CDF: Predictably Secure Web Documents

Peter Snyder*, Laura Watiker†, Cynthia Taylor*, Chris Kanich*

* University of Illinois at Chicago
† Oberlin College
Overview

- The web is great! But complex!

- Complexity makes reasoning about privacy and security difficult for consumers

- Consider giving advice to non technical users

- **Knowing what we know now:**
  Is there a way to improve web security and privacy, without preventing authors from creating the types of sites users want?
The Web Today

- Interactivity is delivered as (mostly) unrestricted JavaScript
- Difficult to know code will be **benign and “useful”**:  
  - form validation  
  - improve user experience  
  - drive user-serving widgets and page elements  
- Or **malicious**:  
  - fingerprint the user  
  - exploit a vulnerability  
  - from untrusted source (XSS)
## Complexity vs. Benefit

<table>
<thead>
<tr>
<th>Web API Standard</th>
<th># Sites Uses</th>
<th>% Blocked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gamepad</td>
<td>3</td>
<td>0.0%</td>
</tr>
<tr>
<td>Performance Timeline, Lv. 2</td>
<td>1,728</td>
<td>93.7%</td>
</tr>
<tr>
<td>WebRTC 1.0</td>
<td>28</td>
<td>29.2%</td>
</tr>
<tr>
<td>XMLHttpRequest</td>
<td>7,957</td>
<td>13.9%</td>
</tr>
</tbody>
</table>
Complexity vs. Benefit

Figure 6: Popularity of standards versus their block rate, on a log scale.

Figure 7: Comparison of block rates of standards using advertising vs. tracking blocking extensions.

5.7 Site Complexity

We also evaluated sites based on their complexity. We define complexity as the number of standards used on a given website. As Figure 8 shows, most sites use many standards: between 14 and 32 of the 74 available in the browser. No site used more than 41 standards, and a second mode exists around the zero mark, showing that a small but measurable number of sites use little to no JavaScript at all.
## Goals

**Keep**
- HTTP(S)
- Decentralized / Rapid Deployment
- Interactivity
- Styling / Presentation
- Web Browsers

**Gain**
- Predictability
- Security
- Privacy
- Removing arbitrary code execution
Approach: Contained Document Format

1. **Document Format**:
   - JSON format, simple to check
   - Structure (like HTML)
   - Declarations of interactivity (vs. implementation)

2. **Client Proxy**: Translates CDF -> HTML+JS

3. **Trusted Libraries**: Implement safe interactivity
CDF Documents

- **Structure:**
  - Comparable to HTML tags
  - Forces separation of structure and text

- **Events:**
  - Designate *when* something should happen
  - Taken from common DOM and framework provided events

- **Behaviors:**
  - Designate *what* happens when an event triggers
  - Static definition, safely converted into JavaScript by TCB
  - Selected from common web idioms (element manipulation, timers, tabs, network communication, etc)
Parser Example

```javascript
let buttons = document.getElementsByTagName("button");
let stateIndex = 0;
let textStates = ["click on", "click off"];
builtins[0].addEventListener("click", function (event) {
    let newTextIndex = stateIndex++ % textStates.length;
    let newText = textStates[newTextIndex];
    event.target.innerHTML = newText;
});
```
CDF Flow

1. Client Request

2. CDF File

3. CDF → HTML+JS

4. HTML+JS

5. Trusted JS

6. “Safe” Assets
Advantages

• **Limited Trusted Base**
  No plugins, restricted Web API use

• **Client Side Fingerprinting**
  No JS means no JS based approaches (font / plugin enumeration, canvas fingerprinting, etc.)

• **Predictable Information Flow**
  No iframes, no HTTP referrers, restrictions on forms, "tracking speed bump"

• **Page Defacement / XSS**
  Typing in CDF documents, no script injection
Usability Tests

- **Popular blog:**

- **Online-banking:**
  [https://www.bankofamerica.com/](https://www.bankofamerica.com/)

- **Social media:**
  [https://twitter.com/](https://twitter.com/)

- **Collaborative web application:**
  HotCRP
Conclusion

• Modern web provides web authors great flexibility

• This flexibility makes it difficult for consumers to reason about security and privacy online

• With (relatively) small changes, the web could provide more predictable privacy and security, without sacrificing expressivity.

• CDF is a design experiment to explore different privacy / capability tradeoffs.

• Source: https://github.com/bitslab/cdf

• Thank you!