https://envoy-crm.voltade.sg/
NOTE: Because of its name this cookie may be important, but dropping it appears to have no
effect: [CF_AppSession] Cookies that don't have expected effects can reveal flaws in
application logic. In the worst case, this can reveal where authentication via cookie
token(s) is not actually enforced. These cookies affected the response: These cookies did NOT
affect the response: CF_AppSession
https://envoy-crm.voltade.sg/robots.txt
NOTE: Because of its name this cookie may be important, but dropping it appears to have no
effect: [CF_AppSession] Cookies that don't have expected effects can reveal flaws in
application logic. In the worst case, this can reveal where authentication via cookie
token(s) is not actually enforced. These cookies affected the response: These cookies did NOT
affect the response: CF_AppSession
https://envoy-crm.voltade.sg/sitemap.xml
NOTE: Because of its name this cookie may be important, but dropping it appears to have no
effect: [CF_AppSession] Cookies that don't have expected effects can reveal flaws in
application logic. In the worst case, this can reveal where authentication via cookie
token(s) is not actually enforced. These cookies affected the response: These cookies did NOT
affect the response: CF_AppSession
https://envoy-crm.voltade.sg
NOTE: Because of its name this cookie may be important, but dropping it appears to have no
effect: [CF_AppSession] Cookies that don't have expected effects can reveal flaws in
application logic. In the worst case, this can reveal where authentication via cookie
token(s) is not actually enforced. These cookies affected the response: These cookies did NOT
affect the response: CF_AppSession
```

https://envoy-crm.voltade.sg/
NOTE: Because of its name this cookie may be important, but dropping it appears to have no effect: [CF_AppSession] Cookies that don't have expected effects can reveal flaws in application logic. In the worst case, this can reveal where authentication via cookie token(s) is not actually enforced. These cookies affected the response: These cookies did NOT affect the response: CF_AppSession
https://envoy-crm.voltade.sg/robots.txt
NOTE: Because of its name this cookie may be important, but dropping it appears to have no effect: [CF_AppSession] Cookies that don't have expected effects can reveal flaws in application logic. In the worst case, this can reveal where authentication via cookie token(s) is not actually enforced. These cookies affected the response: These cookies did NOT affect the response: CF_AppSession
https://envoy-crm.voltade.sg/sitemap.xml
NOTE: Because of its name this cookie may be important, but dropping it appears to have no effect: [CF_AppSession] Cookies that don't have expected effects can reveal flaws in application logic. In the worst case, this can reveal where authentication via cookie token(s) is not actually enforced. These cookies affected the response: These cookies did NOT affect the response: CF_AppSession

Reference:
https://cwe.mitre.org/data/definitions/205.html

9 Strict-Transport-Security Header Not Set

Severity Level: Info

CWEID: 319

Status: Patched

Description:

HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797.

Vulnerable URL

https://envoy-crm.voltade.sg

Remedies:

Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.

Evidence Url:

`https://envoy-crm.voltade.sg`
`https://envoy-crm.voltade.sg/`
`https://envoy-crm.voltade.sg/robots.txt`
`https://envoy-crm.voltade.sg/sitemap.xml`

Reference:

https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html
https://owasp.org/www-community/Security_Headers
https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security
https://caniuse.com/stricttransportsecurity
https://datatracker.ietf.org/doc/html/rfc6797

10 User Agent Fuzzer

Severity Level: Info

Status: Patched

Description:

Check for differences in response based on fuzzed User Agent (eg. mobile sites, access as a Search Engine Crawler). Compares the response statuscode and the hashcode of the response body with the original response.

Vulnerable URL

https://envoy-crm.voltade.sg

Evidence Url:

`https://envoy-crm.voltade.sg`
`https://envoy-crm.voltade.sg`
`https://envoy-crm.voltade.sg`
`https://envoy-crm.voltade.sg`
`https://envoy-crm.voltade.sg`
`https://envoy-crm.voltade.sg`
`https://envoy-crm.voltade.sg`
`https://envoy-crm.voltade.sg`
`https://envoy-crm.voltade.sg`
`https://envoy-crm.voltade.sg`
`https://envoy-crm.voltade.sg`
`https://envoy-crm.voltade.sg`
`https://envoy-crm.voltade.sg/`
`https://envoy-crm.voltade.sg/`
`https://envoy-crm.voltade.sg/`
`https://envoy-crm.voltade.sg/`
`https://envoy-crm.voltade.sg/`
`https://envoy-crm.voltade.sg/`
`https://envoy-crm.voltade.sg/`
`https://envoy-crm.voltade.sg/`
`https://envoy-crm.voltade.sg/`
`https://envoy-crm.voltade.sg/`
`https://envoy-crm.voltade.sg/`
`https://envoy-crm.voltade.sg/robots.txt`
`https://envoy-crm.voltade.sg/robots.txt`
`https://envoy-crm.voltade.sg/robots.txt`
`https://envoy-crm.voltade.sg/robots.txt`
`https://envoy-crm.voltade.sg/robots.txt`
`https://envoy-crm.voltade.sg/robots.txt`
`https://envoy-crm.voltade.sg/robots.txt`
`https://envoy-crm.voltade.sg/robots.txt`
`https://envoy-crm.voltade.sg/robots.txt`
`https://envoy-crm.voltade.sg/robots.txt`

https://envoy-crm.voltade.sg/robots.txt
https://envoy-crm.voltade.sg/robots.txt
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg
https://envoy-crm.voltade.sg
https://envoy-crm.voltade.sg
https://envoy-crm.voltade.sg
https://envoy-crm.voltade.sg
https://envoy-crm.voltade.sg
https://envoy-crm.voltade.sg
https://envoy-crm.voltade.sg
https://envoy-crm.voltade.sg
https://envoy-crm.voltade.sg
https://envoy-crm.voltade.sg
https://envoy-crm.voltade.sg
https://envoy-crm.voltade.sg/
https://envoy-crm.voltade.sg/
https://envoy-crm.voltade.sg/
https://envoy-crm.voltade.sg/
https://envoy-crm.voltade.sg/
https://envoy-crm.voltade.sg/
https://envoy-crm.voltade.sg/
https://envoy-crm.voltade.sg/
https://envoy-crm.voltade.sg/
https://envoy-crm.voltade.sg/
https://envoy-crm.voltade.sg/
https://envoy-crm.voltade.sg/robots.txt
https://envoy-crm.voltade.sg/robots.txt
https://envoy-crm.voltade.sg/robots.txt
https://envoy-crm.voltade.sg/robots.txt
https://envoy-crm.voltade.sg/robots.txt
https://envoy-crm.voltade.sg/robots.txt
https://envoy-crm.voltade.sg/robots.txt
https://envoy-crm.voltade.sg/robots.txt
https://envoy-crm.voltade.sg/robots.txt
https://envoy-crm.voltade.sg/robots.txt
https://envoy-crm.voltade.sg/robots.txt
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml

https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml
https://envoy-crm.voltade.sg/sitemap.xml

Reference:
https://owasp.org/wstg

SAQ

Sr. No.	Requirements	Applicable	Comments
1	Verify documentation and justification of all the application's trust boundaries, components, and significant data flows.	Yes	All application trust boundaries, components, and significant data flows are documented and reviewed as part of our software development and security assessment processes. Data flow diagrams and architecture documents are kept up to date.
2	Verify the application does not use unsupported, insecure, or deprecated client-side technologies such as NSAPI plugins, Flash, Shockwave, ActiveX, Silverlight, NACL, or client-side Java applets.	Yes	The frontend uses Vue.js, with no deprecated technologies such as Flash, ActiveX, Java applets, Silverlight, or NPAPI plugins. This is enforced via frontend dependency management and code review.
3	Verify that trusted enforcement points, such as access control gateways, servers, and serverless functions, enforce access controls. Never enforce access controls on the client.	Yes	All access control logic is enforced server-side in Rails controllers and policy objects. No access control decisions are made client-side in Vue.js; client logic only determines what to display, never access permission.
4	Verify that all sensitive data is identified and classified into protection levels.	Yes	Sensitive data (e.g., PII, authentication tokens) is identified and tagged in the Rails data model. Data protection requirements are reviewed as part of our development and compliance checklists.
5	Verify that all protection levels have an associated set of protection requirements, such as encryption requirements, integrity requirements, retention, privacy and other confidentiality requirements, and that these are applied in the architecture.	Yes	Each class of sensitive data has defined handling and storage requirements in accordance with our data protection policy. For example, PII fields in Rails models are encrypted and access is restricted by policy.
6	Verify that the application employs integrity protections, such as code signing or subresource integrity. The application must not load or execute code from untrusted sources, such as loading includes, modules, plugins, code, or libraries from untrusted sources or the Internet.	Yes	All third-party JavaScript loaded by the application (e.g., via CDN) uses Subresource Integrity (SRI) tags in HTML templates. Rails' asset pipeline fingerprints static files for cache integrity.
7	Verify that the application has protection from subdomain takeovers if the application relies upon DNS entries or DNS subdomains, such as expired domain names, out of date DNS pointers or CNAMEs, expired projects at public source code repos, or transient cloud APIs, serverless functions, or storage buckets (*autogen-bucket-id*.cloud.example.com) or similar. Protections can include ensuring that DNS names used by applications are regularly checked for expiry or change.	Yes	All DNS records and subdomains (used for APIs, assets, integrations) are reviewed and monitored for expiry or orphaned CNAMEs. Automated checks alert on unused or expired DNS records.
8	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.	Yes	Rate limiting and abuse detection are implemented in Rails controllers for API and sensitive endpoints, using middleware such as rack-attack. HCAPTCHA is optionally enabled on critical forms via Vue.js components.

Sr. No.	Requirements	Applicable	Comments
9	Verify that files obtained from untrusted sources are stored outside the web root, with limited permissions.	Yes	Files uploaded by users are stored using Rails ActiveStorage, which stores uploads outside of the web root (on Cloudflare). Direct links to uploaded files require signed URLs.
10	Verify that files obtained from untrusted sources are scanned by antivirus scanners to prevent upload and serving of known malicious content.	Yes	We have AV protection on Cloudflare Gateway.
11	Verify API URLs do not expose sensitive information, such as the API key, session tokens etc.	Yes	API endpoints and URL paths never include secrets, tokens, or sensitive user information. Authentication tokens are only sent in HTTP headers or secure cookies.
12	Verify that authorization decisions are made at both the URI, enforced by programmatic or declarative security at the controller or router, and at the resource level, enforced by model-based permissions.	Yes	Rails controllers enforce access control via before_action hooks and Pundit policy objects. Resource-level permissions are checked in model methods or policy scopes.
13	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.	Yes	Rails routes are configured to only allow RESTful HTTP methods appropriate for the resource and user role. For example, only admins can access DELETE or PUT for protected resources.
14	Verify that the application build and deployment processes are performed in a secure and repeatable way, such as CI / CD automation, automated configuration management, and automated deployment scripts.	Yes	The application is deployed using automated CI/CD pipelines (GitHub Actions/Forgejo), including environment provisioning and configuration via infrastructure-as-code.
15	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.	Yes	All application components and dependencies can be redeployed using automated scripts, and recovery from backups is tested and documented.
16	Verify that authorized administrators can verify the integrity of all security-relevant configurations to detect tampering.	Yes	Administrative users can verify and audit configuration changes via audit logs in Rails, with configuration management (e.g., via environment variables) tracked in version control and secret managers like 1password.
17	Verify that web or application server and application framework debug modes are disabled in production to eliminate debug features, developer consoles, and unintended security disclosures.	Yes	Rails' config.consider_all_requests_local is set to false and Vue.js production builds have debugging and developer tools disabled.
18	Verify that the supplied Origin header is not used for authentication or access control decisions, as the Origin header can easily be changed by an attacker.	Yes	The application never uses the HTTP Origin header for authentication or access control in Rails controllers or middleware.
19	Verify that user set passwords are at least 12 characters in length	Yes	User-set passwords are required to be at least 12 characters in length, with minimum of 1 uppercase letter, 1 lowercase letter, 1 number, and 1 special character
20	Verify system generated initial passwords or activation codes SHOULD be securely randomly generated, SHOULD be at least 6 characters long, and MAY contain letters and numbers, and expire after a short period of time. These initial secrets must not be permitted to become the long term password.	Yes	Initial passwords and activation codes are generated with SecureRandom.hex or similar, are at least 6 characters, contain both letters and numbers, expire within 24 hours, and cannot be reused as long-term credentials.

Sr. No.	Requirements	Applicable	Comments
21	Verify that passwords are stored in a form that is resistant to offline attacks. Passwords SHALL be salted and hashed using an approved one-way key derivation or password hashing function. Key derivation and password hashing functions take a password, a salt, and a cost factor as inputs when generating a password hash. ([C6](https://owasp.org/www-project-proactive-controls/#div-numbering))	Yes	Passwords are salted and hashed using industry-standard one-way key derivation functions - bcrypt.
22	Verify shared or default accounts are not present (e.g. "root", "admin", or "sa").	Yes	Shared or default accounts (such as â??adminâ?• or â??rootâ?•) are not presently used.
23	Verify that lookup secrets can be used only once.	Yes	Password reset tokens and other lookup secrets are generated by Railsâ?? built-in has_secure_token and can only be used once, expiring immediately upon use.
24	Verify that the out of band verifier expires out of band authentication requests, codes, or tokens after 10 minutes.	Yes	MFA and verification codes are generated with SecureRandom, are valid for 10 minutes or less, and expire automatically in the database after this window.
25	Verify that the initial authentication code is generated by a secure random number generator, containing at least 20 bits of entropy (typically a six digital random number is sufficient).	Yes	Authentication codes are generated with SecureRandom in Rails, providing at least 20 bits of entropy (e.g., 6-digit random codes).
26	Verify that logout and expiration invalidate the session token, such that the back button or a downstream relying party does not resume an authenticated session, including across relying parties. ([C6](https://owasp.org/www-project-proactive-controls/#div-numbering))	Yes	Rails' session management invalidates session tokens on logout or expiration, preventing re-use. Browser back button access is blocked by token expiration.
27	Verify that the application gives the option to terminate all other active sessions after a successful password change (including change via password reset/recovery), and that this is effective across the application, federated login (if present), and any relying parties.	Yes	Upon password change or reset, Rails logic invalidates all other active sessions for the user across all devices.
28	Verify the application uses session tokens rather than static API secrets and keys, except with legacy implementations.	Yes	Session management uses expiring session tokens (Rails cookies or JWTs) rather than static API keys, except for legacy webhook integrations.
29	Verify the application ensures a full, valid login session or requires re-authentication or secondary verification before allowing any sensitive transactions or account modifications.	Yes	Sensitive actions (such as changing email or password) require a valid session and may require the user to re-enter their password or complete a secondary verification step in Vue.js.
30	Verify that the application enforces access control rules on a trusted service layer, especially if client-side access control is present and could be bypassed.	Yes	All access controls are enforced in the Rails backend, and any client-side Vue.js restrictions are supplementary for UX only.
31	Verify that all user and data attributes and policy information used by access controls cannot be manipulated by end users unless specifically authorized.	Yes	All access control attributes are managed server-side in Rails models or policies, and are never exposed for direct manipulation via the Vue.js frontend.
32	Verify that the principle of least privilege exists - users should only be able to access functions, data files, URLs, controllers, services, and other resources, for which they possess specific authorization. This implies protection against spoofing and elevation of privilege. ([C7](https://owasp.org/www-project-proactive-controls/#div-numbering))	Yes	The application enforces least privilege through Rails roles and permissions. Only authorized users can access specific endpoints or resources; privilege escalation is prevented by policy scope.

Sr. No.	Requirements	Applicable	Comments
33	Verify that access controls fail securely including when an exception occurs. ([C10](https://owasp.org/www-project-proactive-controls/#div-numbering))	Yes	Access control logic in Rails returns HTTP 403 or 401 and logs the incident on failure, ensuring secure default behavior even during exceptions.
34	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.	Yes	All Rails resource actions (show, update, delete) require explicit ownership or permission checks in the controller and Pundit policies to prevent IDOR attacks.
35	Verify administrative interfaces use appropriate multi-factor authentication to prevent unauthorized use.	Yes	Administrative interfaces require two-factor authentication, implemented using Deviseâ??s two-factor extension in Rails. Vue.js admin panels enforce re-authentication for sensitive actions.
36	Verify that the application has defenses against HTTP parameter pollution attacks, particularly if the application framework makes no distinction about the source of request parameters (GET, POST, cookies, headers, or environment variables).	Yes	Defenses are in place against HTTP parameter pollution, and the framework distinguishes parameter sources. All input parameters are validated using Zod schemas to reject malformed or unexpected parameters.
37	Verify that the application sanitizes user input before passing to mail systems to protect against SMTP or IMAP injection.	Yes	All user input sent to mail systems is sanitized using validators, stripping CR (\r) and LF (\n) characters to prevent SMTP header injection. The email sending logic only allows whitelisted input fields (name, email, subject, message).
38	Verify that the application avoids the use of eval() or other dynamic code execution features. Where there is no alternative, any user input being included must be sanitized or sandboxed before being executed.	Yes	The application does not use eval() or similar JavaScript dynamic code execution in any component. All template rendering is performed using secure, server-side templating (e.g., EJS with auto-escaping enabled).
39	Verify that the application protects against SSRF attacks, by validating or sanitizing untrusted data or HTTP file metadata, such as filenames and URL input fields, and uses allow lists of protocols, domains, paths and ports.	Yes	All URL inputs (such as webhook endpoints or file import URLs) are validated using allow-lists for protocols (https only) and hostnames. Internal and private IP ranges are explicitly blocked via custom validation middleware.
40	Verify that the application sanitizes, disables, or sandboxes user-supplied Scalable Vector Graphics (SVG) scriptable content, especially as they relate to XSS resulting from inline scripts, and foreignObject.	Yes	SVG uploads are sanitized on the Rails backend using the loofah gem, stripping scripts, event handlers, and foreignObject elements. Vue.js only renders SVGs as sources and never as inline HTML.
41	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 ã•ã•? or O'Hara). ([C4](https://owasp.org/www-project-proactive-controls/#div-numbering))	Yes	Rails view templates use automatic output escaping via ERB by default. Where raw output is required, data is explicitly sanitized. Vue.js bindings auto-escape HTML by default in templates, and all dynamic bindings are context-appropriate.

Sr. No.	Requirements	Applicable	Comments
42	Verify that the application protects against JSON injection attacks, JSON eval attacks, and JavaScript expression evaluation. ([C4](https://owasp.org/www-project-proactive-controls/#div-numbering))	Yes	JSON responses are generated using Railsâ?? render json: helpers, which serialize data securely. No user-controlled input is ever concatenated into executable JavaScript in Vue.js or Rails views.
43	Verify that the application protects against LDAP injection vulnerabilities, or that specific security controls to prevent LDAP injection have been implemented. ([C4](https://owasp.org/www-project-proactive-controls/#div-numbering))	Yes	If LDAP is used, all queries are constructed using parameterized methods (e.g., Net::LDAP in Ruby) with user inputs sanitized to prevent injection of special characters or filters.
44	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.	Yes	Sensitive database columns in Rails use built-in or gem-based encryption (attr_encrypted, ActiveRecord::Encryption). Underlying PostgreSQL storage is encrypted at rest via managed cloud infrastructure.
45	Verify that all cryptographic operations are constant-time, with no 'short-circuit' operations in comparisons, calculations, or returns, to avoid leaking information.	Yes	Password comparisons and token checks in Rails are performed using constant-time methods to prevent timing attacks.
46	Verify that random GUIDs are created using the GUID v4 algorithm, and a Cryptographically-secure Pseudo-random Number Generator (CSPRNG). GUIDs created using other pseudo-random number generators may be predictable.	Yes	Unique identifiers are generated using Rubyâ??s SecureRandom.uuid or JavaScriptâ??s crypto.getRandomValues, both of which use cryptographically secure random number generators.
47	Verify that key material is not exposed to the application but instead uses an isolated security module like a vault for cryptographic operations. ([C8](https://owasp.org/www-project-proactive-controls/#div-numbering))	Yes	Secret keys for encryption are managed via Rails credentials, environment variables, or cloud secret managers. The Rails app accesses keys through secure interfaces and never exposes them in code or logs.
48	Verify that the application does not log credentials or payment details. Session tokens should only be stored in logs in an irreversible, hashed form. ([C9, C10](https://owasp.org/www-project-proactive-controls/#div-numbering))	Yes	Rails logging is configured to filter out sensitive parameters (filter_parameters in application.rb), ensuring passwords, tokens, and payment details are never written to logs. Any tokens logged for debugging are redacted or hashed.
49	Verify the application protects sensitive data from being cached in server components such as load balancers and application caches.	Yes	Controllers serving sensitive or authenticated content set HTTP headers (Cache-Control: no-store, no-cache, must-revalidate) using Railsâ?? before_action callbacks. Cloudflare edge cache is configured to bypass such routes.
50	Verify that data stored in browser storage (such as localStorage, sessionStorage, IndexedDB, or cookies) does not contain sensitive data.	Yes	Sensitive data is never stored in localStorage, sessionStorage, or IndexedDB by the Vue.js frontend. Authentication is handled via secure, HTTP-only cookies set by Rails
51	Verify that sensitive data is sent to the server in the HTTP message body or headers, and that query string parameters from any HTTP verb do not contain sensitive data.	Yes	Sensitive parameters (tokens, passwords, PII) are only submitted in POST request bodies or headers. Rails controllers log and reject requests containing sensitive data in query strings.

Sr. No.	Requirements	Applicable	Comments
52	Verify accessing sensitive data is audited (without logging the sensitive data itself), if the data is collected under relevant data protection directives or where logging of access is required.	Yes	Rails uses ActiveSupport::Notifications and custom logging for access events to sensitive resources, recording user, timestamp, and action without including the sensitive data itself.
53	Verify that connections to and from the server use trusted TLS certificates. Where locally generated or self-signed certificates are used, the server must be configured to only trust specific local CAs and specific self-signed certificates. All others should be rejected.	Yes	All application endpoints use TLS certificates issued by trusted certificate authorities, managed via Cloudflare and upstream infrastructure. There are no self-signed certificates in production.
54	Verify that proper certification revocation, such as Online Certificate Status Protocol (OCSP) Stapling, is enabled and configured.	Yes	TLS is terminated at Cloudflare, which enables and manages OCSP stapling for all certificates, ensuring revocation status is available to clients.

mail@tacsecurity.com I tacsecurity.com