TAC Security
CYBERSECURITY'S FUTURE

Organization: KKCompany Technologies Pte. Ltd., Taiwan Branch

Application: SAST for CXM CMS

ESOF APPSEC ADA CASA - BASIC REPORT

ESOF
Enterprise Security in One Framework
by TAC Security

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

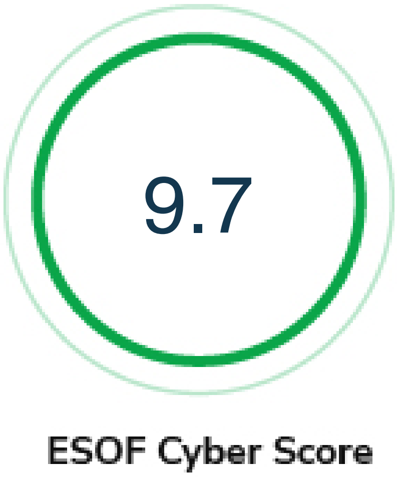
CASA has built upon the industry-recognized standards of the OWASP's Application Security Verification Standard (ASVS) to provide a consistent set of requirements to harden security for any application. Further, CASA provides a uniform way to perform trusted assurance assessments of these requirements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

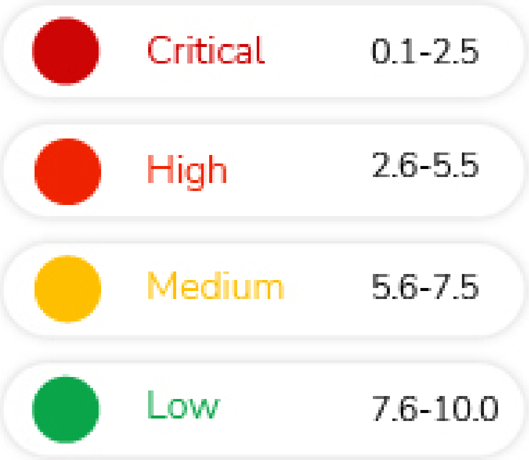
There is no "one size fits all solution" when it comes to evaluating application risk to securing user data. The CASA assessment acknowledges this reality and is adapted with a risk-based, multi-tier assessment approach to evaluate application risk based on user, scope, and other application specific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





●	Critical	0.1-2.5
●	High	2.6-5.5
●	Medium	5.6-7.5
●	Low	7.6-10.0

Target

- **Title:** SAST for CXM CMS
- **Source:**uploaded_scan_file1736323039.zip

Testing Details

Start Date	Jan 8, 2025 07:57:22
Finish Date	Jan 8, 2025 08:30:02

SAQ

Sr. No.	Requirements	Applicable	Comments

Sr. No.	Requirements	Applicable	Comments
1	Verify documentation and justification of all the application's trust boundaries, components, and significant data flows.	Yes	We've documented the trust boundaries, components, and significant data flows.
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 application exclusively uses modern, supported client-side technologies, avoiding deprecated or insecure technologies.
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	Access controls are enforced on the server side through access control mechanism.
4	Verify that all sensitive data is identified and classified into protection levels.	Yes	All sensitive data within the application has been identified and classified into protection levels according to our data classification policy.
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 data protection level has defined protection requirements, including encryption at rest and in transit, integrity checks, and access controls, implemented throughout the application architecture.

Sr. No.	Requirements	Applicable	Comments
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	The application uses code signing and subresource integrity to ensure all code and dependencies are from trusted sources. All external libraries are vetted and obtained from secure repositories.
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	Regular audits are conducted on all DNS entries and subdomains to prevent subdomain takeovers. Automated monitoring tools are in place to alert the team to any changes or expirations in DNS configurations.
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	Anti-automation controls are implemented, including rate limiting to restrict the number of requests per IP, and IP blacklisting to block malicious actors.
9	Verify that files obtained from untrusted sources are stored outside the web root, with limited permissions.	Yes	All files obtained from untrusted sources are securely stored outside the web root directory. These storage locations have limited permissions, ensuring that unauthorized access or execution of these files is prevented.

Sr. No.	Requirements	Applicable	Comments
10	Verify that files obtained from untrusted sources are scanned by antivirus scanners to prevent upload and serving of known malicious content.	Yes	Uploaded files from untrusted sources are automatically scanned using cloud security solution before being processed or served.
11	Verify API URLs do not expose sensitive information, such as the API key, session tokens etc.	Yes	API URLs are carefully designed to exclude sensitive information. Authentication tokens and API keys are transmitted securely in HTTP headers, and no sensitive data is included in the URL parameters.
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	Authorization is enforced at both the URI level and at the resource level through model-based permissions. This dual-layered approach ensures that access control is granular and robust, preventing unauthorized access to sensitive resources.
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	Role-based access controls are implemented, with strict restrictions on RESTful HTTP methods based on user roles.

Sr. No.	Requirements	Applicable	Comments
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 build and deployment processes are fully automated using CI/CD tools. Automated configuration management is handled through Terraform, ensuring consistency and security across all environments.
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	Deployment scripts are fully automated and stored in Repository. A comprehensive, documented runbook outlines the deployment and recovery procedures, which have been tested to ensure that the application, configuration, and all dependencies can be redeployed or restored.
16	Verify that authorized administrators can verify the integrity of all security-relevant configurations to detect tampering.	Yes	Authorized administrators use Git and AWS Config to regularly verify the integrity of all security-relevant configurations. Any unauthorized changes are promptly detected and addressed. Configuration files are version-controlled and monitored for tampering, ensuring ongoing security compliance.

Sr. No.	Requirements	Applicable	Comments
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	Debug modes, developer consoles, and related debug features are disabled in all production environments. Configuration files are reviewed to ensure that no debug settings are active.
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	Authentication and access control decisions are solely based on secure methods. The Origin header is not utilized for any security-sensitive decisions.
19	Verify that cookie-based session tokens utilize the 'SameSite' attribute to limit exposure to cross-site request forgery attacks. ([C6](https://owasp.org/www-project-proactive-controls/#div-numbering))	No	We do not use cookie-based session tokens in our application.
20	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))	No	We do not use LDAP in our application.
21	Verify that the application protects against Local File Inclusion (LFI) or Remote File Inclusion (RFI) attacks.	Yes	The application enforces strict input validation and sanitization on all file path inputs to prevent LFI/RFI attacks.
22	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	All regulated and sensitive data is encrypted at rest using industry-standard encryption algorithms. We securely manage encryption keys through AWS Key Management Service, utilizing AWS managed keys to ensure that data remains protected against unauthorized access.

Sr. No.	Requirements	Applicable	Comments
23	Verify that all cryptographic operations are constant-time, with no 'short-circuit' operations in comparisons, calculations, or returns, to avoid leaking information.	Yes	All cryptographic operations are implemented in a constant-time manner using secure libraries.

TAC Security

CYBERSECURITY'S FUTURE