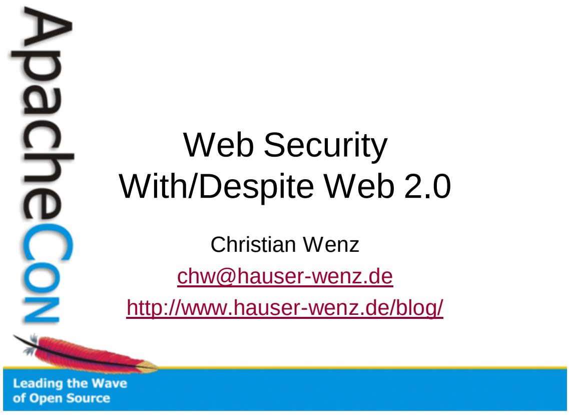
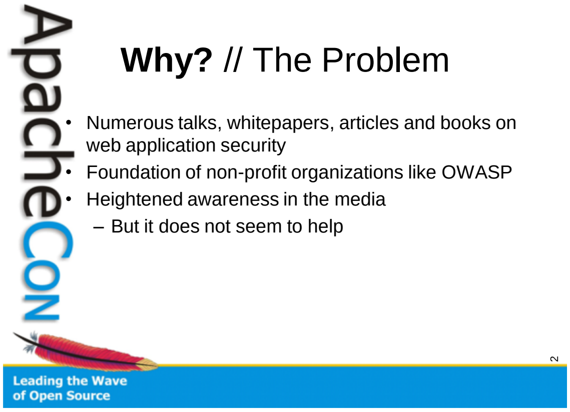


Web Security With/Despite Web 2.0

Christian Wenz
chw@hauser-wenz.de
http://www.hauser-wenz.de/blog/

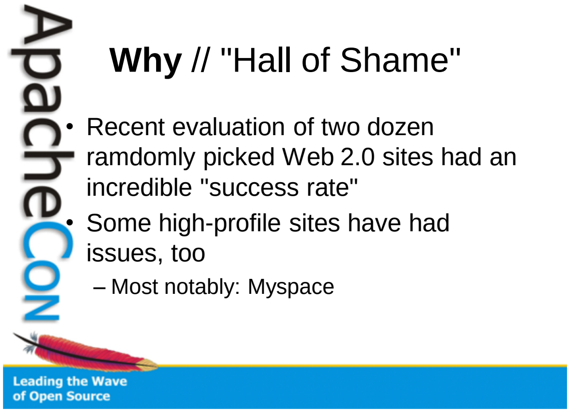
ApacheCon
Leading the Wave of Open Source



Why? // The Problem

- Numerous talks, whitepapers, articles and books on web application security
- Foundation of non-profit organizations like OWASP
- Heightened awareness in the media
  - But it does not seem to help

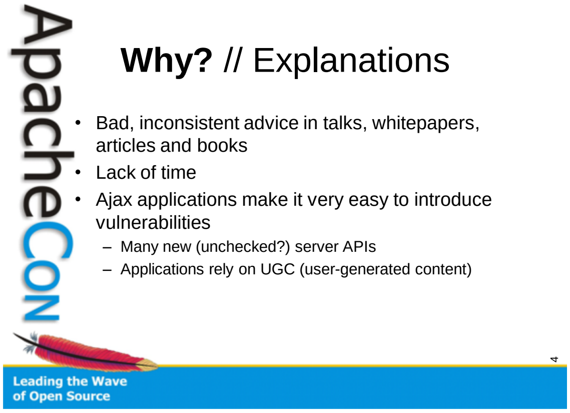
ApacheCon
Leading the Wave of Open Source



Why // "Hall of Shame"

- Recent evaluation of two dozen ramdomly picked Web 2.0 sites had an incredible "success rate"
- Some high-profile sites have had issues, too
  - Most notably: Myspace

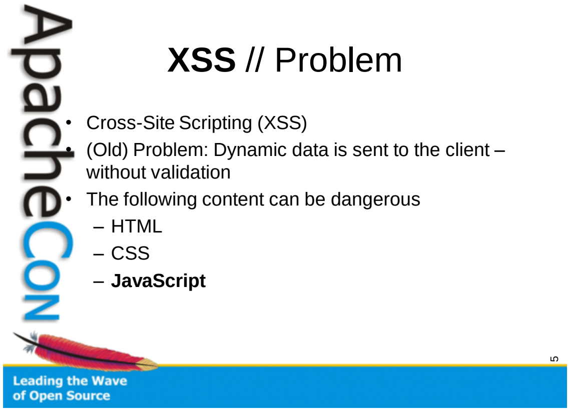
ApacheCon
Leading the Wave of Open Source



Why? // Explanations

- Bad, inconsistent advice in talks, whitepapers, articles and books
- Lack of time
- Ajax applications make it very easy to introduce vulnerabilities
  - Many new (unchecked?) server APIs
  - Applications rely on UGC (user-generated content)

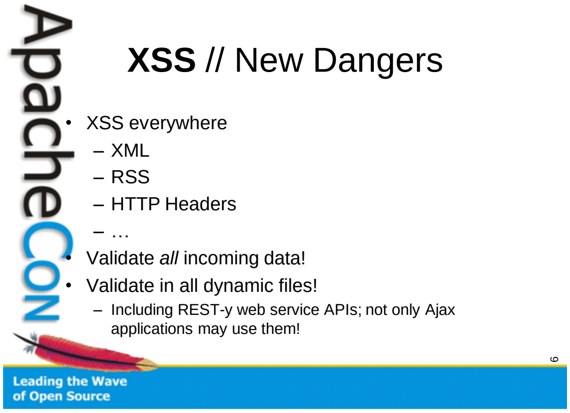
Leading the Wave of Open Source



XSS // Problem

- Cross-Site Scripting (XSS)
- (Old) Problem: Dynamic data is sent to the client – without validation
- The following content can be dangerous
  - HTML
  - CSS
  - **JavaScript**

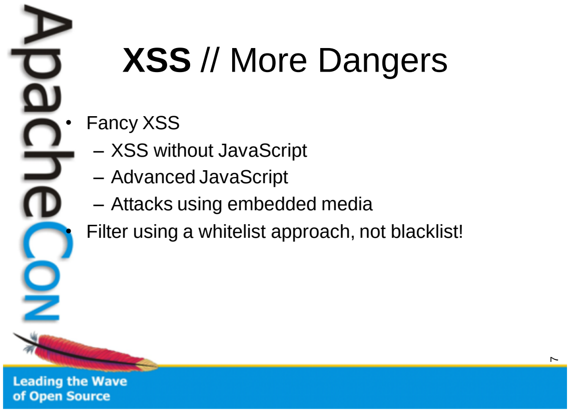
Leading the Wave of Open Source



XSS // New Dangers

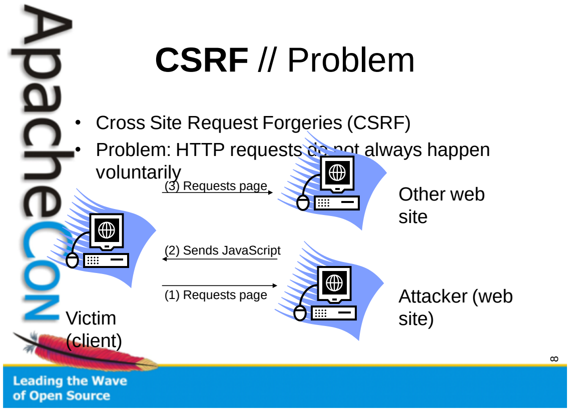
- XSS everywhere
  - XML
  - RSS
  - HTTP Headers
  - …
- Validate *all* incoming data!
- Validate in all dynamic files!
  - Including REST-y web service APIs; not only Ajax applications may use them!

Leading the Wave of Open Source



XSS // More Dangers

- Fancy XSS
  - XSS without JavaScript
  - Advanced JavaScript
  - Attacks using embedded media
- Filter using a whitelist approach, not blacklist!



CSRF // Problem

- Cross Site Request Forgeries (CSRF)
- Problem: HTTP requests do not always happen voluntarily

(3) Requests page

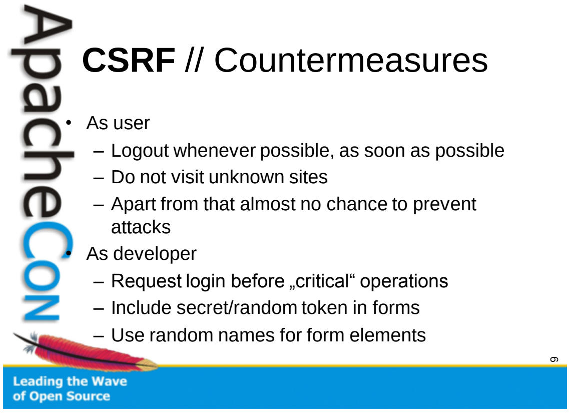
Other web site

(2) Sends JavaScript

(1) Requests page

Attacker (web site)

Victim (client)



CSRF // Countermeasures

- As user
  - Logout whenever possible, as soon as possible
  - Do not visit unknown sites
  - Apart from that almost no chance to prevent attacks
- As developer
  - Request login before „critical“ operations
  - Include secret/random token in forms
  - Use random names for form elements

ApacheCon

SQL Injection // Problem

- SQL Injection
- (Old) Problem: Dynamic data is used in SQL statements – without validation
- The list of attacks does not end with **' OR ''='** !

Leading the Wave of Open Source

ApacheCon

SQL Injection // Bad Ideas

- Filter for „1=1“
- Filter for **'**
- Filter for /*

- Again: No blacklist, but whitelist
  - Or database-specific escape functions/methods
  - Or even better: Prepared statements (if supported)

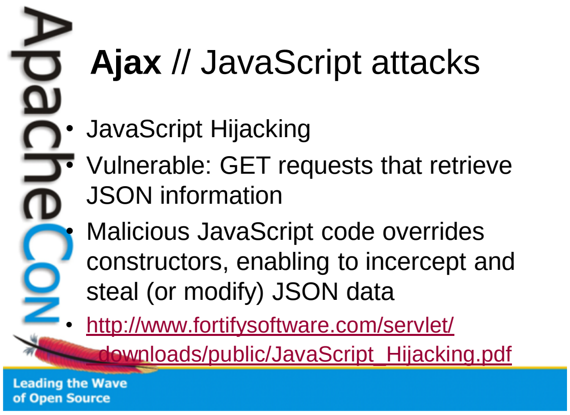
Leading the Wave of Open Source

ApacheCon

SQL Injection // Fancy attacks

- Prompting error messages
- **UNION** attacks
- Blind SQL attacks
- Using built-in functionality
- Second-order attacks
- DoS attacks

Leading the Wave of Open Source

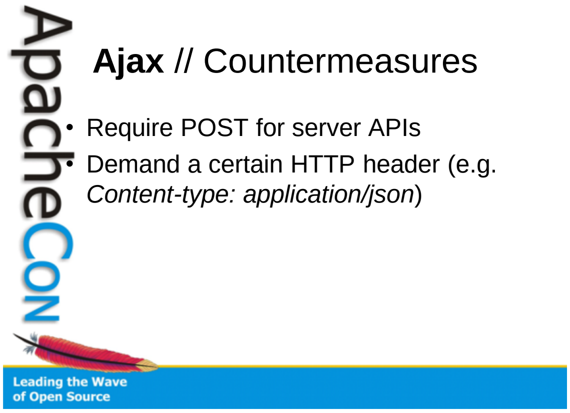


Ajax // JavaScript attacks

- JavaScript Hijacking
- Vulnerable: GET requests that retrieve JSON information
- Malicious JavaScript code overrides constructors, enabling to incercept and steal (or modify) JSON data
- http://www.fortifysoftware.com/servlet/ downloads/public/JavaScript_Hijacking.pdf

ApacheCon

Leading the Wave
of Open Source

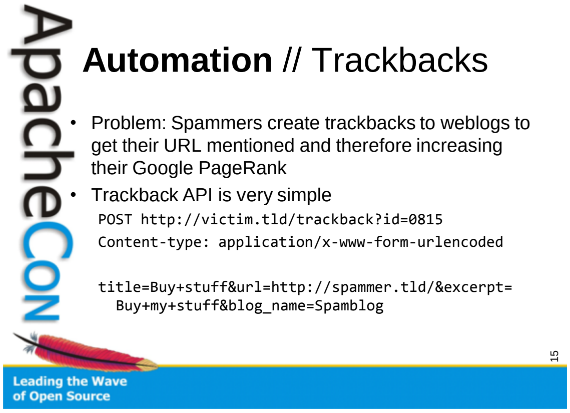


Ajax // Countermeasures

- Require POST for server APIs
- Demand a certain HTTP header (e.g. *Content-type: application/json*)

ApacheCon

Leading the Wave
of Open Source



Automation // Trackbacks

- Problem: Spammers create trackbacks to weblogs to get their URL mentioned and therefore increasing their Google PageRank
- Trackback API is very simple

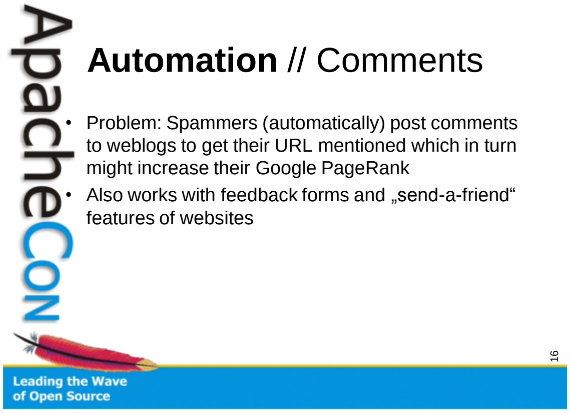
```
POST http://victim.tld/trackback?id=0815
Content-type: application/x-www-form-urlencoded

title=Buy+stuff&url=http://spammer.tld/&excerpt=
  Buy+my+stuff&blog_name=Spamblog
```

15

ApacheCon

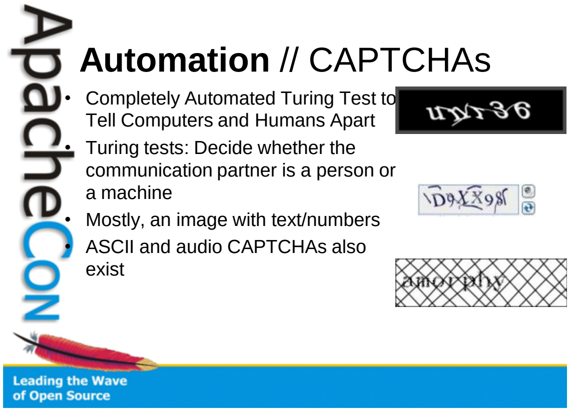
Leading the Wave
of Open Source



Automation // Comments

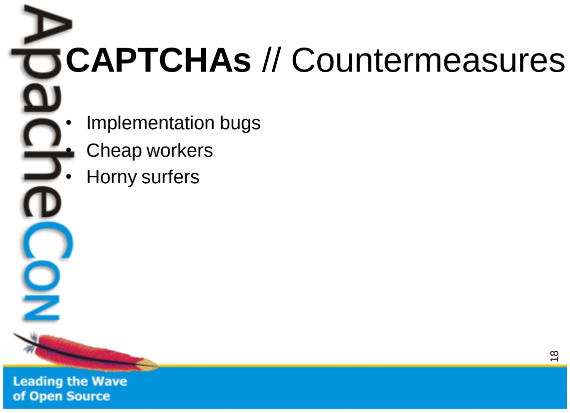
- Problem: Spammers (automatically) post comments to weblogs to get their URL mentioned which in turn might increase their Google PageRank
- Also works with feedback forms and „send-a-friend“ features of websites

16



Automation // CAPTCHAs

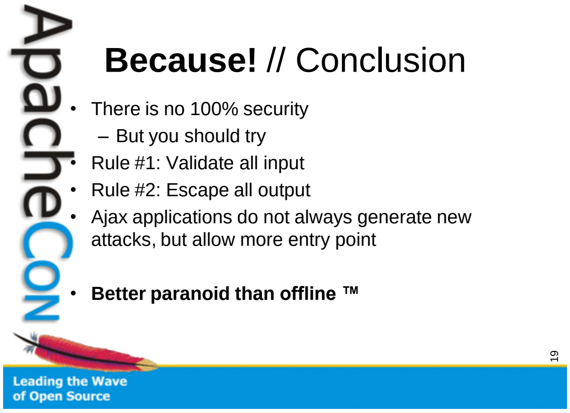
- Completely Automated Turing Test to Tell Computers and Humans Apart
- Turing tests: Decide whether the communication partner is a person or a machine
- Mostly, an image with text/numbers
- ASCII and audio CAPTCHAs also exist



CAPTCHAs // Countermeasures

- Implementation bugs
- Cheap workers
- Horny surfers

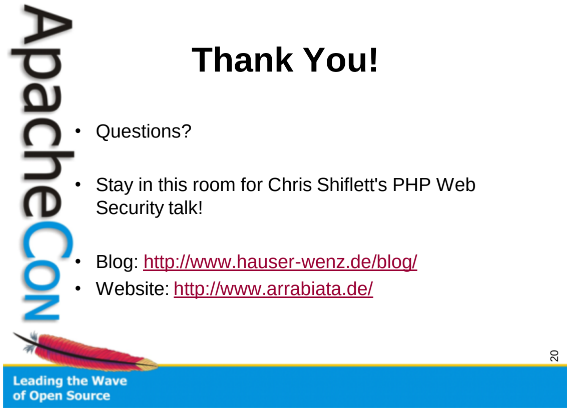
18



Because! // Conclusion

- There is no 100% security
  - But you should try
- Rule #1: Validate all input
- Rule #2: Escape all output
- Ajax applications do not always generate new attacks, but allow more entry point
- **Better paranoid than offline ™**

Leading the Wave of Open Source



Thank You!

- Questions?
- Stay in this room for Chris Shiflett's PHP Web Security talk!
- Blog: http://www.hauser-wenz.de/blog/
- Website: http://www.arrabiata.de/

Leading the Wave of Open Source