HTML 5
The good, the bad, the ugly
A presentation by Mario Heiderich, 2010
Introduction

- Mario Heiderich
  - Based in Cologne
  - Grad. Eng. and PhD student
  - Freelancer and researcher at the RUB
  - JavaScript, markup and malware research
  - International speaker
  - HTML5 Security Cheatsheet [heideri.ch/jso]
  - PHPIDS [php-ids.org]
  - @0x6D6172696F [twitter.com]
This Presentation

- HTML5 and DOM Level 3
- Overview and History
- HTML5 and current browsers
- The good
- The bad
- The ugly
- Discussion
„It must be admitted that many aspects of HTML appear at first glance to be nonsensical and inconsistent.“

[w3.org/TR/html5/introduction.html#introduction]
History lesson

- **1990 – 1995** :: first revisions derived from SGML and hosted by CERN and IETF
- **1995** :: W3C took over releasing HTML3.0
- **1997** :: HTML3.2 - many new extensions
- **1998** :: HTML4 - still used today - and DOM Level 1
- **2000** :: DOM Level 2 Core
- **2003** :: DOM Level 2 HTML, XForms
- **2004** :: The idea for HTML5 was born - WHATWG founded
- **2006 - 2007** :: W3C redecided - particiapted in HTML5
- **2007 - now** :: WHATWG and W3C collaborate on HTML5
Current Status

- HTML5 is **not ready** yet – work in progress
- Same for vendor support
- No child of SGML anymore
- W3C != WHATWG, HTML5 != HTML5
- New DOM interfaces
- Webforms 2.0 enhanced
- XHTML5
In short words

- HTML5 brings us
  - A pile of new tags and structural elements
  - Many new attributes
  - Easier ways to create usable websites
  - Generally a lack of strictness
  - New form elements
  - New DOM interfaces and methods
  - And many more things – we cannot enumerate them all in one hour...
Browsers

- HTML5 develops
- So do the user agents. Or at least have to.
  - **Opera**: advantage through supporting a dead specification
  - **Chrome**: release often - release early
  - **IE9**: a lot of catching up to do
  - **Firefox**: finally a new parser - `html5.enable=true`
Consistency?

- Impossible!
- W3C HTML5 specs are 4.4 MB of text
- WHATWG HTML5 specs – 707 A4 pages
- This is a lot of implementation work
- Don't we forget about
  - CSS3,
  - ECMAScript 5
  - SVG
  - Canvas etc. etc.
So... security?

- Some say HTML5 itself is a vulnerability
- Not that funny - not that wrong
- Secure implementations require
  - Clear specifications ✗
  - Manageable amount or work ✗
  - Thorough and diverse testing ✗
  - Fast and precise feedback loops ✗
  - Quick and comprehensive patch deployment ✗
Results

- Inconsistent and ever-evolving specs
- Browsers rush for implementation
  - [html5test.com] and others
- Webdevs still build buggy websites
- Necessary legacy support
  - IE6 is still around … UK gov, PayPal, etc.

- But now for some actual goodies!
The Good

- New form elements and element types enhance usability
- `<input type="` – range, tel, color, datetime-local, email, url, ...
- New `<output>` tag
- Autofocus and active form elements
- Client-side validation and placeholders
- Form elements - outside the form
More good

- Iframe restrictions `<iframe sandbox />`
- Seemlessness for iframes
- Local storage mechanisms
- Client side databases
- Geolocation services
- Notification interfaces
- Interaction with USB and RS232 devices
- Multimedia and inline SVG
So good!

- Animations and transformations
- WebGL and 3D acceleration inside the browser
- Video and audio support
- New webfont technologies
- Less Flash and Silverlight – more open standards
- Accessibility and document structuring
Any bad?

- W3C and WHATWG mean HTML5 to be
  - An easy way to create interactive and rich content for everyone
  - Less XMLish strictness – more open structure and fun
  - Simplification instead of over-specification
  - The focus is neither the server nor the browser – but the user
  - HTML4 was screen, XHTML was open – HTML5 is web
Bad stuff please!

- Hijacking forms with the new form attribute #1
- Stealing personal data via autofill
- Stealing focus and keystrokes #8
- Dossing the client with bad validation regex #14
- Bypassing blacklists with new event handlers #23
- Using harmless attributes to execute JavaScript #10
- Disabling framebusters with sandboxes
- *Enough already? No?*
The Ugly

- Abusing the `history.pushState()` API
  - URL spoofing #103
  - Redirection to infected websites
  - Overflowing users history
- Abusing local storage on non domains
  - `about:blank` is not a domain - or is it?
  - Cross-medium attacks on Opera
  - Payload hiding and obfuscation
More Ugliness

- SQL injections on the client
  - `openDatabase()` uses SQLite
  - An 0-day in SQLite affects all user agents
  - SQL injections in the DOM
  - DOMSQLI superseding DOMXSS
- Circumventing protection mechanisms with sandboxed iframes
- Using evil SVG chameleons 🎨
Roundup

- HTML5 does ship awesomeness
- But it also is an actual vulnerability nest
- We *now* know why
- HTML4 was static - few new vulnerabilities for *years* (except for vendor specific extensions like HTML+TIME, Data Islands, HTA, HTC, ActiveX, -moz-binding, -o-link-source and many many more)
- HTM5 is dynamic - forcing vendors to progress
- That by design leverages insecurity
Discussion

- How to improve the situation?
- Where will we be in two years? Or five?
- How to make the average user understand risks in the www?
- And what will the average user be like?
- Will there be too many web developers in five years - or just mashup architects?
- Will we still have servers - or just CDNs?
Questions

- Please feel free to ask and comment!
- Or mail me later on mario.heiderich@gmail.com

- Thanks for your time!
Links

- [http://simon.html5.org/html5-elements](http://simon.html5.org/html5-elements) Overview on HTML tags and elements
- [http://www.w3.org/TR/html5/](http://www.w3.org/TR/html5/) The W3C spec draft
- [http://www.w3.org/TR/html5-diff/](http://www.w3.org/TR/html5-diff/) W3C differences between HTML4 and HTML5
- [http://code.google.com/p/html5security/w/list](http://code.google.com/p/html5security/w/list) Articles on HTML5 security
- [http://lists.w3.org/Archives/Public/public-web-security/](http://lists.w3.org/Archives/Public/public-web-security/) W3C HTML security mailing list