sawakami.co.th

CDN WAF

Diagnosis Results

Last Update：2025/10/29 17:48

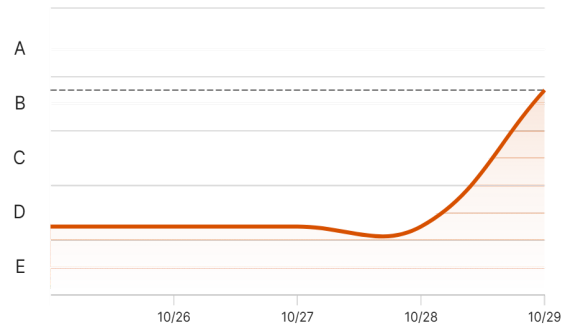
Basic | Detailed | Run Diagnosis

This Site ----- All Registered Sites Average

B

Number of Reports
26

0 1 1 4 22

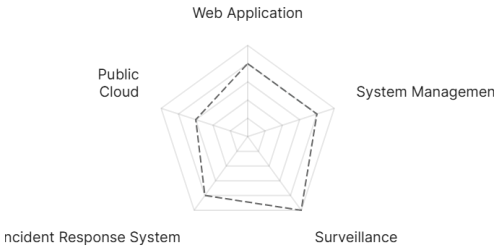


Questionnaire Results

Last Update：

Assessment

This Site ----- All Registered Sites Average



Report Items

	Category	Title

Detailed Scan Report 1

Service — **Exposure of WAF**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The introduction of a Web Application Firewall (WAF) can be detected by external access.
In some cases, specific WAF products (e.g. Cloudflare, AWS WAF, F5 BIG-IP, etc.) can be identified through tools such as Wappalyzer.
WAF detection information can be used by attackers to select evasion strategies and bypass techniques.

<Detected WAF>
- Cloudflare Inc. - Cloudflare

Risk Details

Making WAF visible may allow attackers to understand its behavior and limitations in advance, making it easier for them to try and error to circumvent countermeasures.
If there are known bypass methods or operating characteristics for a specific product, the chances of a targeted attack being successful may increase.

Countermeasures

At this point in time, it cannot be determined that "detecting the existence of a WAF" is a security risk, but the following points are helpful:

- Consider customizing response headers and error messages to make detection more difficult
- Understand the possibility of detection and strengthen rules in WAF settings and establish a log monitoring system
- Regularly perform self-checks with detection tools (e.g. Wappalyzer, Shodan) and strive to understand public information

Basic Scan Report 25

Service — **Security header "X-Frame-Options" is not or improperly enabled**

Consider taking action.

Description

There is an issue with **X-Frame-Options** in the web server response headers.

- X-Frame-Options is not set

Existing websites require the X-Frame-Options header to be set correctly and applied as soon as possible.
X-Frame-Options allows you to control how your site is framed within iframe, frame, embed, and object tags, and is an important security measure to prevent click-hijacking attacks.
Therefore, we recommend setting the X-Frame-Options header to "deny" or "sameorigin" to ensure correct behavior.

<Target URL list>
- https://sawakami.co.th
- https://sawakami.co.th/
- https://sawakami.co.th/about
- https://sawakami.co.th/all-fund
- https://sawakami.co.th/board
- https://sawakami.co.th/career
- https://sawakami.co.th/contact
- https://sawakami.co.th/director
- https://sawakami.co.th/historical
- https://sawakami.co.th/investment
- https://sawakami.co.th/supervise

Show All

Risk Details

Level: Low | Information Disclosure | Missing Access Control

Malicious sites may be vulnerable to clickjacking attacks that trick users into clicking links on the site.
This increases the risk of users unintentionally participating in operations by attackers, which may lead to unauthorized operations and information leaks.

Countermeasures

Appropriate settings in the **X-Frame-Options** header are recommended.

1. Check the web server configuration file and set the X-Frame-Options header
2. Use DENY or SAMEORIGIN options to prevent clickjacking attacks (e.g. "**Header** always set X-Frame-Options "DENY" in Apache, "add_header X-Frame-Options "DENY"; in Nginx)
3. Save settings and restart web server
4. Verify that the X-Frame-Options header is set correctly using a testing tool or browser
5. Review settings regularly to ensure security is maintained

Service — **Security header "Access-Control-Allow-Origin" is enabled**

Consider taking action.

Description

There is a problem with "Access-Control-Allow-**Origin**" in the web server response header.

- "Access-Control-Allow-Origin" is set as "*"
Settings that allow access from all origins increase security risks and should be used with caution. Please narrow down the settings to a specific origin.

To resolve these issues, we recommend that you re-evaluate the need for cross-origin access and strengthen security by setting "Access-Control-Allow-Origin" appropriately.

<Target URL list>
- https://sawakami.co.th/modules/frontend/theme/apexchart/apexcharts.css
- https://sawakami.co.th/modules/frontend/theme/css/demo-gym.css
- https://sawakami.co.th/modules/frontend/theme/css/skin-law-firm-2.css
- https://sawakami.co.th/modules/frontend/theme/css/theme-blog.css

- https://sawakami.co.th/modules/frontend/theme/css/theme-elements.css
- https://sawakami.co.th/modules/frontend/theme/css/theme-shop.css
- https://sawakami.co.th/modules/frontend/theme/css/theme.css
- https://sawakami.co.th/modules/frontend/theme/fontawesome-free/css/all.min.css
- https://sawakami.co.th/modules/frontend/theme/nanoscroller/nanoscroller.css
- https://sawakami.co.th/robots.txt
- https://sawakami.co.th/storage/invoice/1710228832.pdf
- https://sawakami.co.th/storage/websetting/1710393726s.png

Show All

Risk Details

Level: Low | Information Disclosure | Missing Access Control

This may allow resources to be loaded from unintended origins.
If the settings are incorrect, there is a risk that user information of the system operated by the web server being diagnosed may be leaked to a third party site.

Countermeasures

When configuring Access-Control-Allow-**Origin**, we recommend specifying the minimum number of origins required for the appropriate functionality. Also, consider removing this header setting if it is unnecessary.

1. Check the web server and application configuration files and limit the origins allowed in the Access-Control-Allow-Origin header to the minimum necessary.
2. Specify only a specific origin without using wildcards (*) (e.g. Access-Control-Allow-Origin: https://example.com)
3. Save settings and restart server or application
4. Check with a testing tool or browser that the headers are set correctly
5. If unnecessary, remove the Access-Control-Allow-Origin header setting to avoid risks associated with cross-origin resource sharing (CORS)

Service

Improper setting of HTTP header "Set-Cookie"

Consider taking action.

Description

"**Set-Cookie**" is not set properly in the web server's response header.

- HttpOnly attribute
If the HttpOnly attribute is not set, the cookie will be accessible to client-side scripts and there is a risk that it can be used for cross-site scripting (XSS) attacks. Enable the HttpOnly attribute to prevent access from client-side scripts.

We recommend that you configure the "Set-**Cookie**" settings appropriately to enhance cookie security.

<Target URL list>
- URL: https://sawakami.co.th Cookie: XSRF-TOKEN
- URL: https://sawakami.co.th/ Cookie: XSRF-TOKEN
- URL: https://sawakami.co.th/about Cookie: XSRF-TOKEN
- URL: https://sawakami.co.th/all-fund Cookie: XSRF-TOKEN
- URL: https://sawakami.co.th/board Cookie: XSRF-TOKEN
- URL: https://sawakami.co.th/career Cookie: XSRF-TOKEN
- URL: https://sawakami.co.th/contact Cookie: XSRF-TOKEN
- URL: https://sawakami.co.th/director Cookie: XSRF-TOKEN
- URL: https://sawakami.co.th/historical Cookie: XSRF-TOKEN
- URL: https://sawakami.co.th/investment Cookie: XSRF-TOKEN
- URL: https://sawakami.co.th/supervise Cookie: XSRF-TOKEN

Show All

Risk Details

Level: Low | Information Disclosure | Authentication Weakness

If the HttpOnly attribute is not set, JavaScript can gain access to cookies, increasing the risk of session hijacking by an attacker.

Countermeasures

We recommend appropriate settings for **Set-Cookie**.

1. Check the web server and application configuration files and add security-enhancing options to the Set-**Cookie** header.
2. Use the HttpOnly option to prevent cookies from being accessed by JavaScript
3. Use the Secure option to ensure cookies are only sent over **HTTPS** communication
4. Configure Strict or Lax to prevent cross-site request forgery (CSRF) using the SameSite option
5. Save settings and restart server or application
6. Check with your browser or testing tool that cookie settings are applied correctly
7. Review settings regularly and update them according to security requirements

Service **Security header "Content-Security-Policy" is not or improperly enabled**

Consider taking action.

Description

There is an issue with "Content-Security-Policy" in the web server response header.

- "Content-Security-Policy" is not set
Properly setting this header limits the types of content that the browser allows to load and execute, preventing client-side attacks such as cross-site scripting (XSS) and click hijacking.
We recommend implementing Content-Security-Policy to restrict loading of specific sources and scripts.

<Target URL list>
- https://sawakami.co.th
- https://sawakami.co.th/
- https://sawakami.co.th/about
- https://sawakami.co.th/all-fund
- https://sawakami.co.th/board
- https://sawakami.co.th/career
- https://sawakami.co.th/contact
- https://sawakami.co.th/director
- https://sawakami.co.th/investment
- https://sawakami.co.th/sitemap.xml
- https://sawakami.co.th/supervise

This allows the browser to safely handle content and prevent client-side attacks.

Show All

Risk Details

Level: Low Information Disclosure Insufficient Input Validation

May be vulnerable to cross-site scripting (XSS), clickjacking, and other code injection attacks.
This increases the risk that an attacker can run malicious scripts or trick users into performing unauthorized actions.

Countermeasures

Proper configuration of Content-Security-Policy (CSP) is recommended.

1. Check your web server and application configuration files, add CSP headers, and set a policy to only allow trusted resources.
2. Allow only specific resources whenever possible (e.g. default-src 'self'; script-src 'self' https://trusted-scripts.example.com; style-src 'self' https://trusted-styles.example.com)
3. Avoid using unsafe-inline and unsafe-eval to minimize security risks
4. Save settings and restart web server and application
5. Verify that CSP policies are applied correctly using browser developer and testing tools
6. Regularly review settings and update policies according to security requirements

Service **Detecting version information (x-powered-by)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

You can check version information externally.
Detailed version information can be useful information for attackers and can be used as clues for attacks.
Therefore, consider concealing version information.

<Target URL list>
- https://sawakami.co.th
- https://sawakami.co.th/
- https://sawakami.co.th/about
- https://sawakami.co.th/board
- https://sawakami.co.th/career
- https://sawakami.co.th/contact
- https://sawakami.co.th/director
- https://sawakami.co.th/investment
- https://sawakami.co.th/sitemap.xml

<Target version information list>
- PHP/8.3.27

Show All

Risk Details

Information Exposure to Attackers

There is a risk of attacks targeting known vulnerabilities based on version information.
Additionally, security measures may be deemed insufficient, inviting further investigation and attacks.

Countermeasures

We recommend hiding version information.

1. Check the web server configuration file (Apache, Nginx, etc.)
2. Add setting to not display version information
3. Save settings and restart web server
4. Check if version information is hidden using a browser or testing tool

Service **Subdomain detected**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

A list of subdomains related to the domain to be diagnosed.
If subdomain information is known to an attacker, there is a risk that it will become the target of attacks or investigations targeting vulnerabilities in each subdomain.
Therefore, we recommend that you tighten access restrictions to subdomains and avoid exposing unnecessary subdomains.

<Target subdomain list>
- go.sawakami.co.th
- lp-eopen.sawakami.co.th
- reqreset.sawakami.co.th
- streamingfundplus.sawakami.co.th
- test.sawakami.co.th
- www.sawakami.co.th
- www2.sawakami.co.th

Show All

Risk Details

Information Disclosure Information Exposure to Attackers

This may provide useful information to an attacker. If a domain exists that you don't know exists or that you don't intend to make public but is accessible, it can be a source of attack.
In addition, even if a domain is registered, if there is no server to resolve the name, there is a risk that it will become the target of a subdomain takeover attack.

Countermeasures

If an unintended external public domain exists, we recommend appropriate access control to that domain.

1. Check publicly available domains and identify domains with unintended external exposure
2. Control access using firewall and DNS settings so that only authorized IP addresses and ranges can access.
3. Save your settings and ensure access control is working properly
4. Regularly check domain access logs and monitor for unauthorized access or abnormal connections.

Service

Exposure of middleware framework name by HTTP header: x-powered-by

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The middleware framework name is disclosed in "X-Powered-By" included in the web server response header.
If this information is exposed externally, it becomes useful information for attackers, increasing the risk of attacks targeting specific vulnerabilities.
Therefore, we recommend removing the "X-Powered-By" information from the response header to strengthen security.

<List of target middleware/frameworks>
- PHP
- PleskLin

<Target URL list>
- https://sawakami.co.th
- https://sawakami.co.th/
- https://sawakami.co.th/about
- https://sawakami.co.th/board
- https://sawakami.co.th/career
- https://sawakami.co.th/contact
- https://sawakami.co.th/director
- https://sawakami.co.th/en/
- https://sawakami.co.th/investment
- https://sawakami.co.th/jp/
- https://sawakami.co.th/sitemap.xml

Show All

Risk Details

Information Exposure to Attackers

Disclosing the software name can be useful information for attackers and may provide clues for attacks that exploit known vulnerabilities.
This increases the risk that your system will become a target for attacks targeting vulnerabilities.

Countermeasures

We recommend that you remove the X-Powered-By header if it is not needed, as there is a risk of leaking the technology stack and version information in use.

- For PHP, configure the following settings in php.ini to remove the X-Powered-By header.
- For Apache, set **Header** unset X-Powered-By to remove the X-Powered-By header.
- For Nginx, set fastcgi_param to remove the X-Powered-By header

-Save the settings, restart the web server, and verify that the X-Powered-By header is removed.

Review your settings regularly to ensure that unintended information is not disclosed to external parties.

Service

Exposure of Web-related technology stack (ApexCharts.js)

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- ApexCharts.js

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (Bootstrap)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- Bootstrap

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (core-js/3.27.1)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- core-js (3.27.1)

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (D3)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- D3

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (GSAP/3.9.1)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- GSAP (3.9.1)

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (jQuery/3.6.0)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- jQuery (3.6.0)

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs

- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (jQuery Sparklines)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- jQuery Sparklines

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (jQuery UI/1.10.3)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- jQuery UI (1.10.3)

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (Laravel)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- Laravel

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (Lightbox)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- Lightbox

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (Livewire)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- Livewire

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (Modernizr)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- Modernizr

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (OWL Carousel)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- OWL Carousel

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (PhotoSwipe)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- PhotoSwipe

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (PHP/8.3.27)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- PHP (8.3.27)

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (Splide)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- Splide

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (SweetAlert2)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- SweetAlert2

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used

Service **Exposure of Web-related technology stack (xCharts)**

If issues are occurring in unintended situations, please verify the diagnosis target.

Description

The technologies used by a web application (CMS, JavaScript framework, server software, etc.) can be easily identified from the outside.
If this information is publicly available, attackers are more likely to launch attacks targeting known vulnerabilities linked to specific technologies. In particular, the risk is more pronounced when version information and framework type can be identified.

<Detected technology stack>
- xCharts

<Target URL list>
- https://sawakami.co.th/

Risk Details

By externally identifying the technologies and version information used, it becomes easier for attackers to select and execute attack methods that correspond to those technologies. As a result, there is concern that the risk of intrusion through targeted attacks and automated scanning will increase.

Countermeasures

To minimize the exposure of your technology stack to the outside world, we recommend the following measures:

- Removal and masking of unnecessary response headers (X-Powered-By, Server, etc.)
- Version management of JavaScript libraries and review of cache and reference settings when using CDNs
- Review of web server and CMS configurations (restrictions on directory indexes and public APIs, etc.)
- Regular vulnerability assessments and inventory of technologies used