HTML5 Web Security

Michael Schmidt – IT Security Analyst
michael.schmidt@csnc.ch
What’s this talk about?
What is HTML5?
HTML5 TEST - http://html5test.com/

Your browser scores 328 and 13 bonus points out of a total of 450 points.

**Parsing rules**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;!DOCTYPE html&gt;</code> triggers standards mode</td>
<td>✔</td>
</tr>
<tr>
<td>HTML5 tokenizer</td>
<td>✔</td>
</tr>
<tr>
<td>HTML5 tree building</td>
<td>✔</td>
</tr>
<tr>
<td>HTML5 defines rules for embedding SVG and MathML inside a regular HTML document. Support for SVG and MathML is not required though, so bonus points are awarded if your browser supports embedding these two technologies.</td>
<td>✔</td>
</tr>
<tr>
<td>SVG in <code>text/html</code></td>
<td>✔</td>
</tr>
<tr>
<td>MathML in <code>text/html</code></td>
<td>✔</td>
</tr>
</tbody>
</table>

Out of a total of 450 points.
Overview

- The Web Sockets API
- Custom scheme and content handlers
- Cross-Origin Resource Sharing
- Web Messaging
- Location information
- Data for offline use
- Javascript threads
- Data stored on client
- Geolocation API
- Offline Web Application
- Web Storage
- Web Worker

© Compass Security AG  www.csnc.ch
Vulnerabilities, Threats and Countermeasures (if any)
Cross-Origin Resource Sharing

- The Web Sockets API
- Custom scheme and content handlers
- Cross-Origin Resource Sharing

- Web Messaging
- Location information
- Data for offline use
- Javascript threads
- Data stored on client

- Geolocation API
- Offline Web Application
- Web Storage
- Web Worker

© Compass Security AG www.csnc.ch
Cross-Origin Resource Sharing I
GET / HTTP/1.1
Host: domainB.csnc.ch
Origin: http://domainA.csnc.ch

HTTP/1.1 200 OK
Content-Type: text/html
Access-Control-Allow-Origin: http://domainA.csnc.ch
CORS – Vulnerabilities & Threats I

- Accessing internal websites
- Scanning the internal network
Remote attacking a web server

Easier exploiting of Cross-Site Request Forgery (XSRF)

Establishing a remote shell (DEMO)
**Countermeasures**

Use the `Access-Control-Allow-Origin` header to restrict the allowed domains.

Never set the header to *.

Do not base access control on the origin header.

To mitigate DDoS attacks the Web Application Firewall (WAF) needs to block CORS requests if they arrive in a high frequency.
Web Storage

- Domain A, Domain B, Domain C

- The Web Sockets API
- Custom scheme and content handlers
- Cross-Origin Resource Sharing

- HTML5 Overview
- Domain A, Domain B, Domain C

- Web Messaging
- IFrame Sandboxing

- Location information
- Data for offline use
- Javascript threads

- Geolocation API
- Offline Web Application
- Web Worker

- © Compass Security AG
- www.csnc.ch
Web Storage

Local Storage Example

Please insert your String to be saved in Local Storage here ...

Click this link to save the String into the Local Storage of your browser (key=TestValue).
Click this link to load the saved value from the Local Storage of your browser (key=TestValue).
Click this link to delete the saved value from the Local Storage of your browser (key=TestValue).

Value of field TestValue:

Please insert your String to be saved in Local Storage here ...

```
window > history

| globalStorage | 0 items in Global Storage |
| history       | 1 history entries         |
| innerHeight   | 882                       |
| innerWidth    | 510                       |
| length        | 0                         |

localStorage

1 items in Storage TestValue="Please insert your String to be saved in Local Storage here ..."

http://internet.csnc.ch/helloWorld/localStorageExample/list ( constructor=Location, host="internal.csnc.ch", more... )
BarProp ( constructor=BarProp, visible=true )
BarProp ( constructor=BarProp, visible=true )
```
Web Storage – Vuln. & Threats

Session Hijacking
- If session identifier is stored in local storage, it can be stolen with JavaScript.
- No `HTTPOnly` flag.

Disclosure of Confidential Data
- If sensitive data is stored in the local storage, it can be stolen with JavaScript.

User Tracking
- Additional possibility to identify a user.

Persistent attack vectors
- Attack vectors can be stored persistently in the victim’s browser.
Countermeasures

Use cookies instead of Local Storage for session handling.

Do not store sensitive data in Local Storage.
Offline Web Application

The Web Sockets API

Custom scheme and content handlers

Cross-Origin Resource Sharing

HTML5 Overview

<iframe src="anydomainA.csnc.ch [...]" width="100%" height="200"

Web Messaging

<iframe src="anydomainB.csnc.ch [...]"

Location information

Data for offline use

Javascript threads

Geolocation API

Web Storage

Web Worker
<!DOCTYPE HTML>
<html manifest="/cache.manifest">
<body>
...

**Example cache.manifest**

```
CACHE MANIFEST
/style.css
/helper.js
/csnc-logo.jpg
NETWORK:
/visitor_counter.jsp
FALLBACK:
/ /offline_Error_Message.html
```
OWA – Vulnerabilities & Threats

**Cache Poisoning**
- Caching of the root directory possible.
- HTTP and HTTPs caching possible.

**Persistent attack vectors**
- Attack vectors can be stored persistently in the victim’s browser.

**User Tracking**
- Additional possibility to identify a user.
- Unique identifiers could be stored along with the cached files.
Offline Web Application – Attack 1/2
Offline Web Application – Attack 2/2

7: Request to www.filebox-solution.com
Poisoned content is loaded from browser cache

8: POST /listener.do HTTP/1.1
username=admin&password=compass.123

9: JavaScript Execution

10: Login request

11: HTTP/1.1 200 OK
Login successful
User-Training
Web Messaging

Embedding HTML Page
internal.csnc.ch

postMessage()

<iframe src="external.csnc.ch" [...]

Stealing confidential data

- Sensitive data may be sent accidentally to a malicious Iframe.

Expands attack surface to the client

- Iframes can send malicious content to other Iframes.
- Input validation on the server is not longer sufficient.
Countermeasures

The target in `postMessage()` should be defined explicitly and not set to *.

The receiving Iframe should not accept messages from any domain. 
_E.g._ `e.origin == "http://internal.csnc.ch"

The received message needs to be validated on the client to avoid malicious content being executed.
Custom scheme and content handlers

- The Web Sockets API
- Cross-Origin Resource Sharing
- Web Messaging
- Data for offline use
- Javascript threads
- Data stored on client
- Geolocation API
- Offline Web Application
- Web Storage
- Web Worker
Custom scheme and content handlers

Stealing confidential data
- An attacker tricks the user to register a malicious website as the e-mail protocol handler.
- Sending e-mails through this web application gives the attacker access to the content of the e-mail.

User Tracking
- Additional possibility to identify a user.
- Unique identifiers could be stored along with the protocol handler.
Countermeasures

User-Training
Web Sockets API

The Web Sockets API provides a way for web browsers to establish and maintain a connection to a server. This allows for real-time communication between the client and server, enabling applications to send and receive data in real-time.

Key components of the Web Sockets API include:

- Geolocation API: A way for web browsers to get information about the user's location.
- Offline Web Application: Enables web applications to work in offline mode.
- Web Storage: A method to store data on the client's machine.
- Web Worker: A way to run JavaScript code in the background, allowing for tasks to be performed without blocking the main thread.

Custom scheme and content handlers, as well as Cross-Origin Resource Sharing, are also important aspects of the Web Sockets API.

Web Messaging:

- IFrame Sandboxing: Helps to prevent cross-site scripting attacks.
- Location information: Used to determine the user's location.
- Data for offline use: Enables applications to work offline.
- Javascript threads: Allows for running multiple concurrent scripts.

Data stored on client:

- Web Worker: A process that runs in the background, allowing for more complex tasks to be performed.

HTML5 Overview:

- DomainA.csnc.ch: A web application.
- DomainB.csnc.ch: Another web application.
- DomainC.csnc.ch: A different web application.

© Compass Security AG
Web Sockets API

1: GET / HTTP/1.1

1.1: HTTP/1.1 200 OK

2: Upgrade: WebSocket

2.1: HTTP/1.1 101 Web Socket Protocol Handshake

Full-Duplex TCP-Channel established
Web Sockets API – Vuln. & Threats

Cache Poisoning
✦ A misunderstanding proxy could lead to a cache poisoning vulnerability.

Scanning the internal network
✦ The browser of a victim can be used for port scanning of internal networks.

Establishing a remote shell
✦ Web Sockets can be used to establish a remote shell to a victim’s browser.
Geolocation API

- The Web Sockets API
- Custom scheme and content handlers
- Cross-Origin Resource Sharing

- Location information
- Data for offline use
- Javascript threads

- Data stored on client

- Offline Web Application
- Web Storage
- Web Worker

domainA.csnc.ch
domainB.csnc.ch
domainC.csnc.ch
Geolocation API

Finding your location: found you!

User Tracking

- User tracking based on the location of a user.
- If users are registered, their physical movement profile could be tracked.
- The anonymity of users could be broken.
User-Training
Web Workers

- The Web Sockets API
- Custom scheme and content handlers
- Cross-Origin Resource Sharing
- Web Messaging
- IFrame Sandboxing
- Location information
- Data for offline use
- Javascript threads
- Data stored on client
- Geolocation API
- Offline Web Application
- Web Storage
Web Workers

Web Workers provide the possibility for JavaScript to run in the background

Prior to Web Workers using JavaScript for long processing jobs was not feasible because
  ✦ it is slower than native code and
  ✦ the browsers freezes till the processing is completed

Web Workers alone are not a security issue.

But they can be used indirectly for launching work intensive attacks without the user noticing it.
Worst Case Scenarios

**Web Workers** \(=\) **Feature!**

Cracking Hashes in JS Cloud (**DEMO**).

**Web Workers** + **Cross-Origin Resource Sharing** \(=\) **Bomb**

**Powerful DDoS attacks.**

**Web Workers** + **Cross-Origin Resource Sharing** + **Web Sockets** \(=\) **Bomb**

**Web-based Botnet.**
... and much more ...
Some HTML5 features are the vulnerabilities themselves
XSS becomes even worse
Use IE6
DEMO – Exploiting Cross-Origin Resource Sharing

Shell of the Future
DEM0 – CORS – Shell of the Future

Simplified:

- Victim’s hijacked browser
- XSS
- Shell of the future console

Abused person

Attack target

Attacker controlled
DEMO – Exploiting Web Workers

Ravan
DEMO – Web Workers – Ravan

http://www.andlabs.org/tools/ravan.html
ECBT
Erster Compass Beer Talk
Questions

HTML5 is being developed as the next major revision of HTML. This code can now be used for new functions that can benefit developers and Internet users.

Here are the most important of them:

1. The **canvas element** can be used for rendering graphs, game graphics or other visual images on the fly. All done without having to rely on plug-ins. The possibilities are endless.

2. Embedding video used to be impossible without third-party plug-ins such as Apple QuickTime® or Adobe Flash®. Thanks to the **video element**, now it’s possible. It is intended by its creators to become the new standard way to show video online.

3. Sniffing a user’s location is not a new thing on the web. In fact, most websites already do this by means of IP address detection. But this is not always reliable, so HTML5’s **geolocation** is an alternate method of correctly pinpointing a user’s location. The new idea is to get the location information from WiFi towers and GPS.

4. The **offline web applications** enable users to continue interacting with web applications and documents even when their network connection is unavailable. The user can, for instance, access email locally without having to connect to the internet or install an external client.

How does this matter to you? You will notice that daily web activities such as uploading YouTube videos to your blog and finding a specific store in your browser on your smartphone will become easier. This means you can have a rich experience on a light, portable, universal platform.
References