Acunetix
by Invicti

# Comprehensive Report

## Acunetix Threat Level 1

One or more low-severity type vulnerabilities have been discovered by the scanner.

## Scan Detail

| | |
|---|---|
| Target | https://api.capitansissy.cw:444/ |
| Scan Type | Full Scan |
| Start Time | Dec 31, 2021, 10:13:28 PM GMT+3 |
| Scan Duration | 16 minutes |
| Requests | 50841 |
| Average Response Time | 1ms |
| Maximum Response Time | 17628ms |

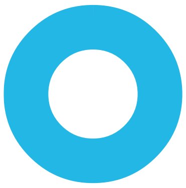
| 0 | 0 | 1 | 3 |
|---|---|---|---|
| High | Medium | Low | Informational |

| Severity | | Vulnerabilities | Instances |
|---|---|---|---|
| ❗ | High | 0 | 0 |
| ❗ | Medium | 0 | 0 |
| ❕ | Low | 1 | 1 |
| ⓘ | Informational | 3 | 3 |
| | Total | 4 | 4 |

## Informational

| | | Instances |
|---|---|---|
| 🟩 | Content Security Policy (CSP) not implemented | 1 |
| 🟩 | Content type is not specified | 1 |
| 🟩 | Subresource Integrity (SRI) not implemented | 1 |

## Low Severity



| | | Instances |
|---|---|---|
| 🟦 | Composer installed.json publicly accessible | 1 |

## Impacts

| SEVERITY | | IMPACT |
|---|---|---|
| Low | 1 | Composer installed.json publicly accessible |
| Informational | 1 | Content Security Policy (CSP) not implemented |
| Informational | 1 | Content type is not specified |
| Informational | 1 | Subresource Integrity (SRI) not implemented |

## Composer installed.json publicly accessible

A **installed.json** file was discovered. Composer is a tool for dependency management in PHP. It allows you to declare the libraries your project depends on and it will manage (install/update) them for you. After installing the dependencies, Composer stores the list of them in a special file for internal purposes.

As the file is publicly accessible, it leads to disclosure of information about components used by the web application.

### Impact

installed.json discloses sensitive information. This information can be used to launch further attacks.

### https://api.capitansissy.cw:444/phpmyadmin/

#### Request

```
GET /phpmyadmin/vendor/composer/installed.json HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/92.0.4512.0 Safari/537.36
Host: api.capitansissy.cw:444
Connection: Keep-alive
```

### Recommendation

Restrict access to vendors directory

### References

Composer Basic usage
https://getcomposer.org/doc/01-basic-usage.md

## Content Security Policy (CSP) not implemented

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks.

Content Security Policy (CSP) can be implemented by adding a **Content-Security-Policy** header. The value of this header is a string containing the policy directives describing your Content Security Policy. To implement CSP, you should define lists of allowed origins for the all of the types of resources that your site utilizes. For example, if you have a simple site that needs to load scripts, stylesheets, and images hosted locally, as well as from the jQuery library from their CDN, the CSP header could look like the following:

```
Content-Security-Policy:
default-src 'self';
script-src 'self' https://code.jquery.com;
```

It was detected that your web application doesn't implement Content Security Policy (CSP) as the CSP header is missing from the response. It's recommended to implement Content Security Policy (CSP) into your web application.

## Impact

CSP can be used to prevent and/or mitigate attacks that involve content/code injection, such as cross-site scripting/XSS attacks, attacks that require embedding a malicious resource, attacks that involve malicious use of iframes, such as clickjacking attacks, and others.

---

## https://api.capitansissy.cw:444/

Paths without CSP header:

- https://api.capitansissy.cw:444/assets/fonts/eot/IRANSansWeb(FaNum
- https://api.capitansissy.cw:444/assets/fonts/ttf/IRANSansWeb(FaNum
- https://api.capitansissy.cw:444/assets/fonts/woff/IRANSansWeb(FaNum
- https://api.capitansissy.cw:444/assets/fonts/woff2/IRANSansWeb(FaNum

### Request

```
GET /assets/fonts/eot/IRANSansWeb(FaNum HTTP/1.1
Referer: https://api.capitansissy.cw:444/assets/css/iransans-serif.css
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/92.0.4512.0 Safari/537.36
Host: api.capitansissy.cw:444
Connection: Keep-alive
```

## Recommendation

It's recommended to implement Content Security Policy (CSP) into your web application. Configuring Content Security Policy involves adding the **Content-Security-Policy** HTTP header to a web page and giving it values to control resources the user agent is allowed to load for that page.

### References

Content Security Policy (CSP)
https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP

Implementing Content Security Policy
https://hacks.mozilla.org/2016/02/implementing-content-security-policy/

## Content type is not specified

These page(s) does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.

### Impact

None

---

### https://api.capitansissy.cw:444/ Verified

Pages where the content-type header is not specified:

- https://api.capitansissy.cw:444/assets/favicon/site.webmanifest

#### Request

```
GET /assets/favicon/site.webmanifest HTTP/1.1
Referer: https://api.capitansissy.cw:444/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/92.0.4512.0 Safari/537.36
Host: api.capitansissy.cw:444
Connection: Keep-alive
```

### Recommendation

Set a Content-Type header value for these page(s).

# Subresource Integrity (SRI) not implemented

Subresource Integrity (SRI) is a security feature that enables browsers to verify that third-party resources they fetch (for example, from a CDN) are delivered without unexpected manipulation. It works by allowing developers to provide a cryptographic hash that a fetched file must match.

Third-party resources (such as scripts and stylesheets) can be manipulated. An attacker that has access or has hacked the hosting CDN can manipulate or replace the files. SRI allows developers to specify a base64-encoded cryptographic hash of the resource to be loaded. The integrity attribute containing the hash is then added to the <script> HTML element tag. The integrity string consists of a base64-encoded hash, followed by a prefix that depends on the hash algorithm. This prefix can either be sha256, sha384 or sha512.

The script loaded from the external URL specified in the Details section doesn't implement Subresource Integrity (SRI). It's recommended to implement Subresource Integrity (SRI) for all the scripts loaded from external hosts.

## Impact

An attacker that has access or has hacked the hosting CDN can manipulate or replace the files.

---

## https://api.capitansissy.cw:444/

Pages where SRI is not implemented:

- https://api.capitansissy.cw:444/
  Script SRC: **https://kit.fontawesome.com/7524e05e75.js**

### Request

```
GET / HTTP/1.1
Referer: https://api.capitansissy.cw:444/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/92.0.4512.0 Safari/537.36
Host: api.capitansissy.cw:444
Connection: Keep-alive
```

## Recommendation

Use the SRI Hash Generator link (from the References section) to generate a <script> element that implements Subresource Integrity (SRI).

For example, you can use the following <script> element to tell a browser that before executing the https://example.com/example-framework.js script, the browser must first compare the script to the expected hash, and verify that there's a match.

```
<script src="https://example.com/example-framework.js"
integrity="sha384-oqVuAfXRKap7fdgcCY5uykM6+R9GqQ8K/uxy9rx7HNQlGYl1kPzQho1wx4JwY8wC"
crossorigin="anonymous"></script>
```

### References

Subresource Integrity
https://developer.mozilla.org/en-US/docs/Web/Security/Subresource_Integrity

SRI Hash Generator
https://www.srihash.org/

## Coverage

- https://api.capitansissy.cw:444
  - assets
    - css
      - iransans-serif.css
      - style.css
    - favicon
      - site.webmanifest
    - fonts
      - eot
        - IRANSansWeb(FaNum
      - ttf
        - IRANSansWeb(FaNum
      - woff
        - IRANSansWeb(FaNum
      - woff2
        - IRANSansWeb(FaNum
    - js
      - soap-action.js
  - languages
  - phpmyadmin
    - doc
      - html
    - examples
    - js
      - src
        - database
        - setup
      - vendor
    - setup
      - lib
    - sql

  - templates
    - config
    - console
    - database
      - export
      - import
      - search
    - error
    - export
    - import
    - login
    - menu
    - plugins
    - setup
      - config
    - sql
    - test
  - themes
  - TMP
  - vendor
    - composer
      - installed.json
    - phpmyadmin
- functions.php
- header.php
- index.asp
- index.php