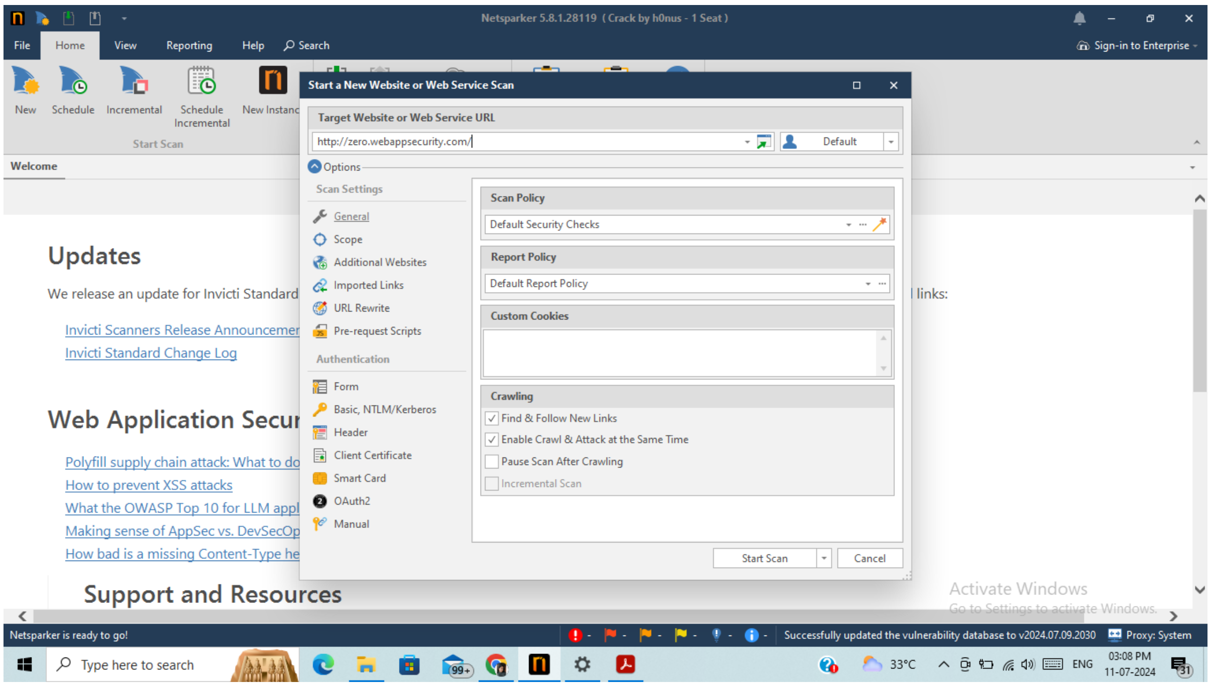
TASK 2

Getting started Netsparker and scanning the URL .



Netsparker 5.8.1.28119 (Crack by h0nus - 1 Seat)

Start a New Website or Web Service Scan

Target Website or Web Service URL: http://zero.webappsecurity.com/ — Default

Options

Scan Settings: General, Scope, Additional Websites, Imported Links, URL Rewrite, Pre-request Scripts

Authentication: Form, Basic, NTLM/Kerberos, Header, Client Certificate, Smart Card, OAuth2, Manual

Scan Policy: Default Security Checks

Report Policy: Default Report Policy

Custom Cookies

Crawling:
- [✓] Find & Follow New Links
- [✓] Enable Crawl & Attack at the Same Time
- [ ] Pause Scan After Crawling
- [ ] Incremental Scan

Start Scan · Cancel

Updates

We release an update for Invicti Standard … links:

Invicti Scanners Release Announcemen…
Invicti Standard Change Log

Web Application Secur…

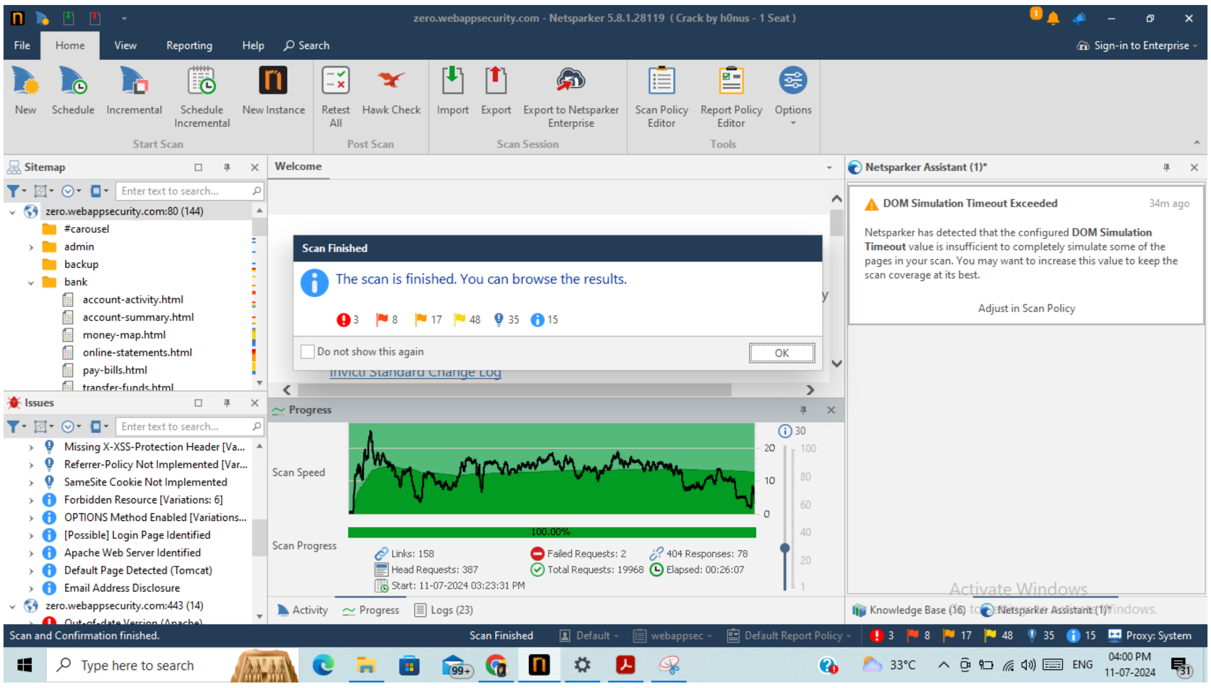
Polyfill supply chain attack: What to do…
How to prevent XSS attacks
What the OWASP Top 10 for LLM appl…
Making sense of AppSec vs. DevSecOp…
How bad is a missing Content-Type he…

Support and Resources

Netsparker is ready to go!

Successfully updated the vulnerability database to v2024.07.09.2030 · Proxy: System

33°C · ENG · 03:08 PM · 11-07-2024



zero.webappsecurity.com - Netsparker 5.8.1.28119 (Crack by h0nus - 1 Seat)

Sitemap: zero.webappsecurity.com:80 (144)
- #carousel
- admin
- backup
- bank
  - account-activity.html
  - account-summary.html
  - money-map.html
  - online-statements.html
  - pay-bills.html
  - transfer-funds.html

Scan Finished

The scan is finished. You can browse the results.

3 · 8 · 17 · 48 · 35 · 15

[ ] Do not show this again — OK

Netsparker Assistant (1)

DOM Simulation Timeout Exceeded — 34m ago

Netsparker has detected that the configured **DOM Simulation Timeout** value is insufficient to completely simulate some of the pages in your scan. You may want to increase this value to keep the scan coverage at its best.

Adjust in Scan Policy

Issues:
- Missing X-XSS-Protection Header [Va…
- Referrer-Policy Not Implemented [Var…
- SameSite Cookie Not Implemented
- Forbidden Resource [Variations: 6]
- OPTIONS Method Enabled [Variations…
- [Possible] Login Page Identified
- Apache Web Server Identified
- Default Page Detected (Tomcat)
- Email Address Disclosure
- zero.webappsecurity.com:443 (14)
  - Out-of-date Version (Apache)

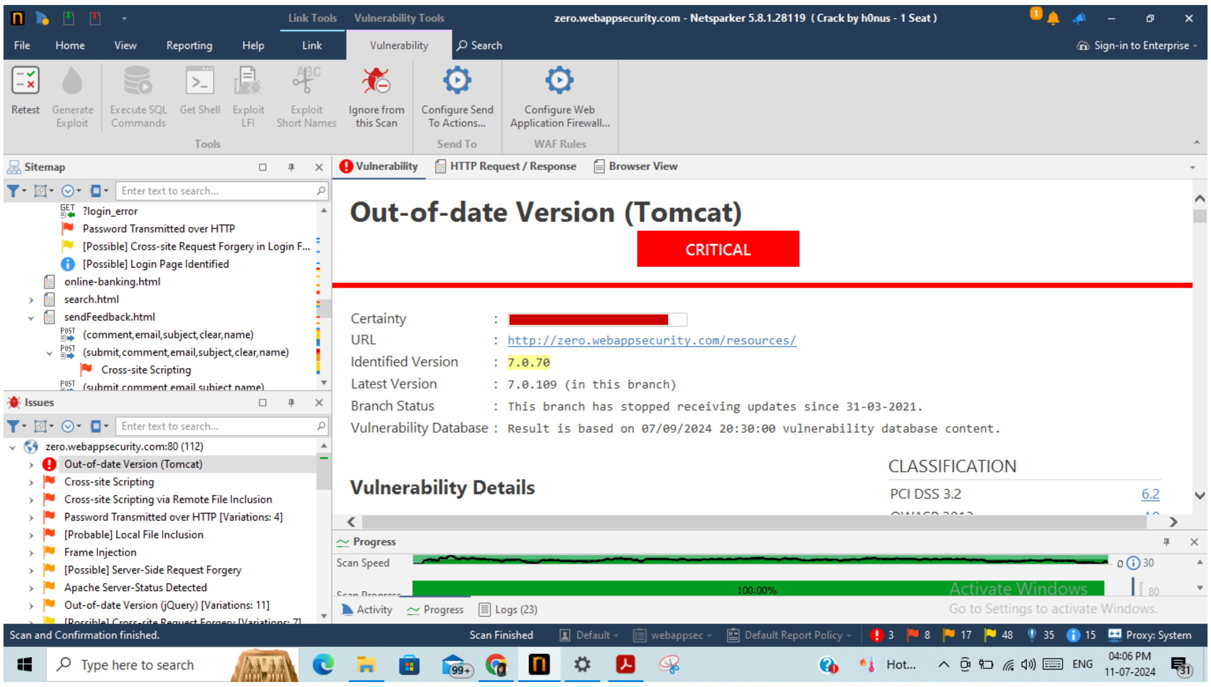
Progress: Scan Speed · Scan Progress 100.00%

Links: 158 · Failed Requests: 2 · 404 Responses: 78
Head Requests: 387 · Total Requests: 19968 · Elapsed: 00:26:07
Start: 11-07-2024 03:23:31 PM

Scan and Confirmation finished. · Scan Finished · Default · webappsec · Default Report Policy · 3 · 8 · 17 · 48 · 35 · 15 · Proxy: System

33°C · ENG · 04:00 PM · 11-07-2024

1) Vulnerability: Out-of-date version of Tomcat.


zero.webappsecurity.com - Netsparker 5.8.1.28119 (Crack by h0nus - 1 Seat)

Sitemap

```
?login_error
Password Transmitted over HTTP
[Possible] Cross-site Request Forgery in Login F...
[Possible] Login Page Identified
online-banking.html
search.html
sendFeedback.html
  (comment,email,subject,clear,name)
  (submit,comment,email,subject,clear,name)
    Cross-site Scripting
```

Issues

```
zero.webappsecurity.com:80 (112)
  Out-of-date Version (Tomcat)
  Cross-site Scripting
  Cross-site Scripting via Remote File Inclusion
  Password Transmitted over HTTP [Variations: 4]
  [Probable] Local File Inclusion
  Frame Injection
  [Possible] Server-Side Request Forgery
  Apache Server-Status Detected
  Out-of-date Version (jQuery) [Variations: 11]
```

Out-of-date Version (Tomcat)

CRITICAL

```
Certainty              :
URL                    : http://zero.webappsecurity.com/resources/
Identified Version     : 7.0.70
Latest Version         : 7.0.109 (in this branch)
Branch Status          : This branch has stopped receiving updates since 31-03-2021.
Vulnerability Database : Result is based on 07/09/2024 20:30:00 vulnerability database content.
```

Vulnerability Details

CLASSIFICATION

PCI DSS 3.2 6.2

Scan and Confirmation finished.

REPORT:

Vulnerability 1: Out-of-date version of Tomcat

URL: http://zero.webappsecurity.com/resources/

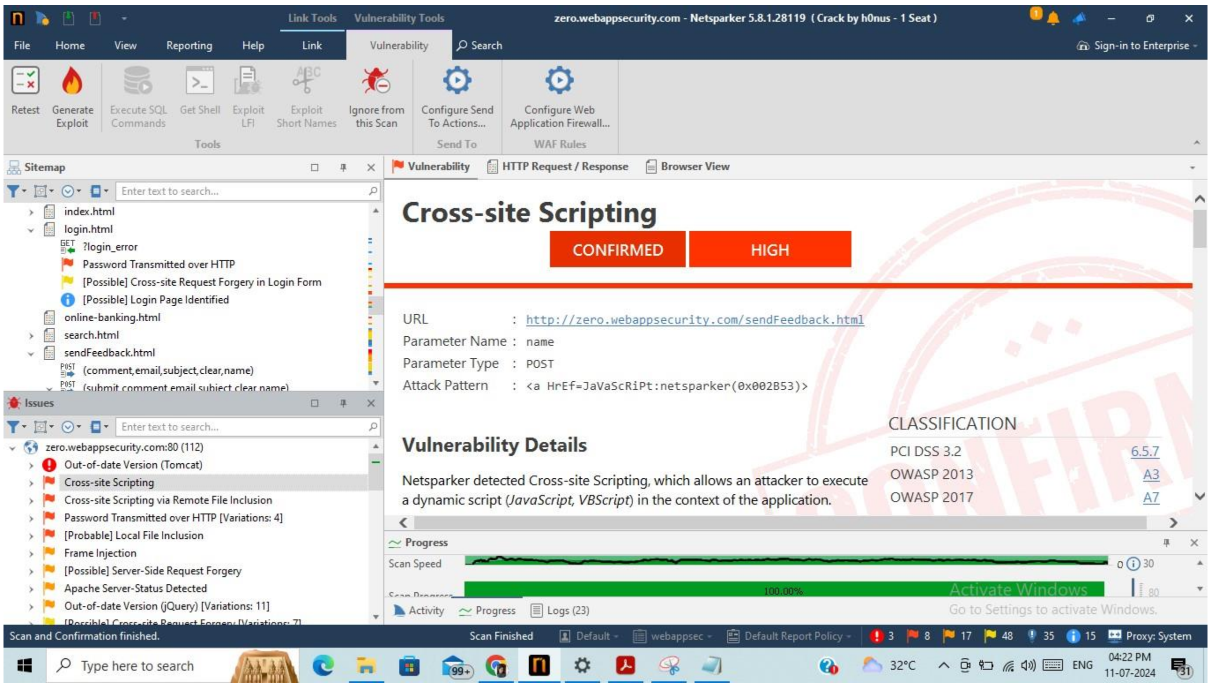
Identified Version: 7.0.70

Vulnerability Database: Result is based on 07/09/2024 20:30:00 vulnerability database content.

Vulnerability Details: Net sparker identified you are using an out-of-date version of Tomcat.

Remedy: Please upgrade your installation of Tomcat to the latest stable version.

2) Vulnerability: Cross site Scripting



Sitemap:

- index.html
- login.html
  - ?login_error
  - Password Transmitted over HTTP
  - [Possible] Cross-site Request Forgery in Login Form
  - [Possible] Login Page Identified
- online-banking.html
- search.html
- sendFeedback.html
  - (comment,email,subject,clear,name)
  - (submit,comment,email,subject,clear,name)

Issues:

- zero.webappsecurity.com:80 (112)
  - Out-of-date Version (Tomcat)
  - Cross-site Scripting
  - Cross-site Scripting via Remote File Inclusion
  - Password Transmitted over HTTP [Variations: 4]
  - [Probable] Local File Inclusion
  - Frame Injection
  - [Possible] Server-Side Request Forgery
  - Apache Server-Status Detected
  - Out-of-date Version (jQuery) [Variations: 11]
  - [Possible] Cross-site Request Forgery [Variations: 7]

Cross-site Scripting

CONFIRMED HIGH

URL : http://zero.webappsecurity.com/sendFeedback.html
Parameter Name : name
Parameter Type : POST
Attack Pattern : <a HrEf=JaVaScRiPt:netsparker(0x002B53)>

Vulnerability Details

Netsparker detected Cross-site Scripting, which allows an attacker to execute a dynamic script (*JavaScript, VBScript*) in the context of the application.

CLASSIFICATION

PCI DSS 3.2	6.5.7
OWASP 2013	A3
OWASP 2017	A7

Scan and Confirmation finished.

REPORT:

URL: http://zero.webappsecurity.com/sendFeedback.html

Vulnerability 2: Cross site Scripting

Parameter Name: name

Parameter Type: POST

Attack Pattern: <a HrEf=JaVaScRiPt:netsparker(0x002B53)>

Vulnerability Details:
 Netsparker detected Cross-site Scripting, which allows an attacker to execute a dynamic script (JavaScript, VBScript) in the context of the application.
This allows several different attack opportunities, mostly hijacking the current session of the user or changing the look of the page by changing the HTML on the fly to steal the user's credentials.

This happens because the input entered by a user has been interpreted as HTML/JavaScript/VBScript by the browser.
Cross-site scripting targets the users of the application instead of the server. Although this is a limitation, since it allows attackers to hijack other users' sessions, an attacker might attack an administrator to gain full control over the application.

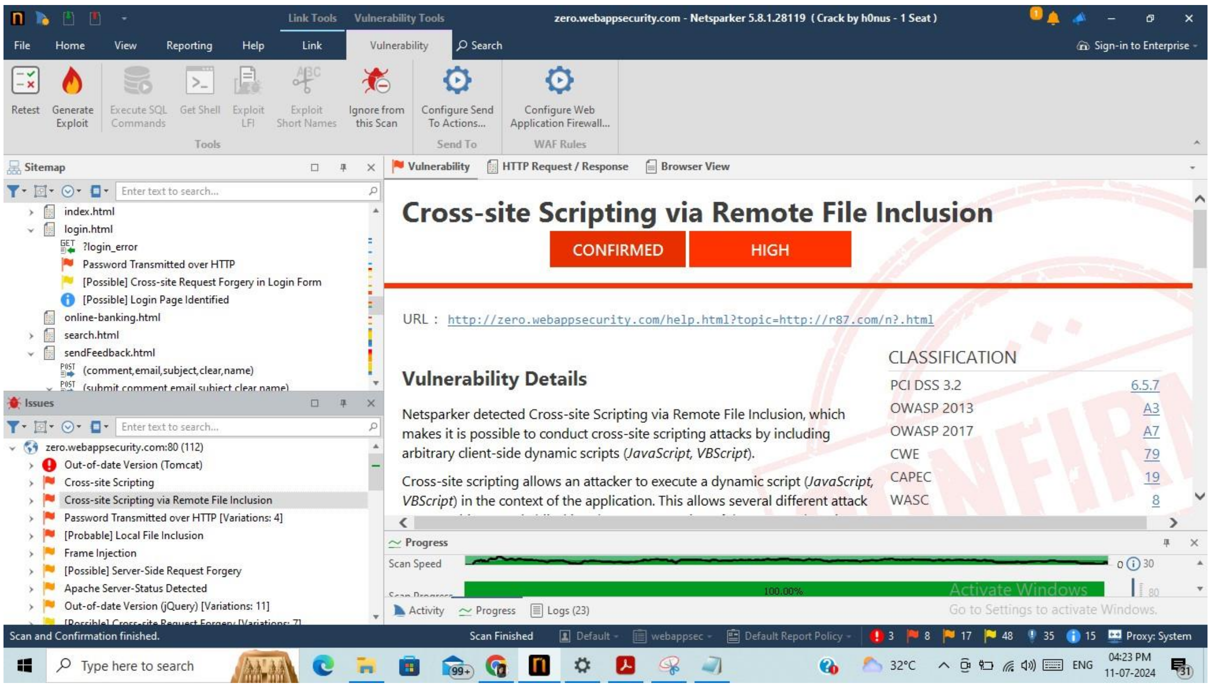
Remedy:

The issue occurs because the browser interprets the input as active HTML, JavaScript or VBScript. To avoid this, output should be encoded according to the output location and context.
For example, if the output goes in to a JavaScript block within the HTML document, then output needs to be encoded accordingly. Encoding can get very complex, therefore it's strongly recommended to use an encoding library such as OWASP ESAPI and Microsoft Anti-cross-site scripting.

Additionally, you should implement a strong Content Security Policy (CSP) as a defense-in-depth measure if an XSS vulnerability is mistakenly introduced. Due to the complexity of XSS-Prevention and the lack of secure standard behavior in programming languages and frameworks, XSS vulnerabilities are still common in web applications.

CSP will act as a safeguard that can prevent an attacker from successfully exploiting Cross-site Scripting vulnerabilities in your website and is advised in any kind of application. Please make sure to scan your application again with Content Security Policy checks enabled after implementing CSP, in order to avoid common mistakes that can impact the effectiveness of your policy. There are a few pitfalls that can render your CSP policy useless and we highly recommend reading the resources linked in the reference section before you start to implement one.

3) Vulnerability : Cross- site Scripting via remote file inclusion



Sitemap:

- index.html
- login.html
  - ?login_error
  - Password Transmitted over HTTP
  - [Possible] Cross-site Request Forgery in Login Form
  - [Possible] Login Page Identified
- online-banking.html
- search.html
- sendFeedback.html
  - (comment,email,subject,clear,name)
  - (submit,comment,email,subject,clear,name)

Issues:

- zero.webappsecurity.com:80 (112)
  - Out-of-date Version (Tomcat)
  - Cross-site Scripting
  - Cross-site Scripting via Remote File Inclusion
  - Password Transmitted over HTTP [Variations: 4]
  - [Probable] Local File Inclusion
  - Frame Injection
  - [Possible] Server-Side Request Forgery
  - Apache Server-Status Detected
  - Out-of-date Version (jQuery) [Variations: 11]
  - [Possible] Cross-site Request Forgery [Variations: 7]

Cross-site Scripting via Remote File Inclusion

CONFIRMED HIGH

URL : http://zero.webappsecurity.com/help.html?topic=http://r87.com/n?.html

Vulnerability Details

Netsparker detected Cross-site Scripting via Remote File Inclusion, which makes it is possible to conduct cross-site scripting attacks by including arbitrary client-side dynamic scripts (*JavaScript, VBScript*).

Cross-site scripting allows an attacker to execute a dynamic script (*JavaScript, VBScript*) in the context of the application. This allows several different attack

CLASSIFICATION	
PCI DSS 3.2	6.5.7
OWASP 2013	A3
OWASP 2017	A7
CWE	79
CAPEC	19
WASC	8

Scan and Confirmation finished.

REPORT:

URL: http://zero.webappsecurity.com/help.html?topic=http://r87.com/n?.html

Vulnerability 3: Cross- site Scripting via remote file inclusion

Vulnerability Details: Netsparker detected Cross-site Scripting via Remote File Inclusion, which makes it is possible to conduct cross-site scripting attacks by including arbitrary client-side dynamic scripts (JavaScript, VBScript).

Cross-site scripting allows an attacker to execute a dynamic script (JavaScript, VBScript) in the context of the application. This allows several different attack opportunities, mostly hijacking the current session of the user or changing the look of the page by changing the HTML on the fly to steal the user's credentials. This happens because the input entered by the user has been interpreted as HTML/JavaScript/VBScript by the browser.

Cross-site scripting targets the users of the application instead of the server. Although this is limitation, since it allows attackers to hijack other users' sessions, an attacker might attack an administrator to gain full control over the application.

Impact:

There are many different attacks that can be leveraged through the use of cross-site scripting, including:
Hijacking user's active session.

Changing the look of the page within the victim's browser.

Mounting a successful phishing attack.

Intercepting data and performing man-in-the-middle attacks.

Remedy

The issue occurs because the browser interprets the input as active HTML, Javascript or VbScript. To avoid this, all input and output from the application should be filtered. Output should be filtered according to the output format and location. Typically, the output location is HTML. Where the output is HTML, ensure all active content is removed prior to its presentation to the server.

Additionally, you should implement a strong Content Security Policy (CSP) as a defence-in-depth measure if an XSS vulnerability is mistakenly introduced. Due to the complexity of XSS-Prevention and the lack of secure standard behavior in programming languages and frameworks, XSS vulnerabilities are still common in web applications.

CSP will act as a safeguard that can prevent an attacker from successfully exploiting Cross Site Scripting vulnerabilities in your website and is advised in any kind of application. Please make sure to scan your application again with Content Security Policy checks enabled after implementing CSP, in order to avoid common mistakes that can impact the effectiveness of your policy. There are a few pitfalls that can render your CSP policy useless and we highly recommend reading the resources linked in the reference section before you start to implement one.