netsparker

6/3/2023 6:33:01 PM (UTC+05:30)

Detailed Scan Report

http://127.0.0.1:8000/docs#/default/predict_predict__post

Scan Time	: 6/3/2023 6:30:31 PM (UTC+05:30)
Scan Duration	: 00:00:00:38
Total Requests	: 339
Average Speed	: 8.7r/s

Risk Level:
MEDIUM

7 IDENTIFIED	**0** CONFIRMED	**0** CRITICAL
0 HIGH	1 MEDIUM	2 LOW
	3 BEST PRACTICE	1 INFORMATION

Identified Vulnerabilities

Critical	0
High	0
Medium	1
Low	2
Best Practice	3
Information	1
TOTAL	**7**

Confirmed Vulnerabilities

Critical	0
High	0
Medium	0
Low	0
Best Practice	0
Information	0
TOTAL	**0**

Vulnerability Summary

CONFIRM	VULNERABILITY	METHOD	URL	PARAMETER
	SSL/TLS Not Implemented	GET	https://127.0.0.1/docs#/default/predict_predict__post	
	Misconfigured Access-Control-Allow-Origin Header	GET	http://127.0.0.1:8000/	URI-BASED
	Missing X-Frame-Options Header	GET	http://127.0.0.1:8000/docs#/default/predict_predict__post	
	Content Security Policy (CSP) Not Implemented	GET	http://127.0.0.1:8000/docs#/default/predict_predict__post	
	Missing X-XSS-Protection Header	GET	http://127.0.0.1:8000/docs#/default/predict_predict__post	
	Subresource Integrity (SRI) Not Implemented	GET	http://127.0.0.1:8000/docs#/default/predict_predict__post	
	Expect-CT Security Header Errors and Warnings	GET	http://127.0.0.1:8000/docs#/default/predict_predict__post	

1. SSL/TLS Not Implemented

MEDIUM | 1

Netsparker detected that SSL/TLS is not implemented.

Impact

An attacker who is able to intercept your - or your users' - network traffic can read and modify any messages that are exchanged with your server.

That means that an attacker can see passwords in clear text, modify the appearance of your website, redirect the user to other web pages or steal session information.

Therefore no message you send to the server remains confidential.

Vulnerabilities

1.1. https://127.0.0.1/docs#/default/predict_predict__post

Certainty

Request

```
[NETSPARKER] SSL Connection
```

Response

Response Time (ms) : 1 **Total Bytes Received** : 27 **Body Length** : 0 **Is Compressed** : No

```
[NETSPARKER] SSL Connection
```

Remedy

We suggest that you implement SSL/TLS properly, for example by using the Certbot toolprovided by the Let's Encrypt certificate authority. It can automatically configure most modern web servers, e.g. Apache and Nginx to use SSL/TLS. Both the tool and the certificates are free and are usually installed within minutes.

CLASSIFICATION

PCI DSS v3.2	6.5.4
OWASP 2013	A6
OWASP 2017	A3
SANS Top 25	311
CAPEC	217
WASC	4
HIPAA	164.306
ISO27001	A.14.1.3

CVSS 3.0 SCORE

Base	6.8 (Medium)
Temporal	6.1 (Medium)
Environmental	6.1 (Medium)

CVSS Vector String

CVSS:3.0/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N/E:P/RL:O/RC:C

CVSS 3.1 SCORE

Base	6.8 (Medium)
Temporal	6.1 (Medium)
Environmental	6.1 (Medium)

CVSS Vector String
CVSS:3.1/AV:A/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N/E:P/RL:O/RC:C

2. Misconfigured Access-Control-Allow-Origin Header

LOW | 1

Netsparker detected a possibly misconfigured Access-Control-Allow-Origin header in resource's HTTP response.

Cross-origin resource sharing (CORS) is a mechanism that allows resources on a web page to be requested outside the domain through XMLHttpRequest.

Unless this HTTP header is present, such "cross-domain" requests are forbidden by web browsers, per the same-origin security policy.

Impact

This is generally not appropriate when using the same-origin security policy. The only case where this is appropriate when using the same-origin policy is when a page or API response is considered completely public content and it is intended to be accessible to everyone.

Vulnerabilities

2.1. http://127.0.0.1:8000/

Method	Parameter	Value
GET	URI-BASED	

Access-Control-Allow-Origin

- http://r87.com

Certainty

Request

```
GET / HTTP/1.1
Host: 127.0.0.1:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: foo=bar
Origin: http://r87.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 2.4118 **Total Bytes Received** : 213 **Body Length** : 22 **Is Compressed** : No

```
HTTP/1.1 404 Not Found
server: uvicorn
content-length: 22
access-control-allow-origin: http://r87.com
content-type: application/json
date: Sat, 03 Jun 2023 13:00:57 GMT
vary: Origin

{"detail":"Not Found"}
```

Remedy

If this page is intended to be accessible to everyone, you don't need to take any action. Otherwise please follow the guidelines for different architectures below in order to set this header and permit outside domain.
Apache

- Add the following line inside either the <directory>, <location>, <files> or <virtualhost> sections of your server config (usually located in `httpd.conf`or `apache.conf`), or within a `.htaccess`file.

  ```
  Header set Access-Control-Allow-Origin "domain"
  ```

IIS6

1. Open Internet Information Service (IIS) Manager
2. Right click the site you want to enable CORS for and go to Properties
3. Change to the HTTP Headers tab

4. In the Custom HTTP headers section, click Add
5. Enter Access-Control-Allow-Origin as the header name
6. Enter domainas the header value

IIS7

- Merge the following xml into the web.config file at the root of your application or site:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.webserver>
    <httpprotocol>
      <customheaders>
        <add name="Access-Control-Allow-Origin" value="domain" />
      </customheaders>
    </httpprotocol>
  </system.webserver>
</configuration>
```

ASP.NET

- If you don't have access to configure IIS, you can still add the header through ASP.NET by adding the following line to your source pages:

```
Response.AppendHeader("Access-Control-Allow-Origin", "domain");
```

External References

- Cross-Origin Resource Sharing
- HTTP access control (CORS)
- Using CORS

CLASSIFICATION

OWASP 2013	A5
OWASP 2017	A6
SANS Top 25	16
WASC	15
ISO27001	A.14.1.2

3. Missing X-Frame-Options Header

LOW | 1

Netsparker detected a missing `X-Frame-Options`header which means that this website could be at risk of a clickjacking attack.

The `X-Frame-Options`HTTP header field indicates a policy that specifies whether the browser should render the transmitted resource within a `frame`or an `iframe`. Servers can declare this policy in the header of their HTTP responses to prevent clickjacking attacks, which ensures that their content is not embedded into other pages or frames.

Impact

Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on a framed page when they were intending to click on the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both.

Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

Vulnerabilities

3.1. http://127.0.0.1:8000/docs#/default/predict_predict__post

Certainty

Request

```
GET /docs#/default/predict_predict__post HTTP/1.1
Host: 127.0.0.1:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 107.1777 **Total Bytes Received** : 1065 **Body Length** : 931 **Is Compressed** : No

```
HTTP/1.1 200 OK
server: uvicorn
content-length: 931
content-type: text/html; charset=utf-8
date: Sat, 03 Jun 2023 13:00:38 GMT


<!DOCTYPE html>
<html>
<head>
<link type="text/css" rel="stylesheet" href="https://cdn.jsdelivr.net/npm/swagger-ui-dist@4/swagger-ui.
css">
<link rel="shortcut icon" href="https://fastapi.tiangolo.com/img/favicon.png">
<title>FastAPI - Swagger UI</title>
</head>
<body>
<div id="swagger-ui">
</div>
<script src="https://cdn.jsdelivr.net/npm/swagger-ui-dist@4/swagger-ui-bundle.js"></script>
<!-- `SwaggerUIBundle` is now available on the page -->
<script>
const ui = SwaggerUIBundle({
url: '/openapi.json',
"dom_id": "#swagger-ui",
"layout": "BaseLayout",
"deepLinking": true,
"showExtensions": true,
"showCommonExtensions": true,
oauth2RedirectUrl: window.location.origin + '/docs/oauth2-redirect',
presets: [
SwaggerUIBundle.presets.apis,
SwaggerUIBundle.SwaggerUIStandalonePreset
],
})
</script>
</body>
</html>
```

Remedy

- Sending the proper X-Frame-Options in HTTP response headers that instruct the browser to not allow framing from other domains.
    - `X-Frame-Options: DENY`It completely denies to be loaded in frame/iframe.
    - `X-Frame-Options: SAMEORIGIN`It allows only if the site which wants to load has a same origin.

    - `X-Frame-Options: ALLOW-FROM` *`URL`* It grants a specific URL to load itself in a iframe. However please pay attention to that, not all browsers support this.
- Employing defensive code in the UI to ensure that the current frame is the most top level window.

External References

- Clickjacking
- Can I Use X-Frame-Options
- X-Frame-Options HTTP Header

Remedy References

- Clickjacking Defense Cheat Sheet

CLASSIFICATION

OWASP 2013	A5
OWASP 2017	A6
SANS Top 25	693
CAPEC	103
ISO27001	A.14.2.5

4. Content Security Policy (CSP) Not Implemented

BEST PRACTICE | 1

CSP is an added layer of security that helps to mitigate mainly Cross-site Scripting attacks.

CSP can be enabled instructing the browser with a Content-Security-Policy directive in a response header;

```
Content-Security-Policy: script-src 'self';
```

or in a meta tag;

```
<meta http-equiv="Content-Security-Policy" content="script-src 'self';">
```

In the above example, you can restrict script loading only to the same domain. It will also restrict inline script executions both in the element attributes and the event handlers. There are various directives which you can use by declaring CSP:

- **script-src:**Restricts the script loading resources to the ones you declared. By default, it disables inline script executions unless you permit to the evaluation functions and inline scripts by the unsafe-eval and unsafe-inline keywords.
- **base-uri:**Base element is used to resolve relative URL to absolute one. By using this CSP directive, you can define all possible URLs which could be assigned to base-href attribute of the document.
- **frame-ancestors**: It is very similar to X-Frame-Options HTTP header. It defines the URLs by which the page can be loaded in an iframe.
- **frame-src / child-src**: frame-src is the deprecated version of child-src. Both define the sources that can be loaded by iframe in the page. (Please note that frame-src was brought back in CSP 3)
- **object-src**: Defines the resources that can be loaded by embedding such as Flash files, Java Applets.
- **img-src**: As its name implies, it defines the resources where the images can be loaded from.
- **connect-src**: Defines the whitelisted targets for XMLHttpRequest and WebSocket objects.
- **default-src**: It is a fallback for the directives that mostly ends with -src suffix. When the directives below are not defined, the value set to default-src will be used instead:
    - child-src
    - connect-src
    - font-src
    - img-src
    - manifest-src
    - media-src
    - object-src
    - script-src
    - style-src

When setting the CSP directives, you can also use some CSP keywords:

- **none**: Denies loading resources from anywhere.
- **self** : Points to the document's URL (domain + port).
- **unsafe-inline**: Permits running inline scripts.
- **unsafe-eval**: Permits execution of evaluation functions such as eval().

In addition to CSP keywords, you can also use wildcard or only a scheme when defining whitelist URLs for the points. Wildcard can be used for subdomain and port portions of the URLs:

```
Content-Security-Policy: script-src https://*.example.com;
Content-Security-Policy: script-src https://example.com:*;
Content-Security-Policy: script-src https:;
```

It is also possible to set a CSP in Report-Only mode instead of forcing it immediately in the migration period. Thus you can see the violations of the CSP policy in the current state of your web site while migrating to CSP:

```
Content-Security-Policy-Report-Only: script-src 'self'; report-uri: https://example.com;
```

Impact

There is no direct impact of not implementing CSP on your website. However, if your website is vulnerable to a Cross-site Scripting attack CSP can prevent successful exploitation of that vulnerability. By not implementing CSP you'll be missing out this extra layer of security.

Vulnerabilities

4.1. http://127.0.0.1:8000/docs#/default/predict_predict__post

Certainty

Request

```
GET /docs#/default/predict_predict__post HTTP/1.1
Host: 127.0.0.1:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 107.1777 **Total Bytes Received** : 1065 **Body Length** : 931 **Is Compressed** : No

```
HTTP/1.1 200 OK
server: uvicorn
content-length: 931
content-type: text/html; charset=utf-8
date: Sat, 03 Jun 2023 13:00:38 GMT


<!DOCTYPE html>
<html>
<head>
<link type="text/css" rel="stylesheet" href="https://cdn.jsdelivr.net/npm/swagger-ui-dist@4/swagger-ui.
css">
<link rel="shortcut icon" href="https://fastapi.tiangolo.com/img/favicon.png">
<title>FastAPI - Swagger UI</title>
</head>
<body>
<div id="swagger-ui">
</div>
<script src="https://cdn.jsdelivr.net/npm/swagger-ui-dist@4/swagger-ui-bundle.js"></script>
<!-- `SwaggerUIBundle` is now available on the page -->
<script>
const ui = SwaggerUIBundle({
url: '/openapi.json',
"dom_id": "#swagger-ui",
"layout": "BaseLayout",
"deepLinking": true,
"showExtensions": true,
"showCommonExtensions": true,
oauth2RedirectUrl: window.location.origin + '/docs/oauth2-redirect',
presets: [
SwaggerUIBundle.presets.apis,
SwaggerUIBundle.SwaggerUIStandalonePreset
],
})
</script>
</body>
</html>
```

Actions to Take

- Enable CSP on your website by sending the `Content-Security-Policy`in HTTP response headers that instruct the browser to apply the policies you specified.
- Apply the whitelist and policies as strict as possible.
- Rescan your application to see if Netsparker identifies any weaknesses in your policies.

Remedy

Enable CSP on your website by sending the `Content-Security-Policy`in HTTP response headers that instruct the browser to apply the policies you specified.

External References

- An Introduction to Content Security Policy
- Content Security Policy (CSP) HTTP Header
- Content Security Policy (CSP)

CLASSIFICATION

SANS Top 25	16
WASC	15
ISO27001	A.14.2.5

5. Missing X-XSS-Protection Header

BEST PRACTICE | 1

Netsparker detected a missing `X-XSS-Protection`header which means that this website could be at risk of a Cross-site Scripting (XSS) attacks.

Impact

This issue is reported as additional information only. There is no direct impact arising from this issue.

Vulnerabilities

5.1. http://127.0.0.1:8000/docs#/default/predict_predict__post

Certainty

Request

```
GET /docs#/default/predict_predict__post HTTP/1.1
Host: 127.0.0.1:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 107.1777 **Total Bytes Received** : 1065 **Body Length** : 931 **Is Compressed** : No

```
HTTP/1.1 200 OK
server: uvicorn
content-length: 931
content-type: text/html; charset=utf-8
date: Sat, 03 Jun 2023 13:00:38 GMT


<!DOCTYPE html>
<html>
<head>
<link type="text/css" rel="stylesheet" href="https://cdn.jsdelivr.net/npm/swagger-ui-dist@4/swagger-ui.
css">
<link rel="shortcut icon" href="https://fastapi.tiangolo.com/img/favicon.png">
<title>FastAPI - Swagger UI</title>
</head>
<body>
<div id="swagger-ui">
</div>
<script src="https://cdn.jsdelivr.net/npm/swagger-ui-dist@4/swagger-ui-bundle.js"></script>
<!-- `SwaggerUIBundle` is now available on the page -->
<script>
const ui = SwaggerUIBundle({
url: '/openapi.json',
"dom_id": "#swagger-ui",
"layout": "BaseLayout",
"deepLinking": true,
"showExtensions": true,
"showCommonExtensions": true,
oauth2RedirectUrl: window.location.origin + '/docs/oauth2-redirect',
presets: [
SwaggerUIBundle.presets.apis,
SwaggerUIBundle.SwaggerUIStandalonePreset
],
})
</script>
</body>
</html>
```

Remedy

Add the X-XSS-Protection header with a value of "1; mode= block".

- `X-XSS-Protection: 1; mode=block`

External References

- Internet Explorer 8 Security Features - MSDN
- X-XSS-Protection HTTP Header
- Internet Explorer 8 XSS Filter

CLASSIFICATION

SANS Top 25	16
WASC	15
HIPAA	164.308(A)
ISO27001	A.14.2.5

6. Subresource Integrity (SRI) Not Implemented

BEST PRACTICE | 1

Subresource Integrity (SRI) provides a mechanism to check integrity of the resource hosted by third parties like Content Delivery Networks (CDNs) and verifies that the fetched resource has been delivered without unexpected manipulation.

SRI does this using hash comparison mechanism. In this way, hash value declared in HTML elements (for now only script and link elements are supported) will be compared with the hash value of the resource hosted by third party.

Use of SRI is recommended as a best-practice, whenever libraries are loaded from a third-party source.

Vulnerabilities

6.1. http://127.0.0.1:8000/docs#/default/predict_predict__post

Identified Sub Resource(s)

- https://cdn.jsdelivr.net/npm/swagger-ui-dist@4/swagger-ui.css
- https://cdn.jsdelivr.net/npm/swagger-ui-dist@4/swagger-ui-bundle.js

Certainty

Request

```
GET /docs#/default/predict_predict__post HTTP/1.1
Host: 127.0.0.1:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 107.1777 Total Bytes Received : 1065 Body Length : 931 Is Compressed : No

```
HTTP/1.1 200 OK
server: uvicorn
content-length: 931
content-type: text/html; charset=utf-8
date: Sat, 03 Jun 2023 13:00:38 GMT


<!DOCTYPE html>
<html>
<head>
<link type="text/css" rel="stylesheet" href="https://cdn.jsdelivr.net/npm/swagger-ui-dist@4/swagger-ui.
css">
<link rel="shortcut icon" href="https://fastapi.tiangolo.com/img/favicon.png">
<title>FastAPI - Swagger UI</title>
</head>
<body>
<div id="swagger-ui">
</div>
<script src="https://cdn.jsdelivr.net/npm/swagger-ui-dist@4/swagger-ui-bundle.js"></script>
<!-- `SwaggerUIBundle` is now available on the page -->
<script>
const ui = SwaggerUIBundle({
url: '/openapi.json',
"dom_id": "#swagger-ui",
"layout": "BaseLayout",
"deepLinking": true,
"showExtensions": true,
"showCommonExtensions": true,
oauth2RedirectUrl: window.location.origin + '/docs/oauth2-redirect',
presets: [
SwaggerUIBundle.presets.apis,
SwaggerUIBundle.SwaggerUIStandalonePreset
],
})
</script>
</body>
</html>
```

Remedy

Using Subresource Integrity is simply to add *integrity*attribute to the *script*tag along with a base64 encoded cryptographic hash value.

```
<script src="https://code.jquery.com/jquery-2.1.4.min.js" integrity="sha384-
```

```
R4/ztc4ZlRqWjqIuvf6RX5yb/v90qNGx6fS48N0tRxiGkqveZETq72KgDVJCp2TC" crossorigin="anonymous"></script>
```

The hash algorithm must be one of **sha256**, **sha384**or **sha512**, followed by a '-' character.

External References

- Subresource Integrity
- Do not let your CDN betray you: Use Subresource Integrity
- Web Application Security with Subresource Integrity
- SRI Hash Generator

CLASSIFICATION

SANS Top 25	16
WASC	15
ISO27001	A.14.2.5

7. Expect-CT Security Header Errors and Warnings

INFORMATION 1

Netsparker detected errors during parsing of Expect-CT header.

Vulnerabilities

7.1. http://127.0.0.1:8000/docs#/default/predict_predict__post

Error	Resolution
Expect-CT header should be served with a hostname as known Expect-CT hosts are identified only by domain names, and never IP addresses.	Serve Expect-CT header with a defined hostname.

Certainty

Request

```
GET /docs#/default/predict_predict__post HTTP/1.1
Host: 127.0.0.1:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 107.1777 **Total Bytes Received** : 1065 **Body Length** : 931 **Is Compressed** : No

```
HTTP/1.1 200 OK
server: uvicorn
content-length: 931
content-type: text/html; charset=utf-8
date: Sat, 03 Jun 2023 13:00:38 GMT


<!DOCTYPE html>
<html>
<head>
<link type="text/css" rel="stylesheet" href="https://cdn.jsdelivr.net/npm/swagger-ui-dist@4/swagger-ui.
css">
<link rel="shortcut icon" href="https://fastapi.tiangolo.com/img/favicon.png">
<title>FastAPI - Swagger UI</title>
</head>
<body>
<div id="swagger-ui">
</div>
<script src="https://cdn.jsdelivr.net/npm/swagger-ui-dist@4/swagger-ui-bundle.js"></script>
<!-- `SwaggerUIBundle` is now available on the page -->
<script>
const ui = SwaggerUIBundle({
url: '/openapi.json',
"dom_id": "#swagger-ui",
"layout": "BaseLayout",
"deepLinking": true,
"showExtensions": true,
"showCommonExtensions": true,
oauth2RedirectUrl: window.location.origin + '/docs/oauth2-redirect',
presets: [
SwaggerUIBundle.presets.apis,
SwaggerUIBundle.SwaggerUIStandalonePreset
],
})
</script>
</body>
</html>
```

External References

- Expect-CT Extension for HTTP
- Expect-CT HTTP Header
- Expect-CT

CLASSIFICATION

SANS Top 25	16
WASC	15
OWASP Proactive Controls	C10
ISO27001	A.14.1.2

Show Scan Detail

Enabled Security Checks : Apache Struts S2-045 RCE,
Apache Struts S2-046 RCE,
BREACH Attack,
Code Evaluation,
Code Evaluation (Out of Band),
Command Injection,
Command Injection (Blind),
Content Security Policy,
Content-Type Sniffing,
Cookie,
Cross Frame Options Security,
Cross-Origin Resource Sharing (CORS),
Cross-Site Request Forgery,
Cross-site Scripting,
Cross-site Scripting (Blind),
Custom Script Checks (Active),
Custom Script Checks (Passive),
Custom Script Checks (Per Directory),
Custom Script Checks (Singular),
Drupal Remote Code Execution,
Expect Certificate Transparency (Expect-CT),
Expression Language Injection,
File Upload,
Header Analyzer,
Heartbleed,
HSTS,
HTML Content,
HTTP Header Injection,
HTTP Methods,
HTTP Status,

	HTTP.sys (CVE-2015-1635), IFrame Security, Insecure JSONP Endpoint, Insecure Reflected Content, JavaScript Libraries, Local File Inclusion, Login Page Identifier, Mixed Content, Open Redirection, Referrer Policy, Reflected File Download, Remote File Inclusion, Remote File Inclusion (Out of Band), Reverse Proxy Detection, RoR Code Execution, Server-Side Request Forgery (DNS), Server-Side Request Forgery (Pattern Based), Server-Side Template Injection, Signatures, SQL Injection (Blind), SQL Injection (Boolean), SQL Injection (Error Based), SQL Injection (Out of Band), SSL, Static Resources (All Paths), Static Resources (Only Root Path), Unicode Transformation (Best-Fit Mapping), WAF Identifier, Web App Fingerprint, Web Cache Deception, WebDAV, Windows Short Filename, XML External Entity, XML External Entity (Out of Band)
URL Rewrite Mode	: Heuristic
Detected URL Rewrite Rule(s)	: None
Excluded URL Patterns	: (log|sign)\-?(out|off) exit endsession gtm\.js WebResource\.axd ScriptResource\.axd
Authentication	: None
Scheduled	: No
Additional Website(s)	: None

This report created with 5.8.1.28119-master-bca4e4e
https://www.netsparker.com