Around the world in 80 commits

A tale of contributing to the OWASP cheatsheet project...

Authoritative Sources

README.md

Welcome to the OWASP Cheat Sheet Series

Welcome to the official repository for the Open Web Application Security Project® (OWASP) Cheat Sheet Series project. The project focuses on providing good security practices for builders in order to secure their applications.

In order to read the cheat sheets and **reference** them, use the project's official website. The project details can be viewed on the OWASP main website without the cheat sheets.

https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Session_Management_Cheat_Sheet.md

aussieklutz commented on May 22, 2019 Contributor

Thanks you for proposing a update/refactor of a cheat sheet.

Please provides the following information about your proposal:

- What update/refactoring do you want to perform?

Web apps are starting to use the HTML5 Web Storage API's, in particular the window.sessionStorage api for session identifier storage. This provides protection from cross protocol (http -> https), cross-window, and retention beyond the window lifetime. It does however allow, by design, access from JS. It may also be vulnerable to TLS downgrade (cipher: null) attacks. Some addressing of this new mechanism would be valuable.

Thanks you again for your contribution 😃

mackowski commented on May 22, 2019 Collaborator

Thanks for this proposal. Looks fine for me!
@righettod what is your view on that?

righettod added this to **Backlog** in **Roadmap 2019** via automation on May 23, 2019

righettod added this to the **Roadmap 2019** milestone on May 23, 2019

righettod assigned **aussieklutz** on May 23, 2019

righettod added ACK_OBTAINED and removed ACK_WAITING labels on May 23, 2019

righettod commented on May 23, 2019 Contributor

This topic can effectively be interesting to deep dive to provide useful technical hints.

@aussieklutz Do you want to work on it?

righettod changed the title ~~Cheat sheet update/refactor proposal: [Session_Management_Cheat_Sheet]~~ Cheat sheet update/refactor proposal: Session Management Cheat Sheet on May 23, 2019

righettod unassigned **aussieklutz** on May 23, 2019

aussieklutz commented on May 23, 2019 Contributor Author

Happy to draft a first pass at it...

How to setup my contributor environment

Follow these steps:

1. Install Visual Studio Code (VSCode).
2. Install the vscode-markdownlint plugin.
3. Open the file Project.code-workspace from VSCode via the menu `File > Open Workspace...`.
4. You are ready to contribute 👍

⏰ What to verify before pushing the updates?

1. Ensure that the markdown files you have created or modified do not have any warnings/errors raised by the linter. You can see it in this bottom bar when the markdown file is opened in VSCode:

2. Ensure that the markdown file you have created/modified do not have any dead links. You can verify that by using this plugin. If you cannot use this plugin then, verify that all the links you have changed or added are valid before pushing.
    i. Install NodeJS to install NPM.
    ii. Install the validation plugin via the command `npm install -g markdown-link-check`
    iii. Use this command (from the repository root folder) on your markdown file to verify the presence of any dead links:

```
markdown-link-check -c .markdownlinkcheck.json [MD_FILE]
```

The should produce output similar to the below. Any identified dead links are shown using a red cross instead of a green tick before the link.

```
$ markdown-link-check -c .markdownlinkcheck.json cheatsheets/Transaction_Authorization_Cheat_Sheet.md
FILE: cheatsheets/Transaction_Authorization_Cheat_Sheet.md
[✓] https://en.wikipedia.org/wiki/Time-based_One-time_Password_Algorithm
[✓] https://en.wikipedia.org/wiki/Chip_Authentication_Program
[✓] http://www.cl.cam.ac.uk/~sjm217/papers/fc09optimised.pdf
...
```

https://github.com/OWASP/CheatSheetSeries/blob/master/CONTRIBUTING.md

TL;DR

- They write in MarkDown
- They use VSCode, and a markdown linting plugin to ensure that the formatting meets a baseline
- Commits are pushed through a CI chain to validate that commits are appropriately formatted

```
166 + # HTML5 Web Storage API
167 +
168 + The Web Hypertext Application Technology Working Group
    (WHATWG) describes the HTML5 Web Storage APIs,
    `localStorage` and `sessionStorage`, as mechanisms for
    storing name-value pairs client-side.
169 + Unlike HTTP cookies, the contents of `localStorage` and
    `sessionStorage` are not automatically shared within
    requests or responses by the browser and are used for
    storing data client-side.
```

Add section on HTML5 Web Storage APIs #115

Edit | Open with

Merged righettod merged 8 commits into OWASP:master from unknown repository on Jun 2, 2019

Conversation 62 | Commits 8 | Checks 1 | Files changed 1 | +57 −0

Add section on HTML5 Web Storage API acda37b

Travis CI

Travis CI - Pull Request

Travis CI / Travis CI - Pull Request

succeeded on May 24, 2019 in 1m 36s

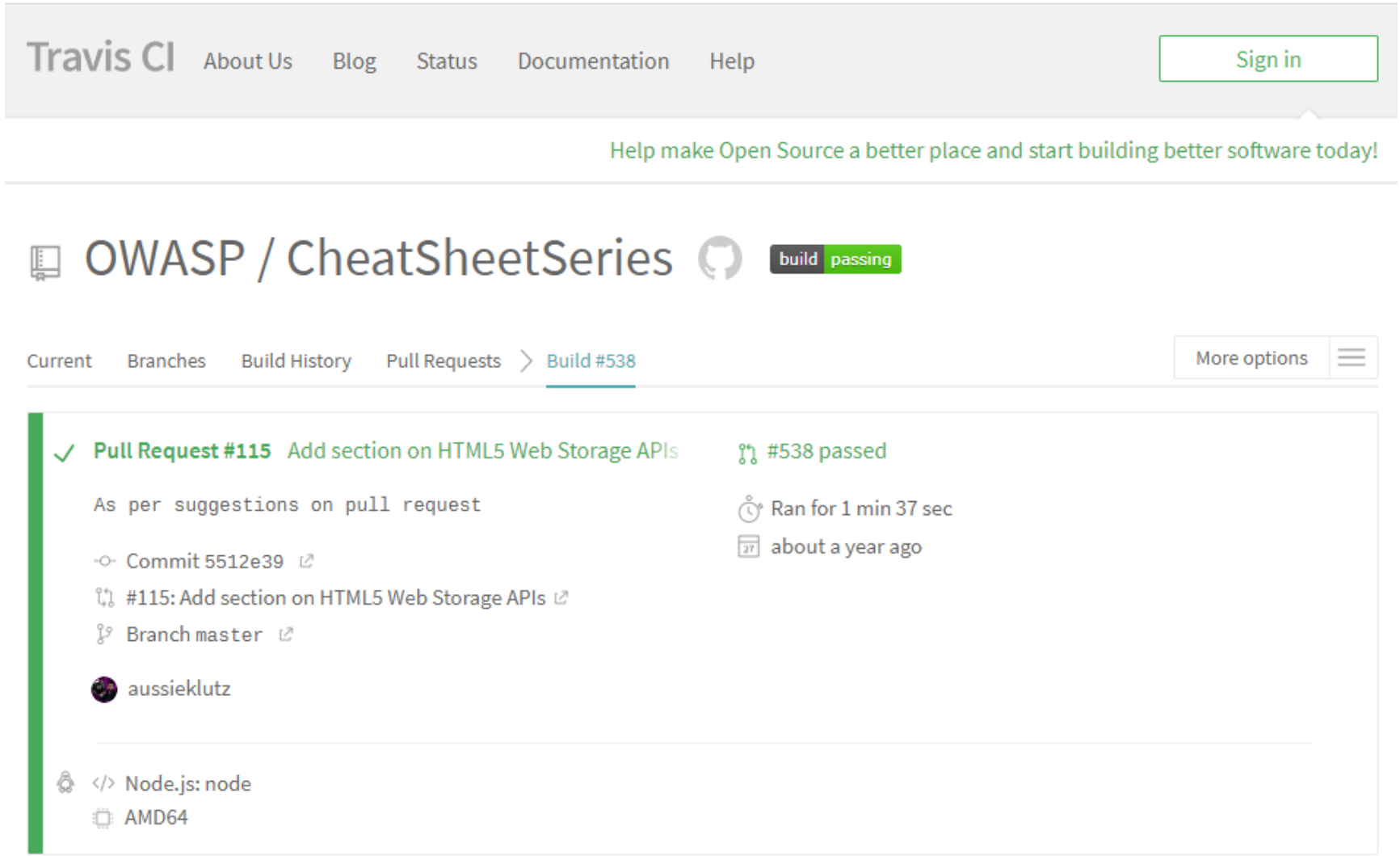
Build Passed

✓ The build **passed**, just like the previous build.

DETAILS

This is a pull request build.

It is running a build against the merge commit, after merging #115 Add section on HTML5 Web Storage APIs.
Any changes that have been made to the master branch before the build ran are also included.


Travis CI
About Us
Blog
Status
Documentation
Help
Sign in
Help make Open Source a better place and start building better software today!
OWASP / CheatSheetSeries
build passing
Current
Branches
Build History
Pull Requests
Build #538
More options
Pull Request #115 Add section on HTML5 Web Storage APIs
As per suggestions on pull request
Commit 5512e39
#115: Add section on HTML5 Web Storage APIs
Branch master
aussieklutz
#538 passed
Ran for 1 min 37 sec
about a year ago
Node.js: node
AMD64

righettod suggested changes on May 28, 2019

View changes

righettod left a comment

Contributor

Thank you very much for the PR 😃
I have added comments in order to clarify some points and provide additional hints.

cheatsheets/Session_Management_Cheat_Sheet.md Outdated Hide resolved

```
@@ -163,6 +163,49 @@ Typically, session management capabilities to track users after authentication m
 - Ensure entire cookie should be encrypted if sensitive data is persisted in the cookie
 - Define all cookies being used by the application, their name and why they are needed

+ # HTML5 Web Storage API
+
+ The Web Hypertext Application Technology Working Group (WHATWG) describes the HTML5 Web Storage APIs, localStorage and sessionStorage, as mechanisms for storing name-value pairs client-side.
```

righettod on May 28, 2019 Contributor

Highlight the sentense *localStorage and sessionStorage* like this: `localStorage` and `sessionStorage`.

Repeat this along the section each time these both keywords are used.

Reply...

Unresolve conversation

ThunderSon marked this conversation as resolved.

cheatsheets/Session_Management_Cheat_Sheet.md Outdated Hide resolved

```
172 +
173 + ### Scope
174 +
175 + Data stored using the localStorage API is accessible by pages which are
    loaded from the same protocol, port and host.
```

righettod on May 28, 2019 Contributor

According to this + this, the origin that define access policy is defined by:

- A scheme (a scheme).
- A host (a host).
- A port (a port).
- A domain (null or a domain). Null unless stated otherwise.

So it's important to mention the 4 elements explicitly.

ThunderSon on May 28, 2019 Contributor

A domain follows the above. It's the same. Later in the provided links you mentioned, in order to identify if origins match, the following is specified:
`If A and B are both tuple origins and their schemes, hosts, and port are identical, then return true.`
What is written seems to be alright, as pages are defined by the protocol (scheme), the host (domain) and its port.

righettod on May 28, 2019 • edited Contributor

OK, thanks for the remark 😃
Perhaps we can add between parenthesis : `protocol (scheme)` to indicate that scheme refer to the protocol.
What do you think?

ThunderSon on May 28, 2019 Contributor

I don't mind at all to stay hand in hand with the official docs.

righettod on May 28, 2019 Contributor

Let's wait the feedback from the author.

aussieklutz on May 30, 2019 Author Contributor

Happy to modify to use the scheme/host/port nomenclature.

ThunderSon commented on May 30, 2019 — Contributor

@aussieklutz awesome update! I like how you modified it.

aussieklutz commented on May 31, 2019 — Contributor, Author

Thanks @ThunderSon, I have implemented the further comments you made.

`Spelling revisions, etc.` ... Verified ✓ 624a19f

ThunderSon approved these changes on May 31, 2019 — View changes

mackowski approved these changes on May 31, 2019 — View changes

ThunderSon added the UPDATE_CS label on May 31, 2019

mackowski approved these changes on May 31, 2019 — View changes

mackowski requested a review from **righettod** on May 31, 2019

righettod suggested changes on Jun 1, 2019 — View changes

righettod left a comment — Contributor

Just change the reference section and it will be OK.
Thanks for the PR 😃

`Update Web Storage APIs - reference section to named links` ··· Verified ✓ `52b456c`

righettod approved these changes on Jun 2, 2019 View changes

righettod left a comment Contributor

Thank you very much 👍

righettod merged commit `0495e19` into `OWASP:master` on Jun 2, 2019 View details

2 checks passed

Roadmap 2019 automation moved this from **Pending** to **Done** on Jun 2, 2019

https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Session_Management_Cheat_Sheet.md#html5-web-storage-api

If you want to have a further dig into my journey, feel free to visit https://github.com/OWASP/CheatSheetSeries/pull/115 and see how it all came together.