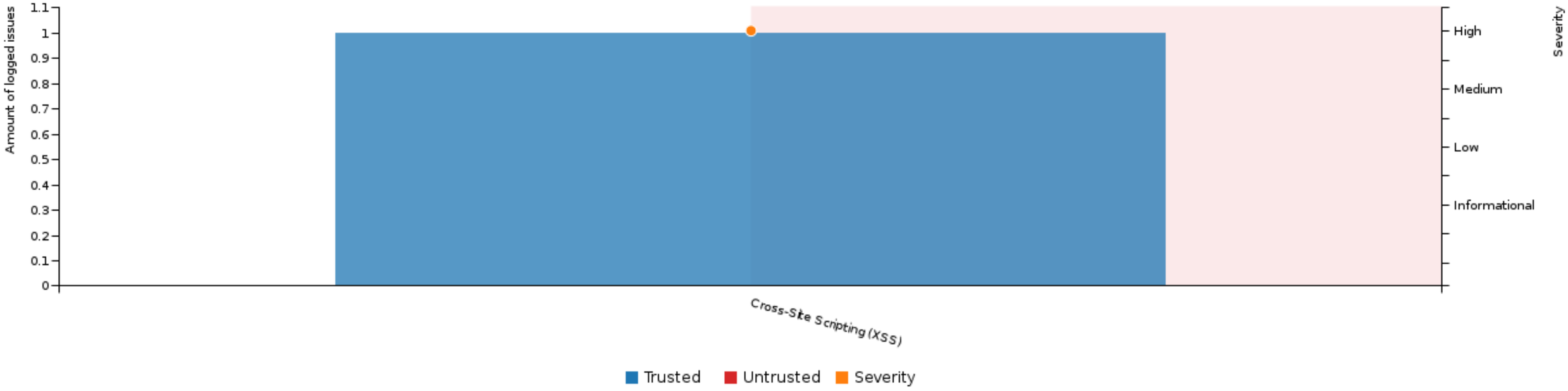
http://testhtml5.vulnweb.com/ Generated on 2017-06-01 11:12:17 +0000

Summary

Charts | Issues 1 | OWASP Top 10

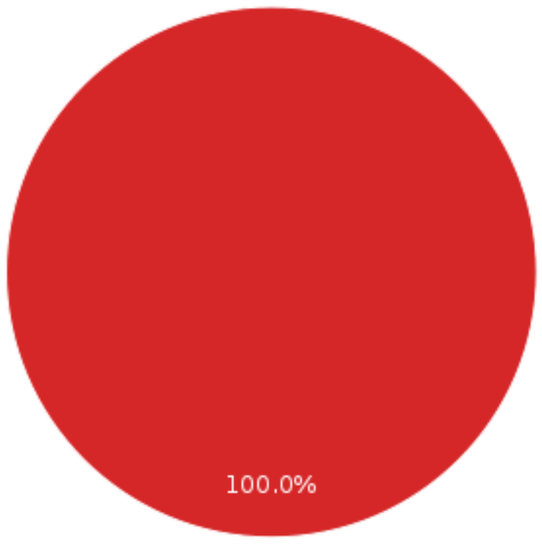
Issues by type, trust, and severity

(Click on the bars or line points for details on the relevant issues.)

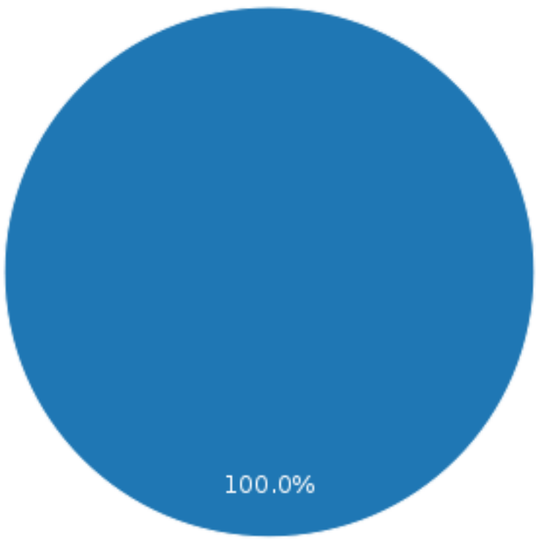


Severities of issues based on possible impact

(Click to see relevant Trusted issues.)

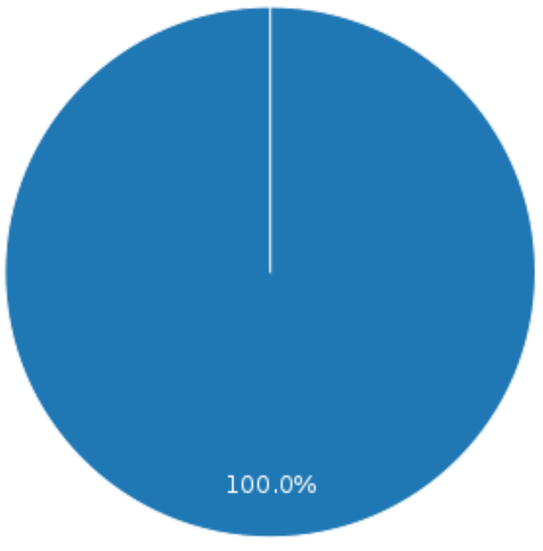


Elements with issues, by type



Trust evaluation (Trusted vs. Untrusted) of issues

(Click to see relevant issues.)



http://testhtml5.vulnweb.com/ Generated on 2017-06-01 11:12:17 +0000

Issues

Trusted 1

At the time these issues were logged there were no abnormal interferences or anomalous server behavior.
These issues are considered trusted and accurate.

High 1

Cross-Site Scripting (XSS) 1 xss

Client-side scripts are used extensively by modern web applications. They perform from simple functions (such as the formatting of text) up to full manipulation of client-side data and Operating System interaction.

Cross Site Scripting (XSS) allows clients to inject scripts into a request and have the server return the script to the client in the response. This occurs because the application is taking untrusted data (in this example, from the client) and reusing it without performing any validation or sanitisation.

If the injected script is returned immediately this is known as reflected XSS. If the injected script is stored by the server and returned to any client visiting the affected page, then this is known as persistent XSS (also stored XSS).

Arachni has discovered that it is possible to insert script content directly into HTML element content.

Remediation guidance

To remedy XSS vulnerabilities, it is important to never use untrusted or unfiltered data within the code of a HTML page.

Untrusted data can originate not only form the client but potentially a third party or previously uploaded file etc.

Filtering of untrusted data typically involves converting special characters to their HTML entity encoded counterparts (however, other methods do exist, see references).
These special characters include:

- `&`
- `<`
- `>`
- `"`
- `'`
- `/`

An example of HTML entity encoding is converting `<` to `<`.

Although it is possible to filter untrusted input, there are five locations within an HTML page where untrusted input (even if it has been filtered) should never be placed:

1. Directly in a script.
2. Inside an HTML comment.
3. In an attribute name.
4. In a tag name.
5. Directly in CSS.

Each of these locations have their own form of escaping and filtering.

Because many browsers attempt to implement XSS protection, any manual verification of this finding should be conducted using multiple different browsers and browser versions.

References

CWE-79

Secunia
WASC
OWASP

In `cookie` input `username` using `GET` at http://testhtml5.vulnweb.com/logout pointing to http://testhtml5.vulnweb.com/logout .

Injected seed	Proof
`<xss_1e03bb386adf05a158bb746c0451f704/>`	`<xss_1e03bb386adf05a158bb746c0451f704/>`

Vector information

Affected page: `http://testhtml5.vulnweb.com/`

Referring page: `http://testhtml5.vulnweb.com/logout`

http://testhtml5.vulnweb.com/ Generated on 2017-06-01 11:12:17 +0000

Plugin results

Health map

Generates a simple list of safe/unsafe URLs.

Total	8	http://testhtml5.vulnweb.com/logout
Without issues	7	http://testhtml5.vulnweb.com/
With issues	1	http://testhtml5.vulnweb.com/#/popular
Issue percentage	13	http://testhtml5.vulnweb.com/.fluid-container
		http://testhtml5.vulnweb.com/.well
		http://testhtml5.vulnweb.com/forgotpw
		http://testhtml5.vulnweb.com/login
		http://testhtml5.vulnweb.com/span

http://testhtml5.vulnweb.com/ Generated on 2017-06-01 11:12:17 +0000

Sitemap 8

HTTP status code	URL
200	http://testhtml5.vulnweb.com/
200	http://testhtml5.vulnweb.com/#/popular
404	http://testhtml5.vulnweb.com/.fluid-container
404	http://testhtml5.vulnweb.com/.well
200	http://testhtml5.vulnweb.com/forgotpw
200	http://testhtml5.vulnweb.com/login
302	http://testhtml5.vulnweb.com/logout
404	http://testhtml5.vulnweb.com/span