

HTML5 Web Security

Thomas Röthlisberger – IT Security Analyst
thomas.roethlisberger@csnc.ch

Compass Security AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

Introduction to HTML5

Vulnerabilities & Threats

Countermeasures

Demo Web Workers

Demo CORS

Quiz and Q&A





Introduction to HTML5

Compass Security AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

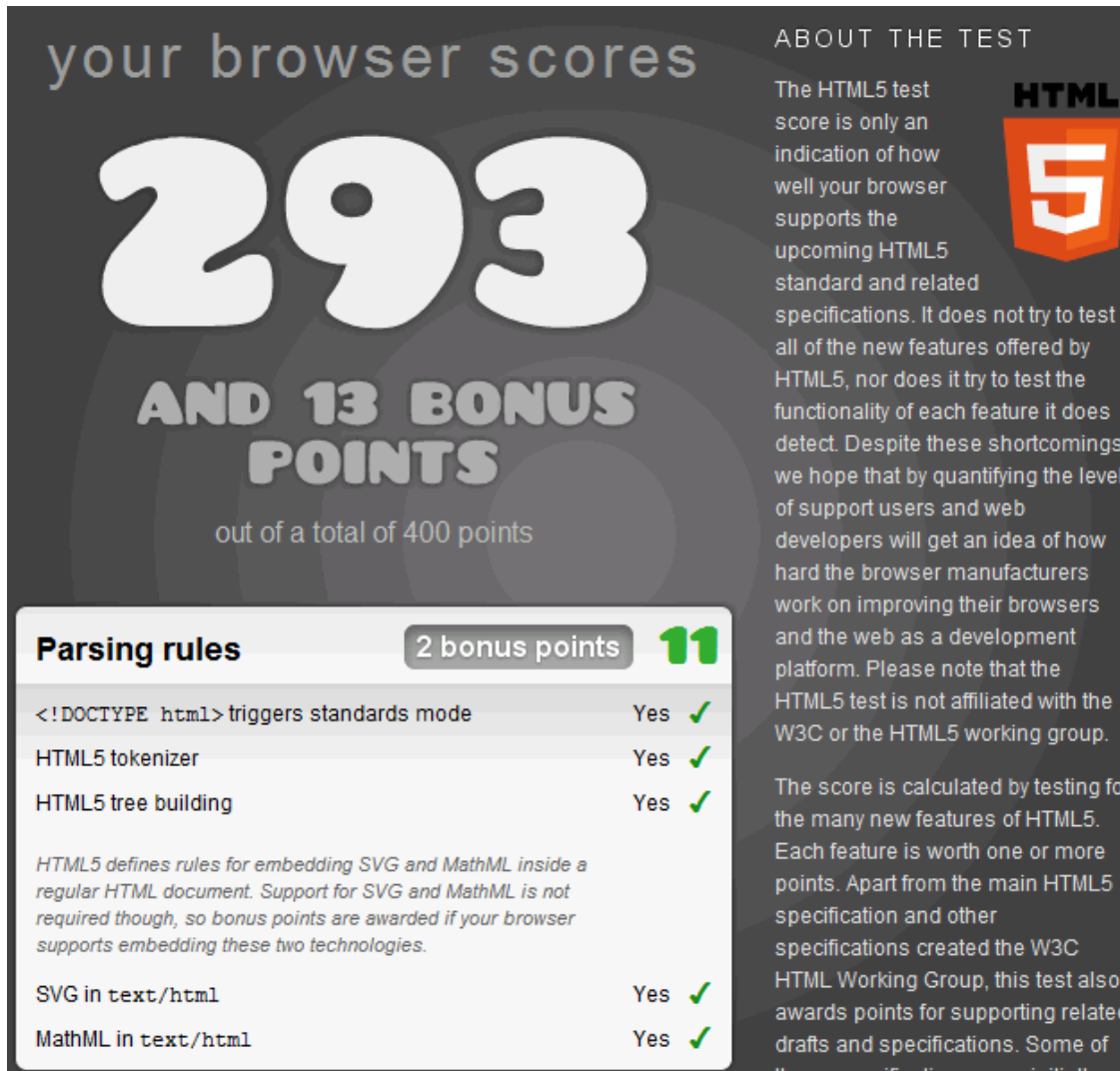
The Hypertext Markup Language version 5 (HTML5) is the successor of HTML 4.01, XHTML 1.0 and XHTML 1.1

Driven by the WHATWG and later also by the W3C

Current status is living standard (February 2011)

The candidate recommendation is planned for 2012 and the recommendation for 2022

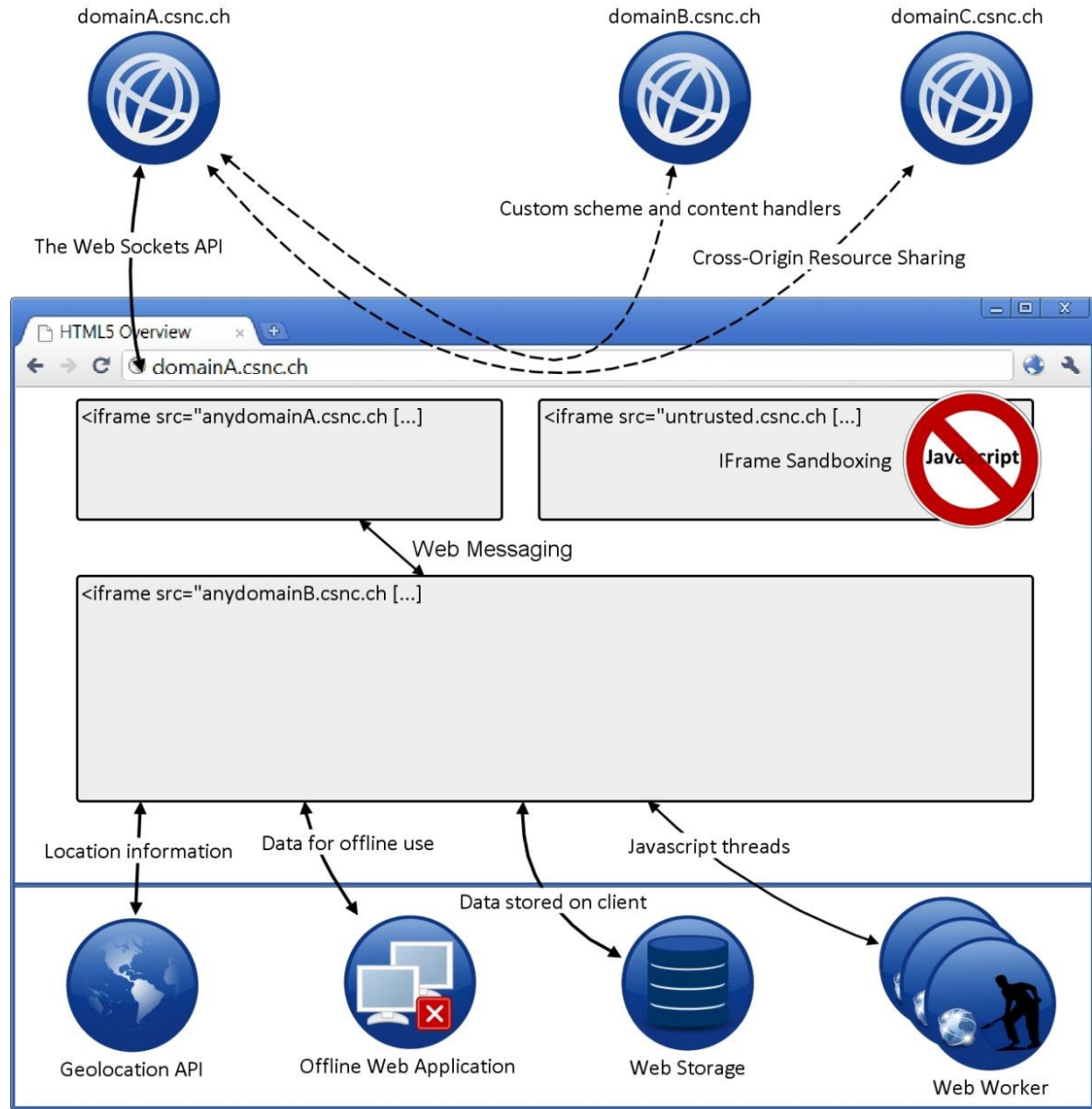
HTML5 is not finished

A screenshot of the HTML5 Test results page. The main heading is "your browser scores" in a light grey font. Below it, the score "293" is displayed in large, white, rounded numbers. Underneath the score, it says "AND 13 BONUS POINTS" in a smaller, white, rounded font, followed by "out of a total of 400 points" in a very light grey font. To the right of the score, there is a section titled "ABOUT THE TEST" in white. Below this title, there is a paragraph of text explaining the test's purpose and limitations. To the right of this text is the HTML5 logo, which is a white "5" inside an orange shield shape. Below the main heading and score, there is a table with a white background and a dark border. The table has a header row with "Parsing rules" on the left, "2 bonus points" in a grey box, and "11" in green. The table contains several rows of test results, each with a description, a "Yes" status, and a green checkmark. The first row is "<!DOCTYPE html> triggers standards mode". The second row is "HTML5 tokenizer". The third row is "HTML5 tree building". Below the table, there is a paragraph of text explaining that HTML5 defines rules for embedding SVG and MathML, and that bonus points are awarded for supporting these technologies. The table also includes rows for "SVG in text/html" and "MathML in text/html", both of which are marked as "Yes" with a green checkmark.

Parsing rules	2 bonus points	11
<!DOCTYPE html> triggers standards mode	Yes	✓
HTML5 tokenizer	Yes	✓
HTML5 tree building	Yes	✓
<i>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.</i>		
SVG in text/html	Yes	✓
MathML in text/html	Yes	✓

out of a
total of
400 points

Overview



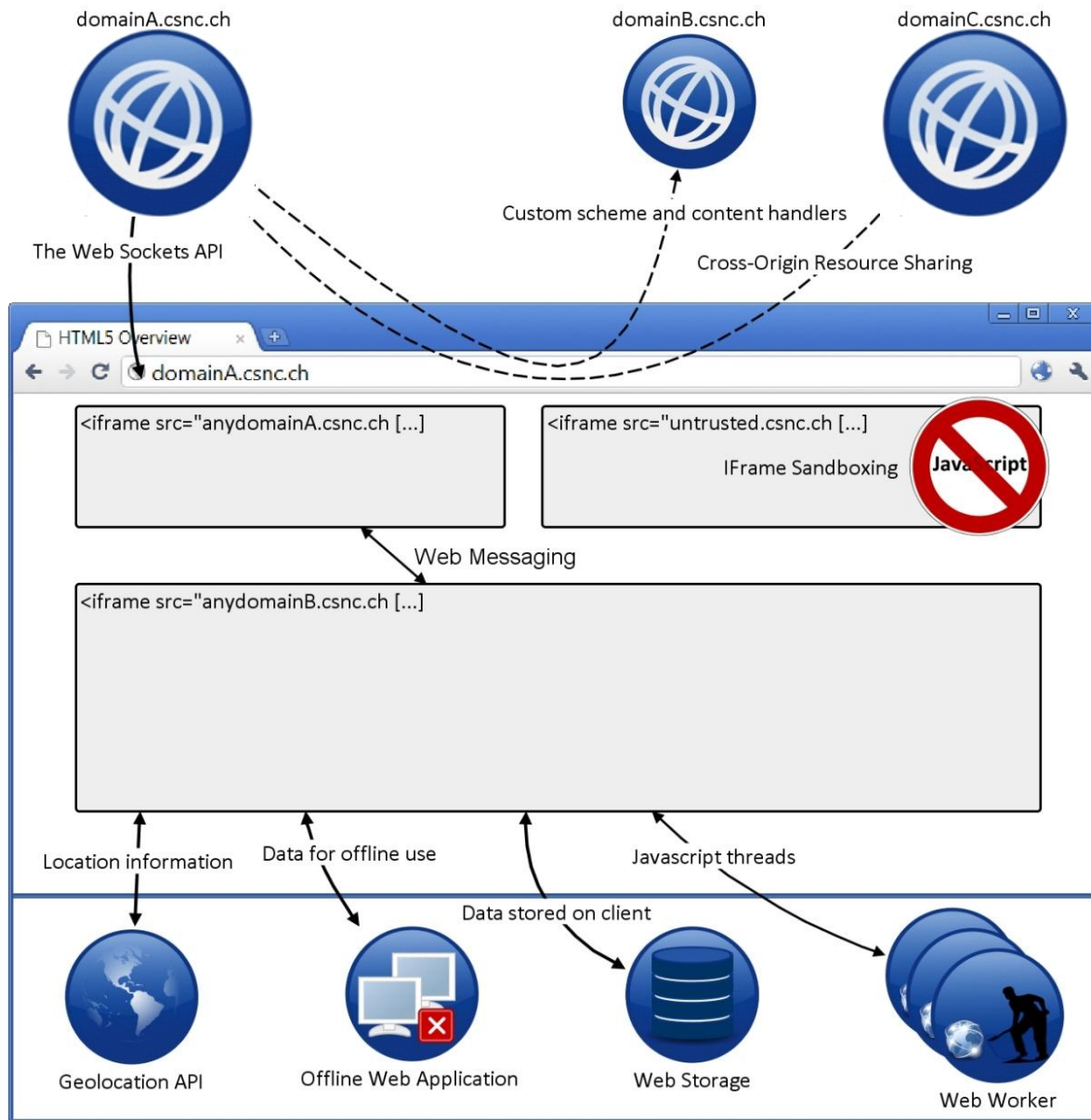
A vertical decorative strip on the left side of the slide features a close-up photograph of a computer keyboard with a yellow padlock resting on one of the keys. The background of the slide is white, with a horizontal dotted line near the top.

Vulnerabilities & Threats

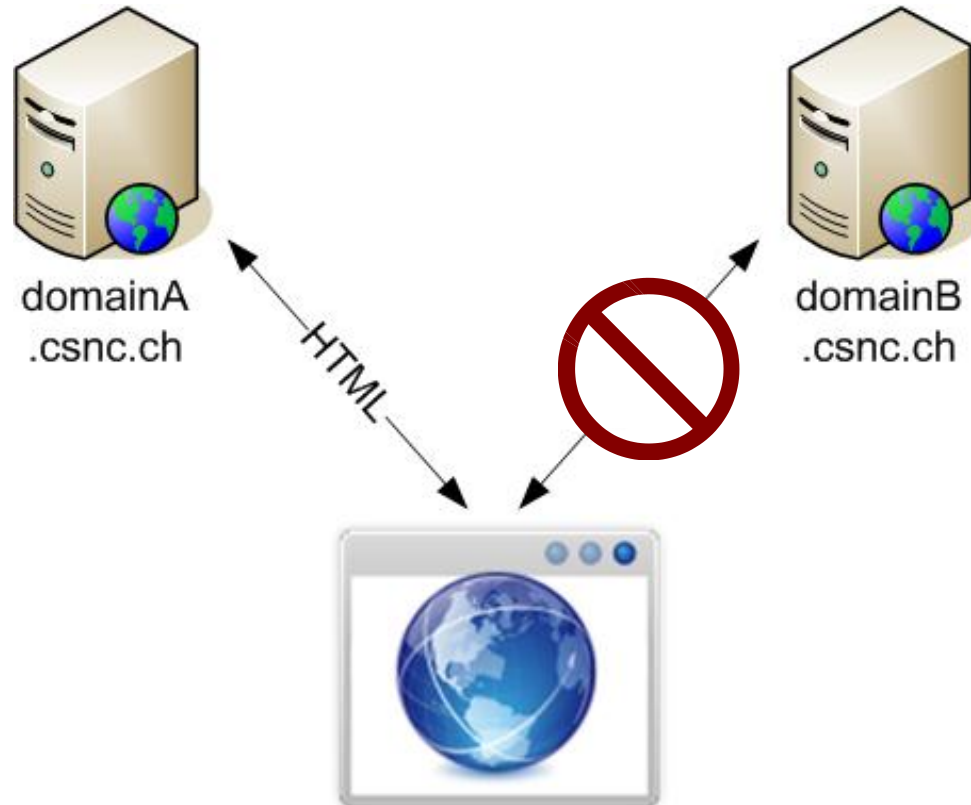
Compass Security AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

Cross-Origin Resource Sharing



Cross-Origin Resource Sharing I



Cross-Origin Resource Sharing II



GET / HTTP/1.1

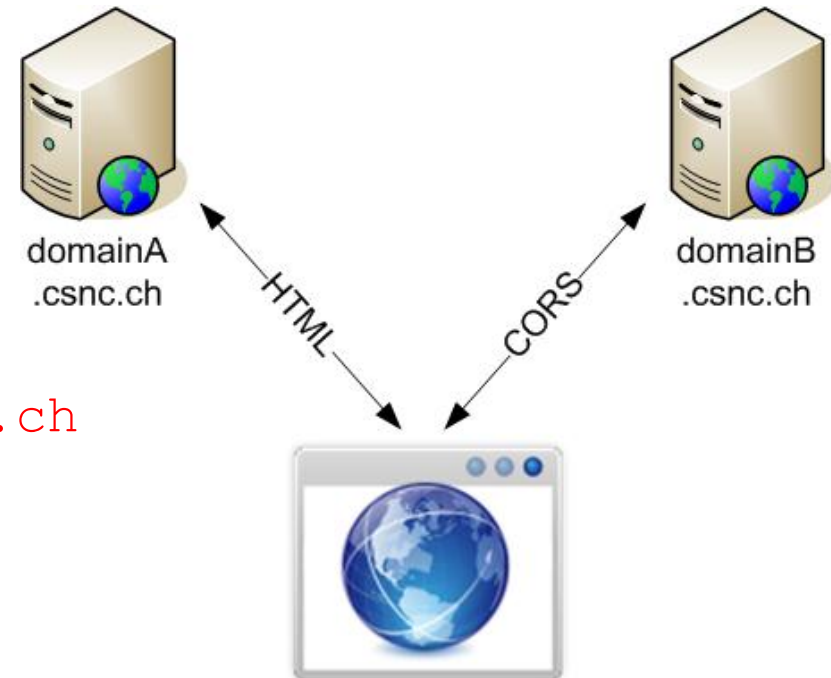
Host: **domainB**.csnc.ch

Origin: **http://domainA**.csnc.ch

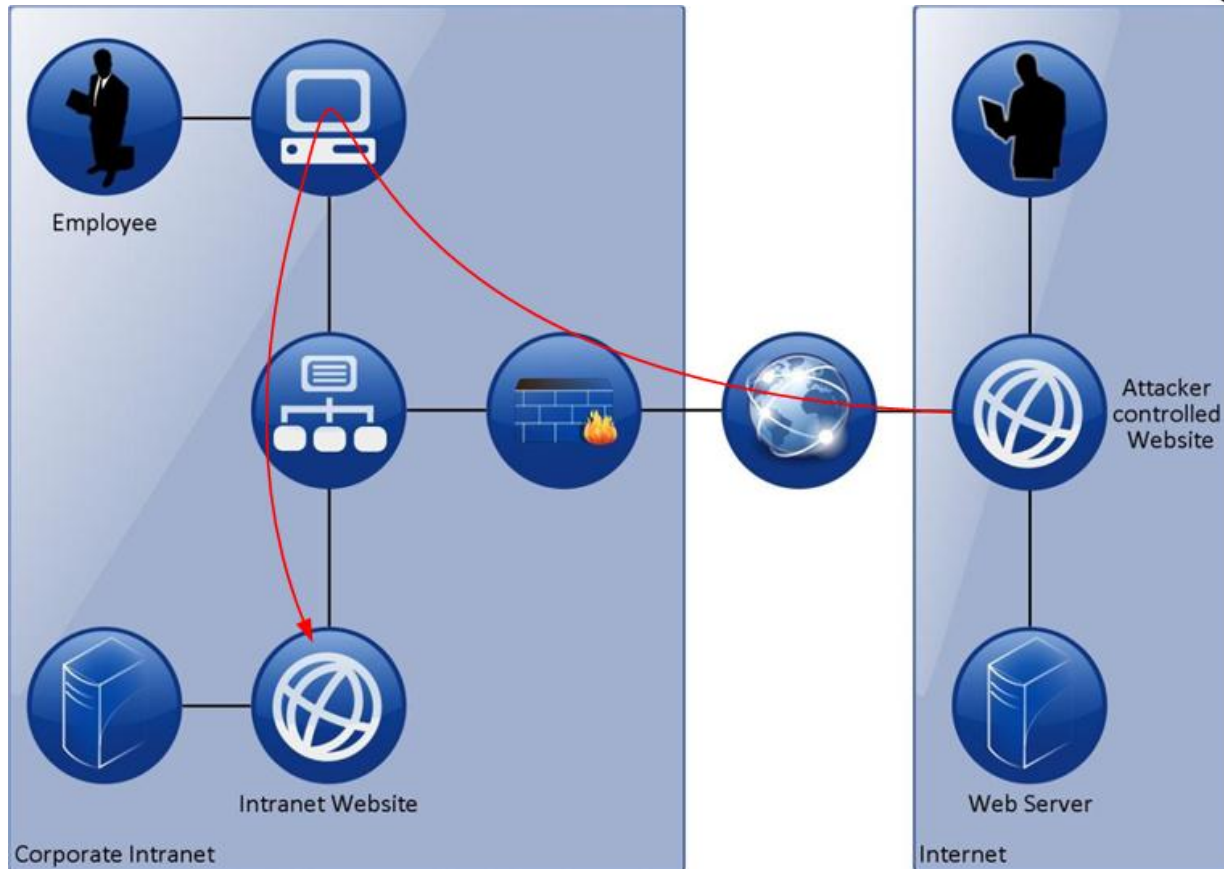
HTTP/1.1 200 OK

Content-Type: text/html

Access-Control-Allow-Origin: **http://domainA**.csnc.ch



CORS – Vulnerabilities & Threats I



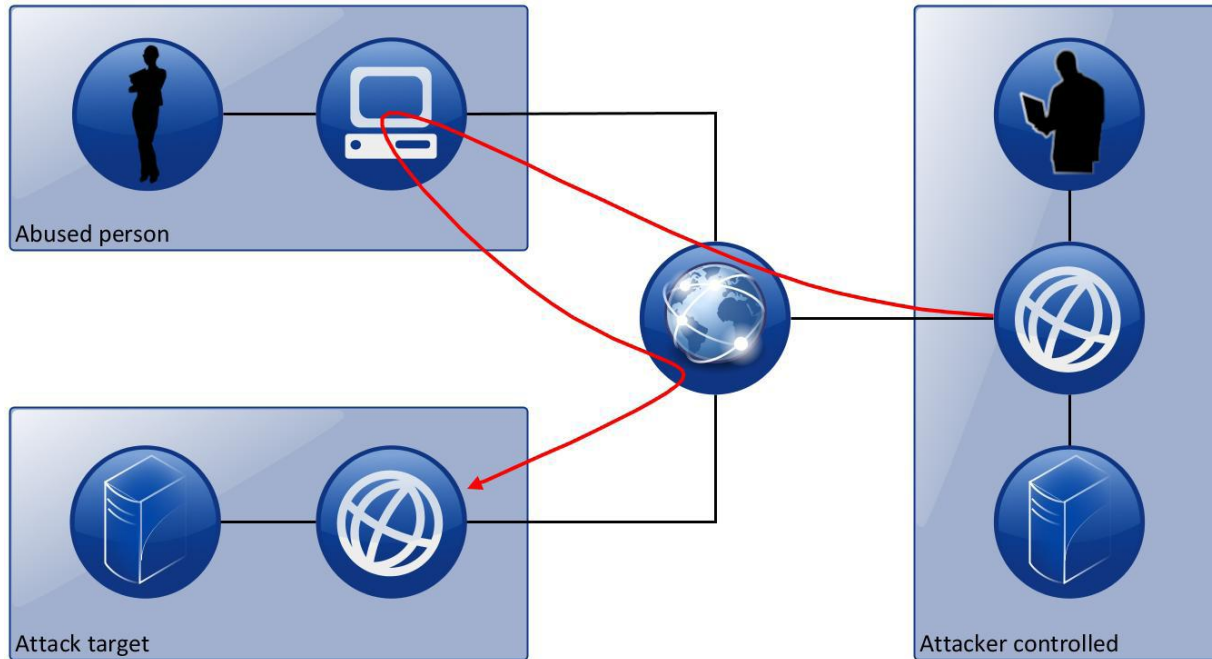
Accessing internal websites



Scanning the internal network



CORS – Vulnerabilities & Threats II



Remote attacking a web server



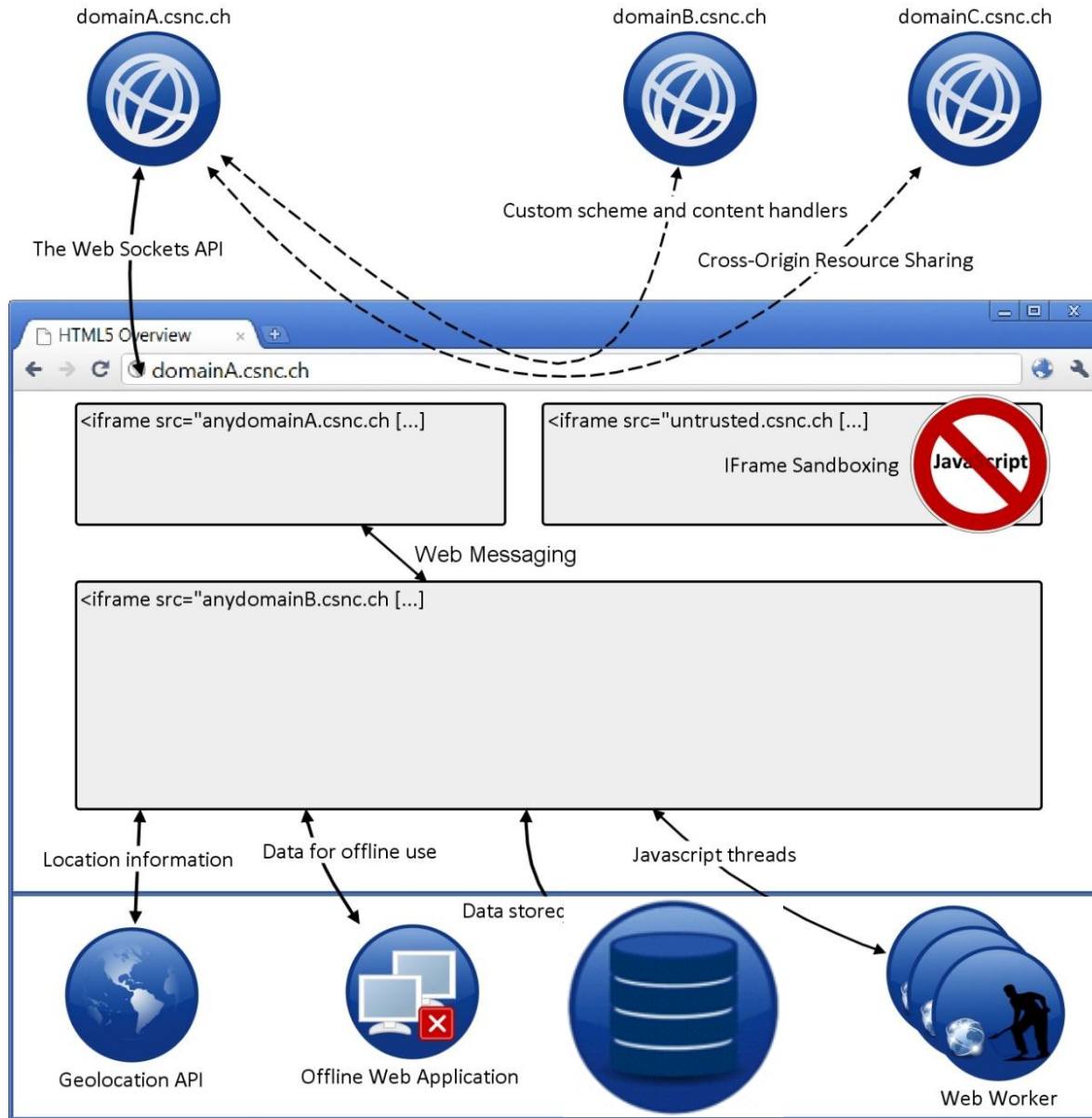
Easier exploiting of Cross-Site Request Forgery (XSRF)

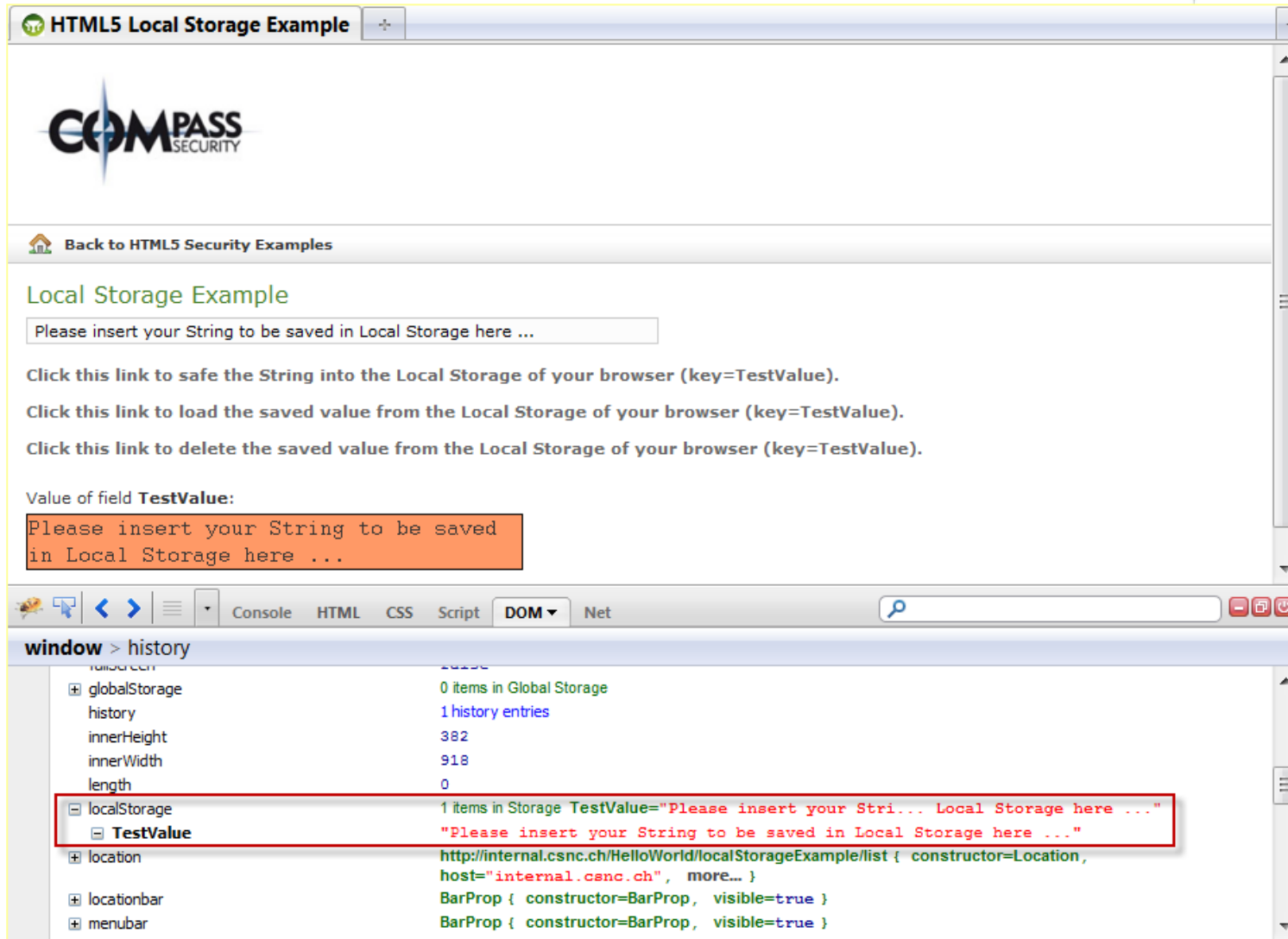


Establishing a remote shell (*DEMO*)



Web Storage





The screenshot shows a web browser window titled "HTML5 Local Storage Example". The page content includes the COMPASS SECURITY logo, a navigation link "Back to HTML5 Security Examples", and a section titled "Local Storage Example". This section contains a text input field with the placeholder text "Please insert your String to be saved in Local Storage here ...". Below the input field are three instructions: "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).", and "Click this link to delete the saved value from the Local Storage of your browser (key=TestValue).". A label "Value of field TestValue:" is followed by a text area containing the same placeholder text, which is highlighted with an orange background.

The browser's developer console is open, showing the "DOM" tab. The "window" object is expanded to show the "localStorage" property, which contains one item with the key "TestValue". The value of this item is "Please insert your String to be saved in Local Storage here ...". The console also shows other properties like "globalStorage", "history", "innerHeight", "innerWidth", "length", "location", "locationbar", and "menubar".



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.

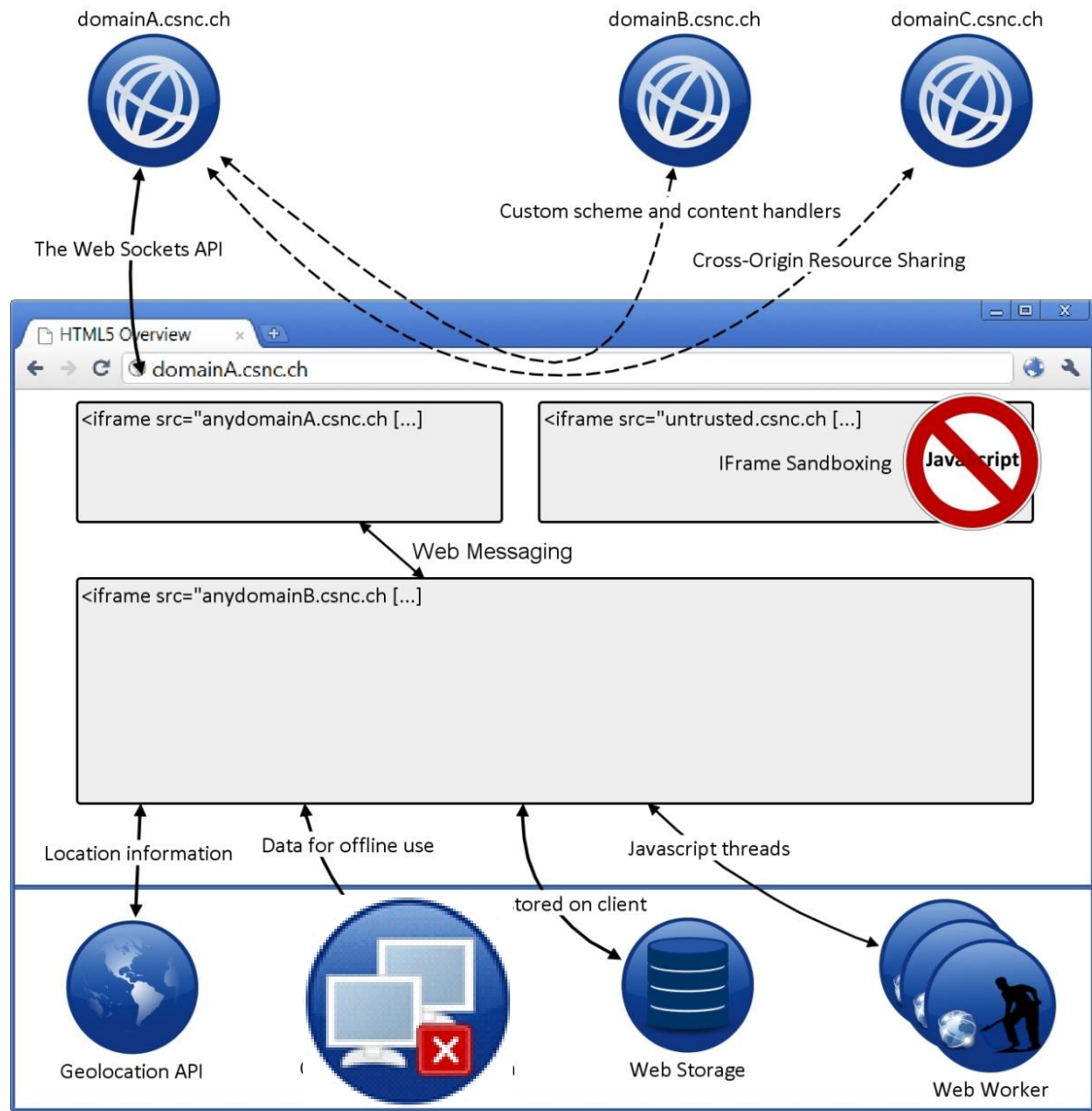
Persistent attack vectors



- ◆ Attack vectors can be stored persistently in the victim's browser.



Offline Web Application



```
<!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
```



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.

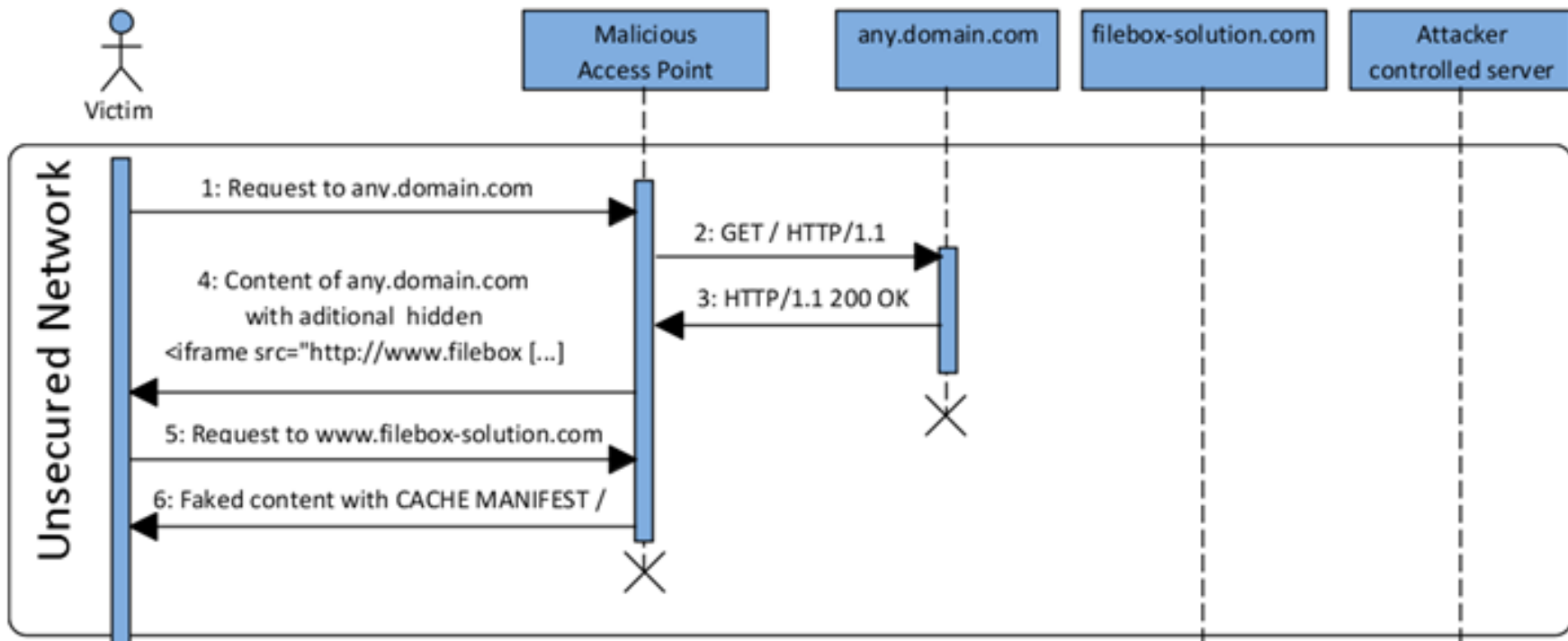
User Tracking

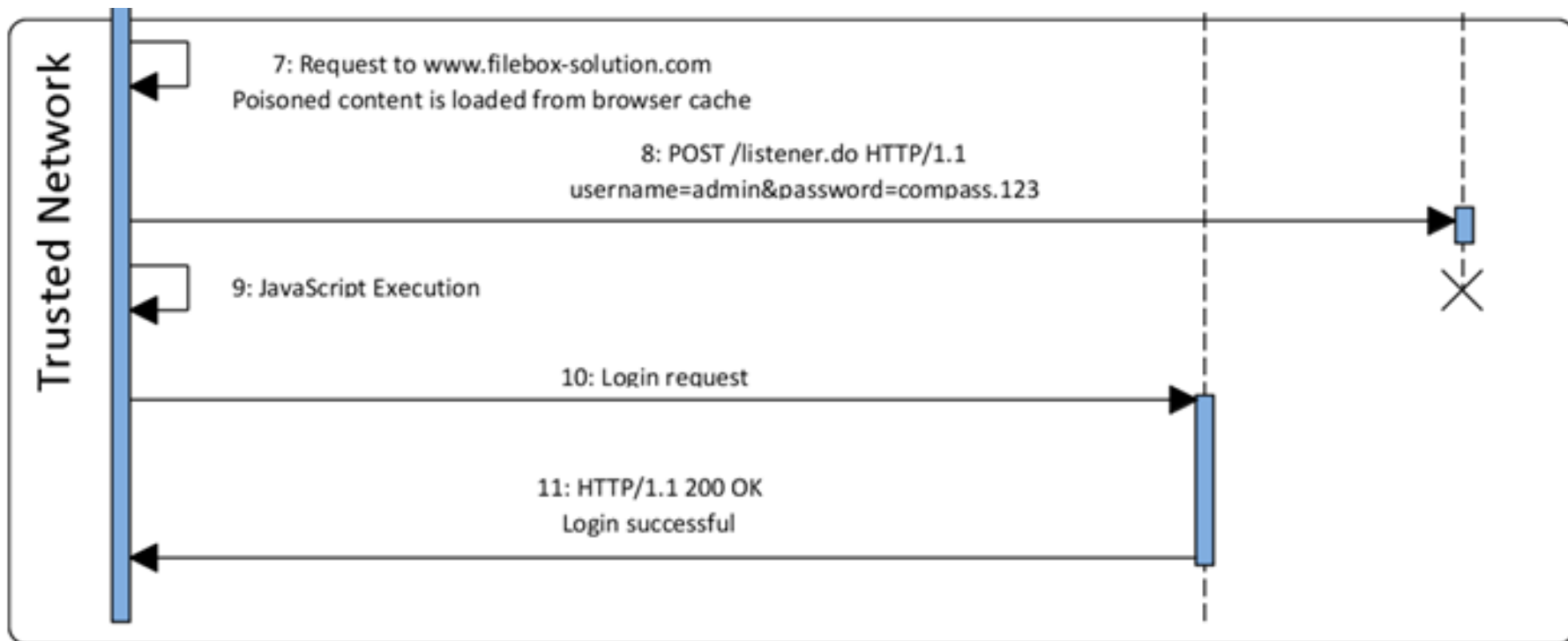


- ◆ Additional possibility to identify a user.
- ◆ Unique identifiers could be stored along with the cached files.

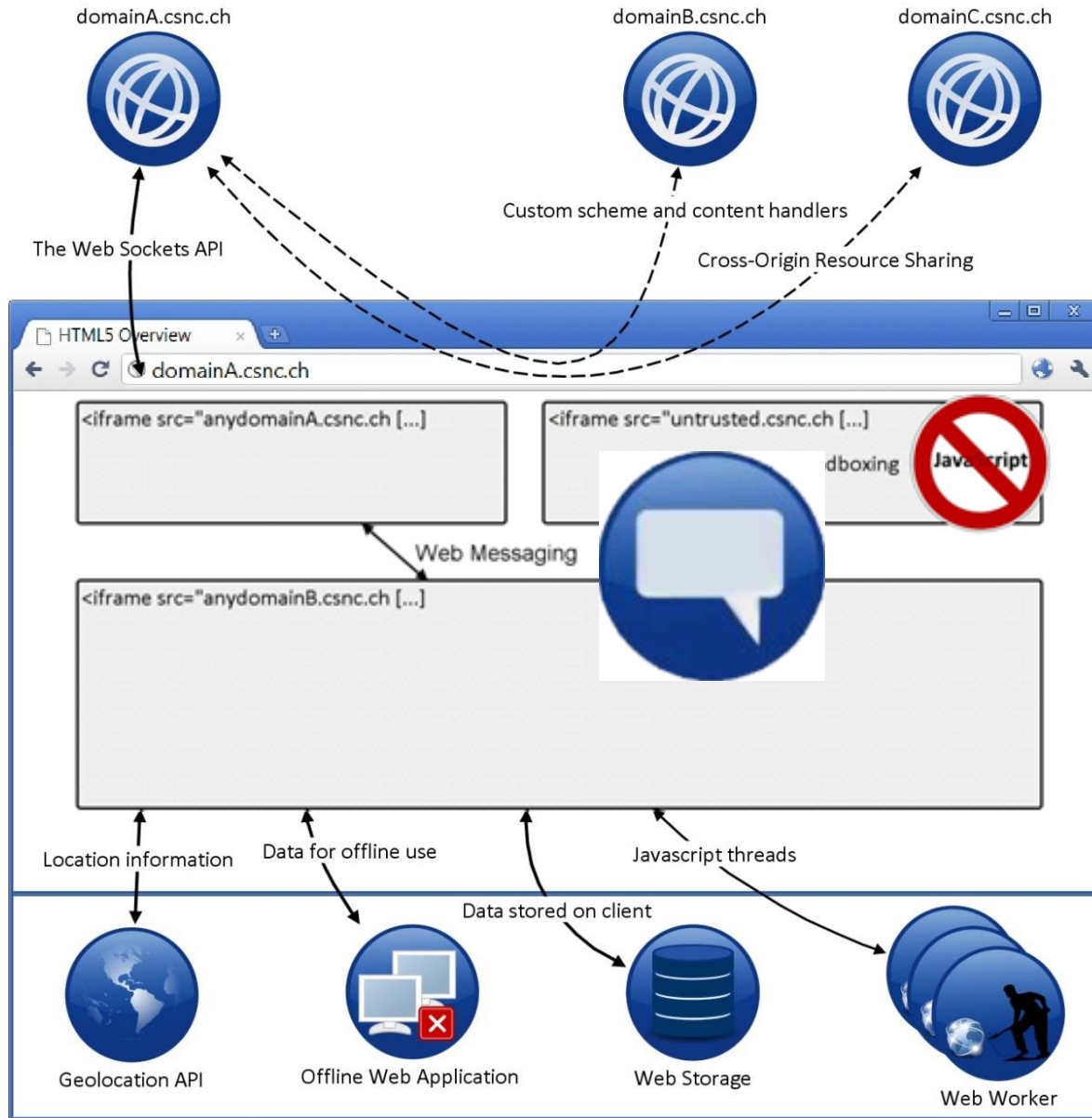


Offline Web Application – Attack 1/2





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.

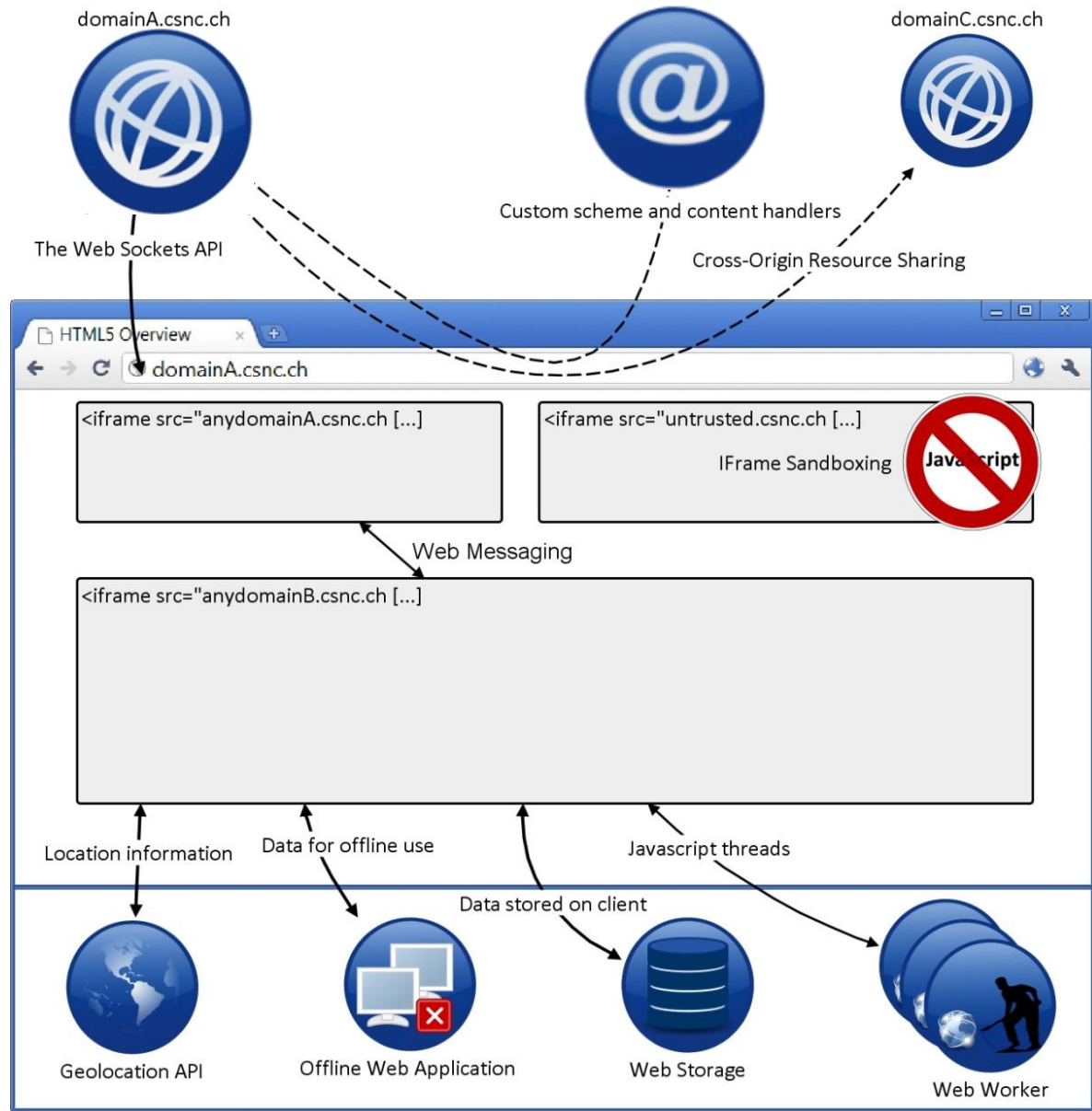
Expands attack surface to the client

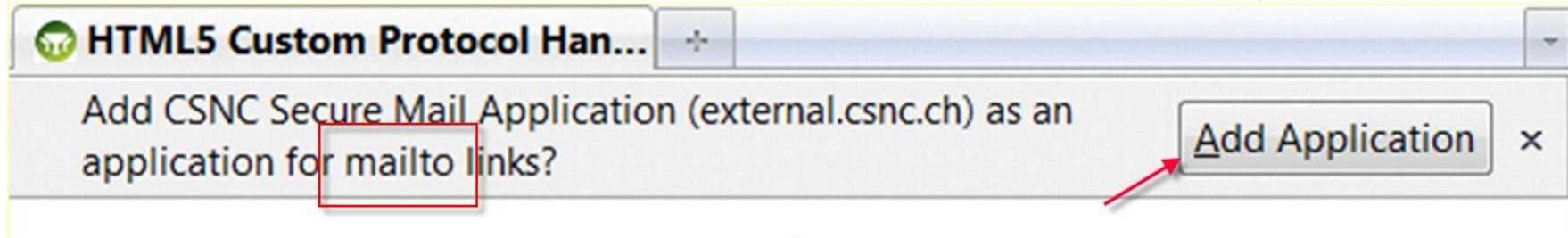


- ◆ Iframes can send malicious content to other Iframes.
- ◆ Input validation on the server is not longer sufficient.



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.

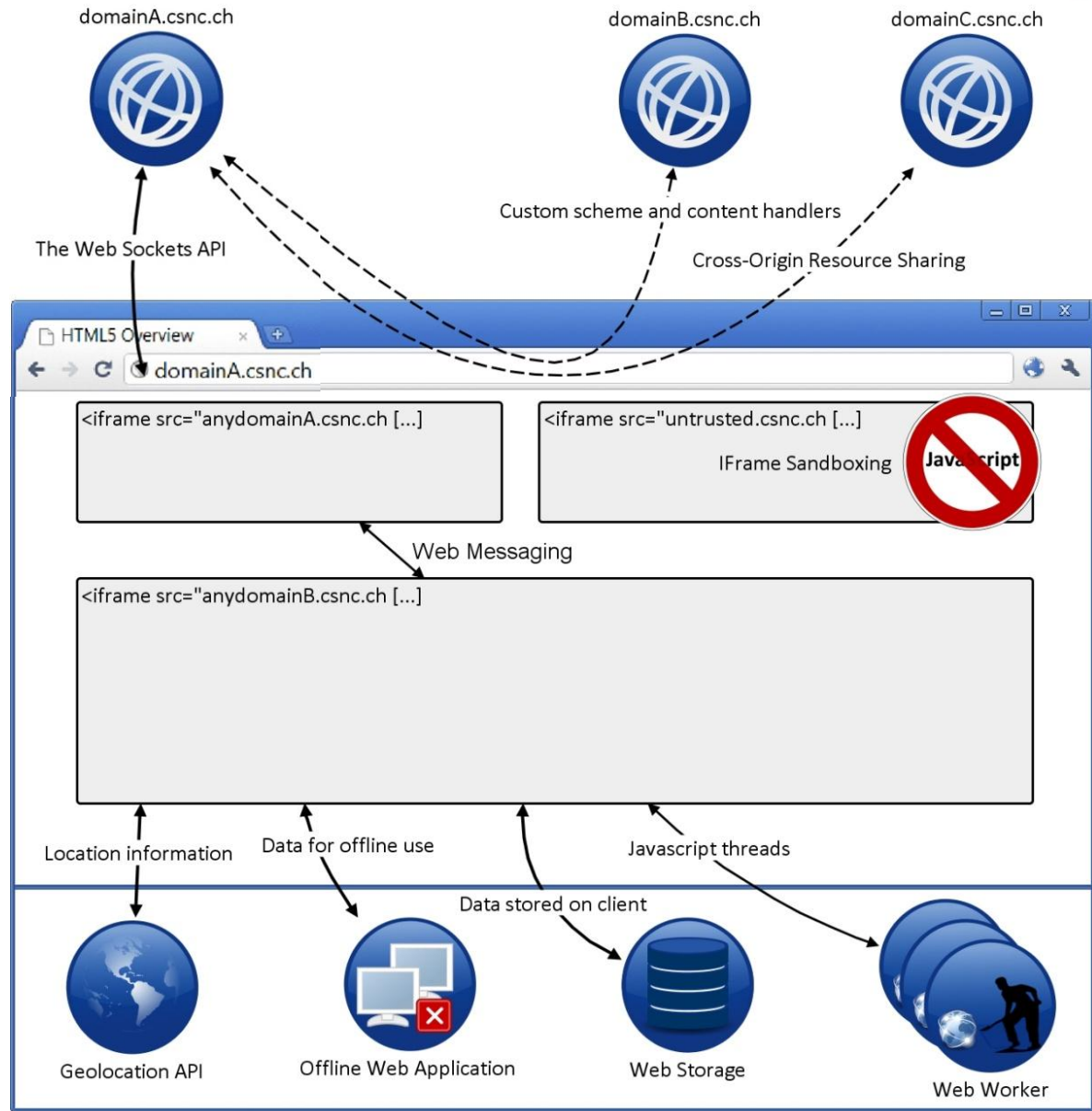
User Tracking



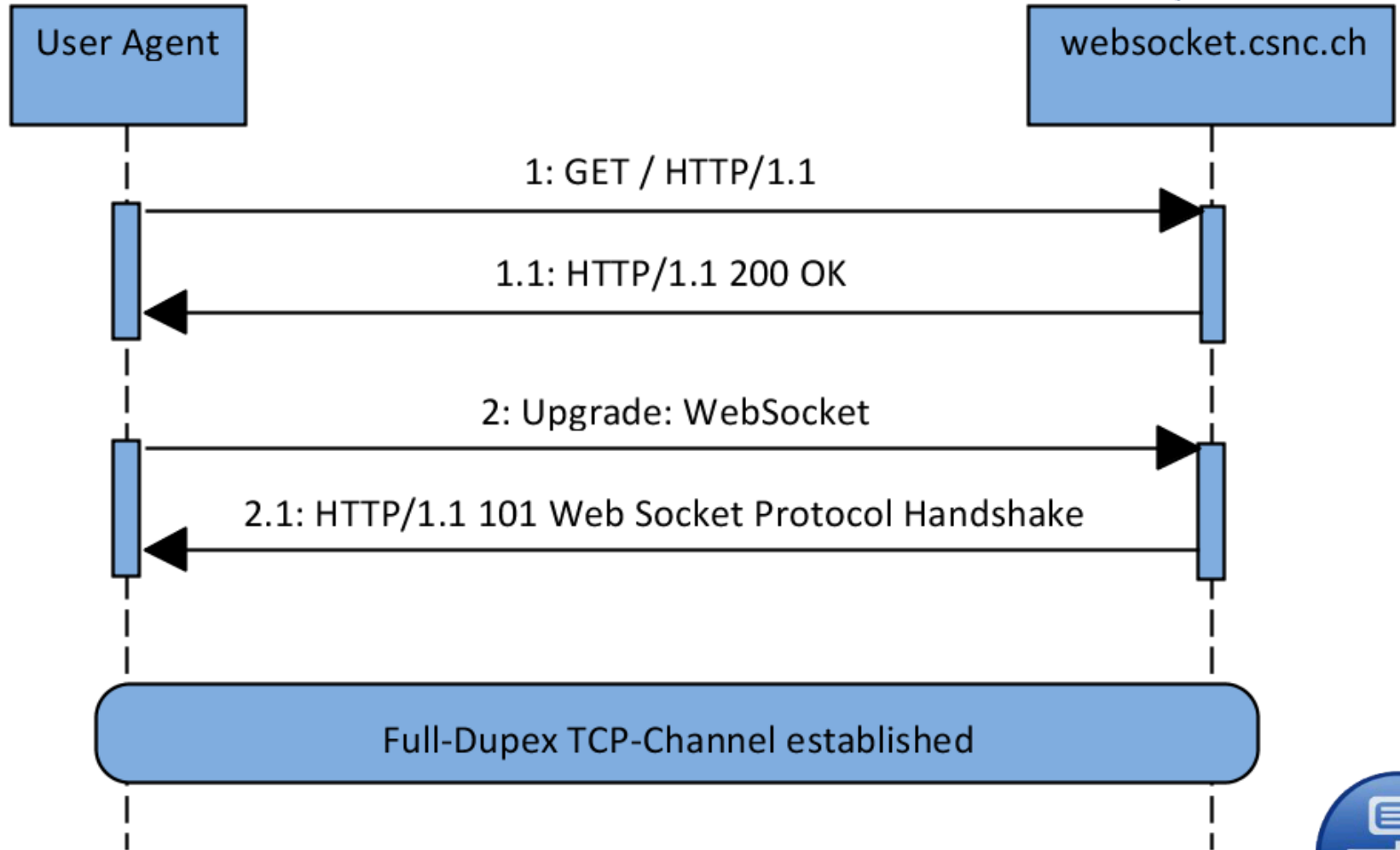
- ◆ Additional possibility to identify a user.
- ◆ Unique identifiers could be stored along with the protocol handler.



Web Sockets API



Web Sockets API



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.

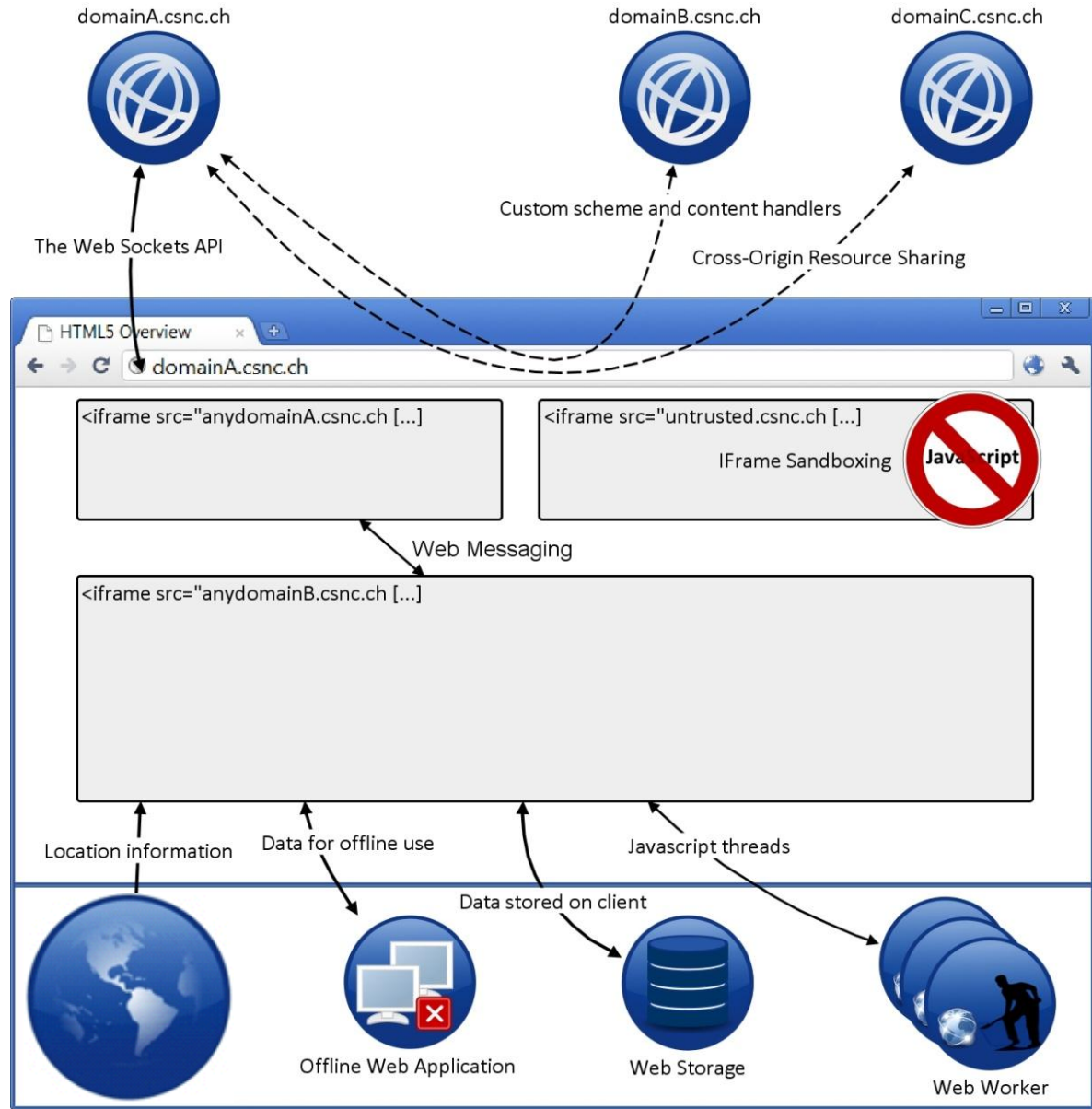
Establishing a remote shell



- ✦ Web Sockets can be used to establish a remote shell to a victim's browser.



Geolocation API



Geolocation API



Finding your location: **found you!**



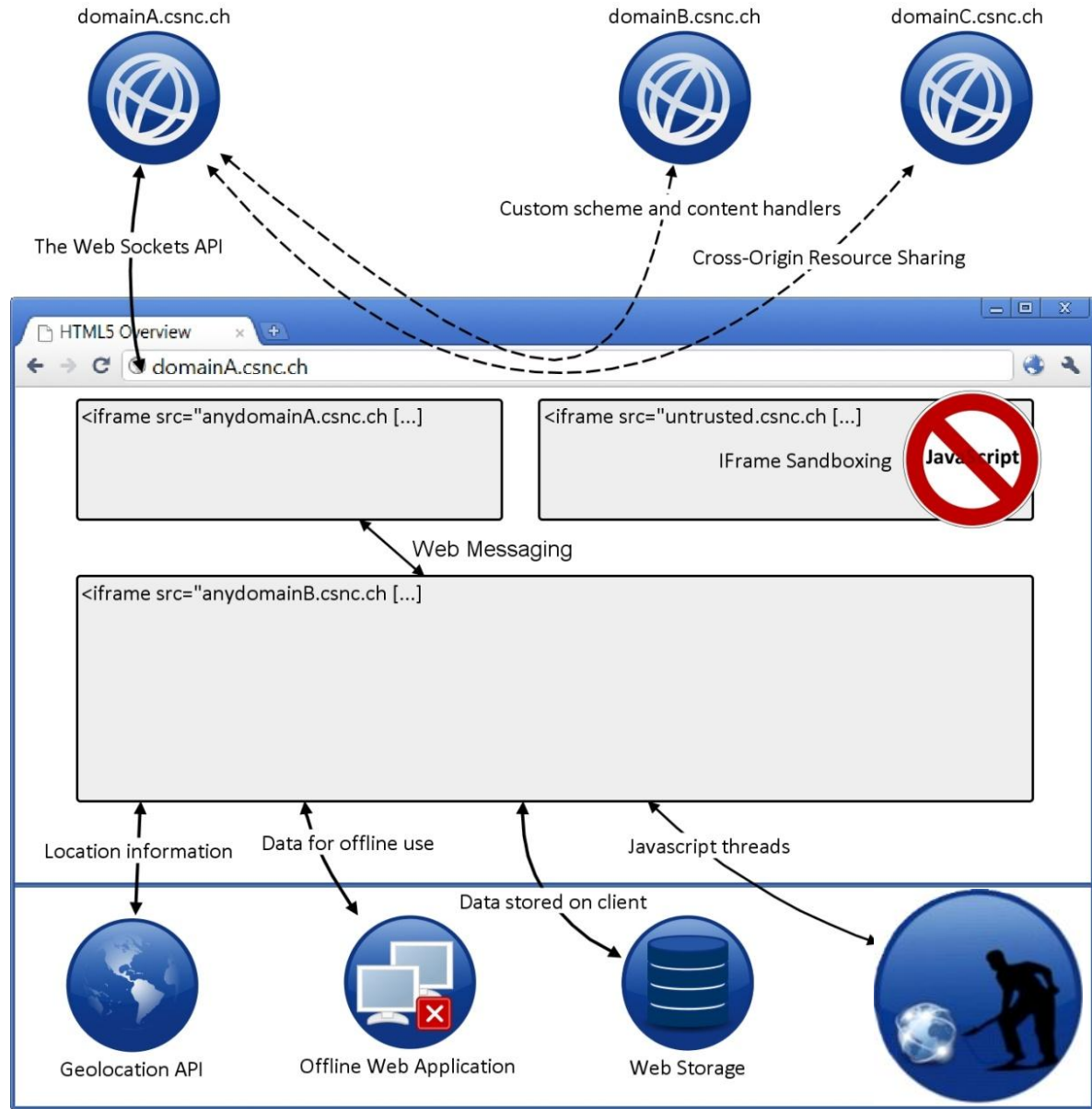
User Tracking



- ◆ 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.



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.

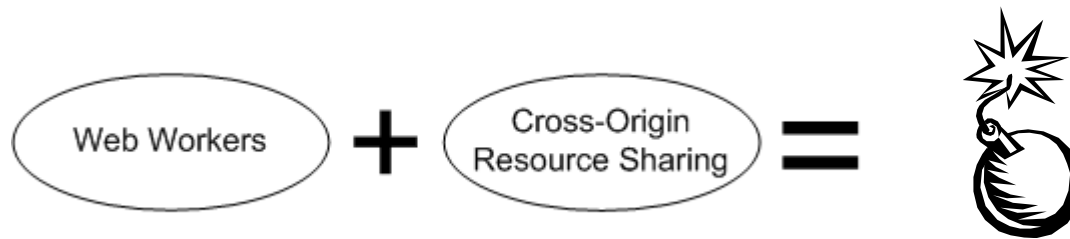


Worst Case Scenarios

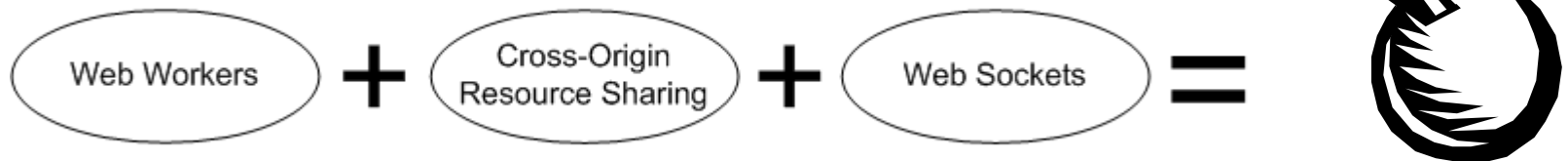


Web Workers = *Feature!*

Cracking Hashes in JS Cloud (*DEMO*).



Powerful DDoS attacks.



Web-based Botnet.



Countermeasures

Compass Security AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

Not all features can be mitigated through secure server-side implementation.

Cross-Site Scripting (XSS) becomes even worse.

Same old Story:

Do input validation and consequent output encoding.

Cross-Origin Resource Sharing

- ✦ 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

- ✦ Use cookies instead of Local Storage for session handling.
- ✦ Do not store sensitive data in Local Storage.

Web Messaging

- ✦ 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.

The risks of the following features cannot be mitigated by server side implementation or configuration:

- ✦ Custom scheme and content handlers
- ✦ Geolocation API
- ✦ Offline Web Applications
- ✦ Web Sockets

Therefore the users need to be trained:

- ✦ Do not accept registration of protocol handlers.
- ✦ Do not accept to share location information.
- ✦ Do not accept caching of web applications.
- ✦ Clear the cache including Local Storage and Offline Web Applications.

The risk of the Web Sockets API needs to be accepted.

- ✦ The only way to avoid Web Sockets would be to disable it in the browser.

DEMO – Exploiting Web Workers

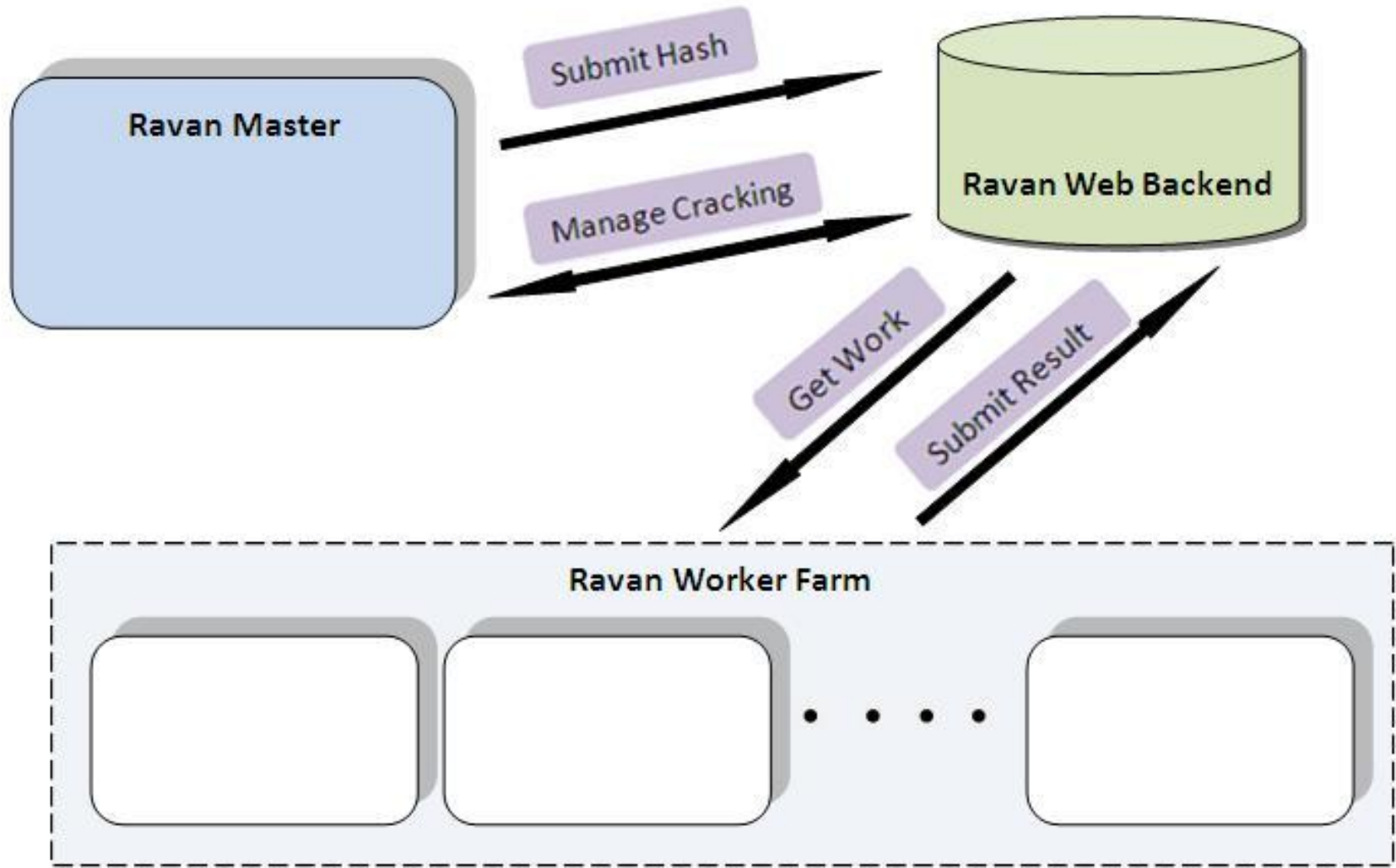
Ravan

Compass Security AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

DEMO – Web Workers – Ravan

<http://www.andlabs.org/tools/ravan.html>



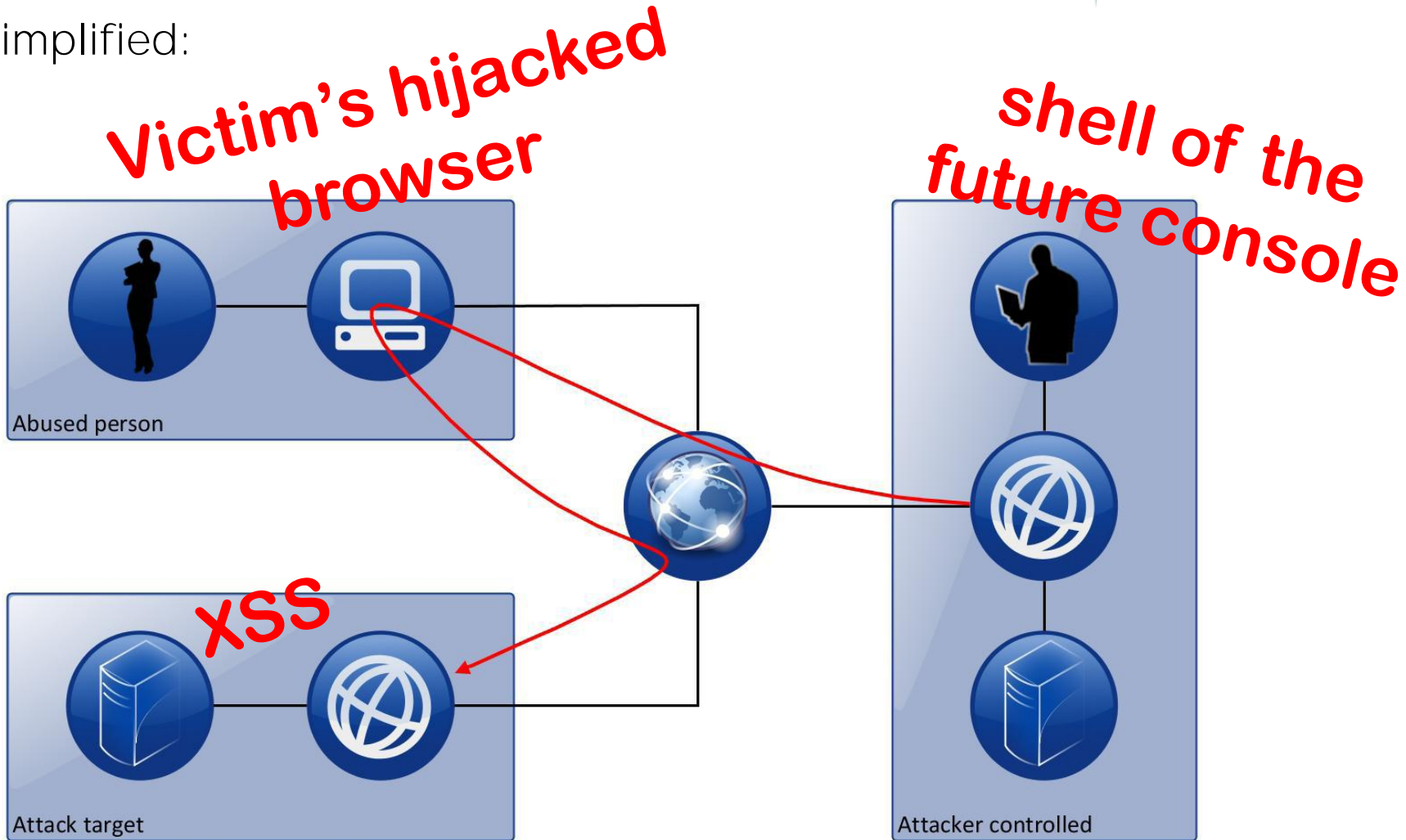
DEMO – Exploiting CORS

Shell of the Future

Compass Security AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

Simplified:





- ✦ Michael Schmidt, master thesis „HTML 5 Web Security“, 31st March 2011
- ✦ Lavakumar Kuppan, Attack and Defense Labs, <http://www.andlabs.org>
- ✦ W3C, HTML5, A vocabulary and associated APIs for HTML and XHTML, <http://dev.w3.org/html5/spec/Overview.html>