Hack & Fix

Hands on ColdFusion Security Training

Pete Freitag, Foundeo Inc.
David Epler, AboutWeb LLC

About Pete

- 17+ Years ColdFusion Experience
- Job: Foundeo Inc. Consulting & Products
  - CFSummit Gold Sponsor
  - HackMyCF / FuseGuard
- blog: petefreitag.com
- twitter: @pfreitag

About David

- 15+ years ColdFusion experience
- Job: AboutWeb - Security Architect
  - Several Security Certs: GWAPT, CEH
  - Learn CF in a Week - Security
  - OWASP Zed Attack Proxy (ZAP) Evangelist
- blog: dcepler.net
- twitter: @dcepler

Agenda

- About the VM
- File Upload Vulnerabilities
- SQL Injection
- Path Traversals
- Cross Site Scripting
- OWASP ZAP
- Sneak Peak - ColdFusion Raijin/Blizzard

About the VM

- Ubuntu Linux (don't worry)
- ColdFusion 11
- MySQL
- Username / password: cf / cf
- CF Admin Username / password: admin / cf

VM Setup

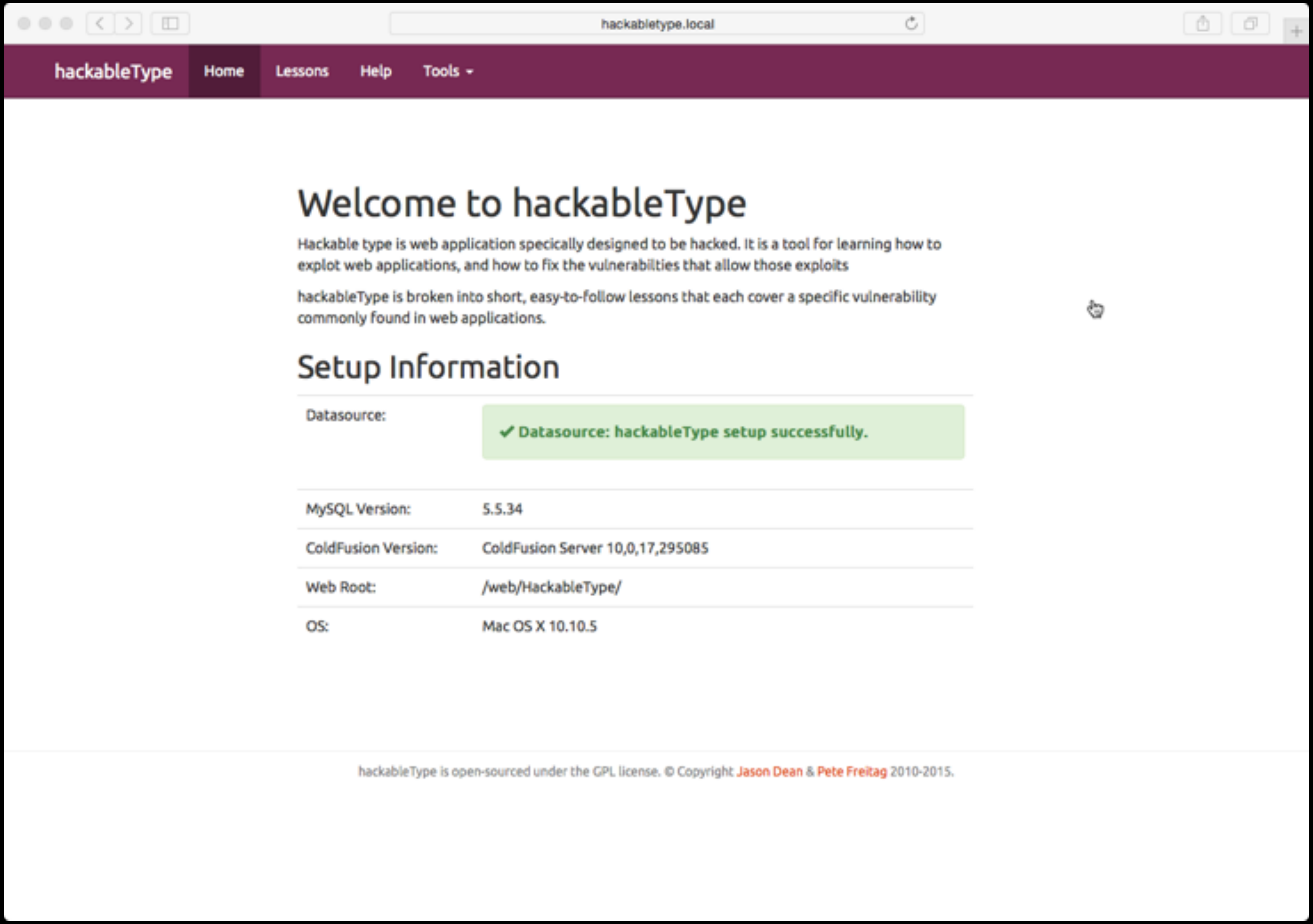
- Open Terminal
- cd /var/www/hackabletype
- git config —global user.email “cfsummit”
- git pull
- sudo a2dismod autoindex
- sudo service apache2 restart

Guiding Principals

- Defense In Depth
- Principal of Least Privilege
- Avoid Security by Obscurity
- Validation can save your bacon
- Even the best developers write insecure code.

Hackable Type

http://hackabletype.local/



Welcome to hackableType

Hackable type is web application specically designed to be hacked. It is a tool for learning how to explot web applications, and how to fix the vulnerabilties that allow those exploits

hackableType is broken into short, easy-to-follow lessons that each cover a specific vulnerability commonly found in web applications.

Setup Information

Datasource:	✓ Datasource: hackableType setup successfully.
MySQL Version:	5.5.34
ColdFusion Version:	ColdFusion Server 10,0,17,295085
Web Root:	/web/HackableType/
OS:	Mac OS X 10.10.5

File Uploads

HackableType: Try to upload and execute a CFM file.

photo (cc) flickr user armchairbuilder 2012

File Uploads Rule #1

Never trust a MIME type

Never trust a MIME

- CF9 and below use the MIME type passed by the browser / client.
  - Attacker can send any MIME type.
- CF10+ can perform server side file inspection (when strict=true, default).
  - We can still get around this.

File Uploads Rule #2

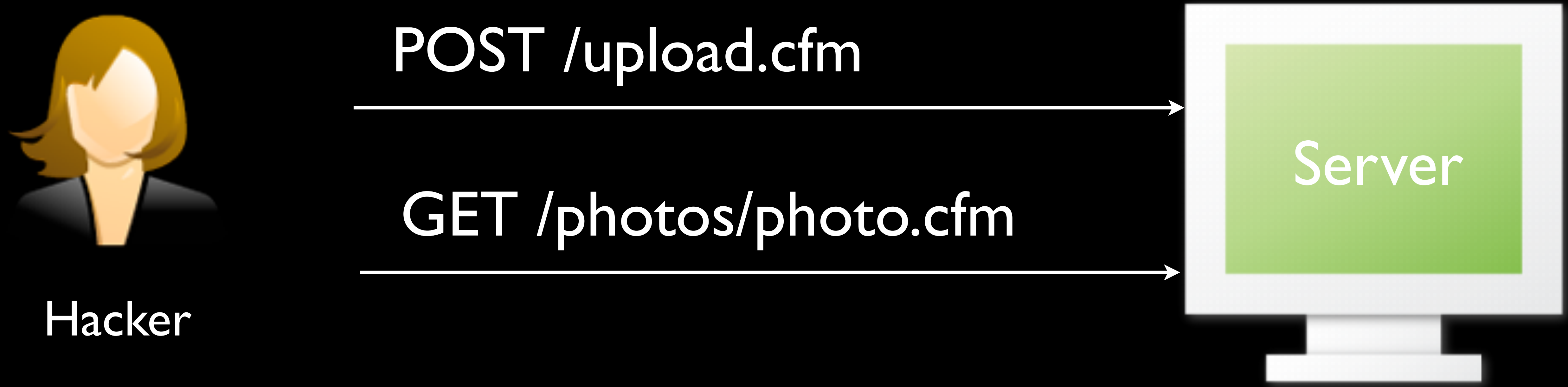
Always Validate The File Extension

Always validate file extension

- CF10 allows you to specify a file extension list in the accept attribute.
- You can also validate cffile.ServerFileExt
- Do both.

File Uploads Rule #3

Never upload directly to webroot



POST /upload.cfm

GET /photos/photo.cfm

Hacker uses a load tool to make repeated concurrent requests.

The attacker will be able to execute photo.cfm before it is deleted.

Don't upload to web root

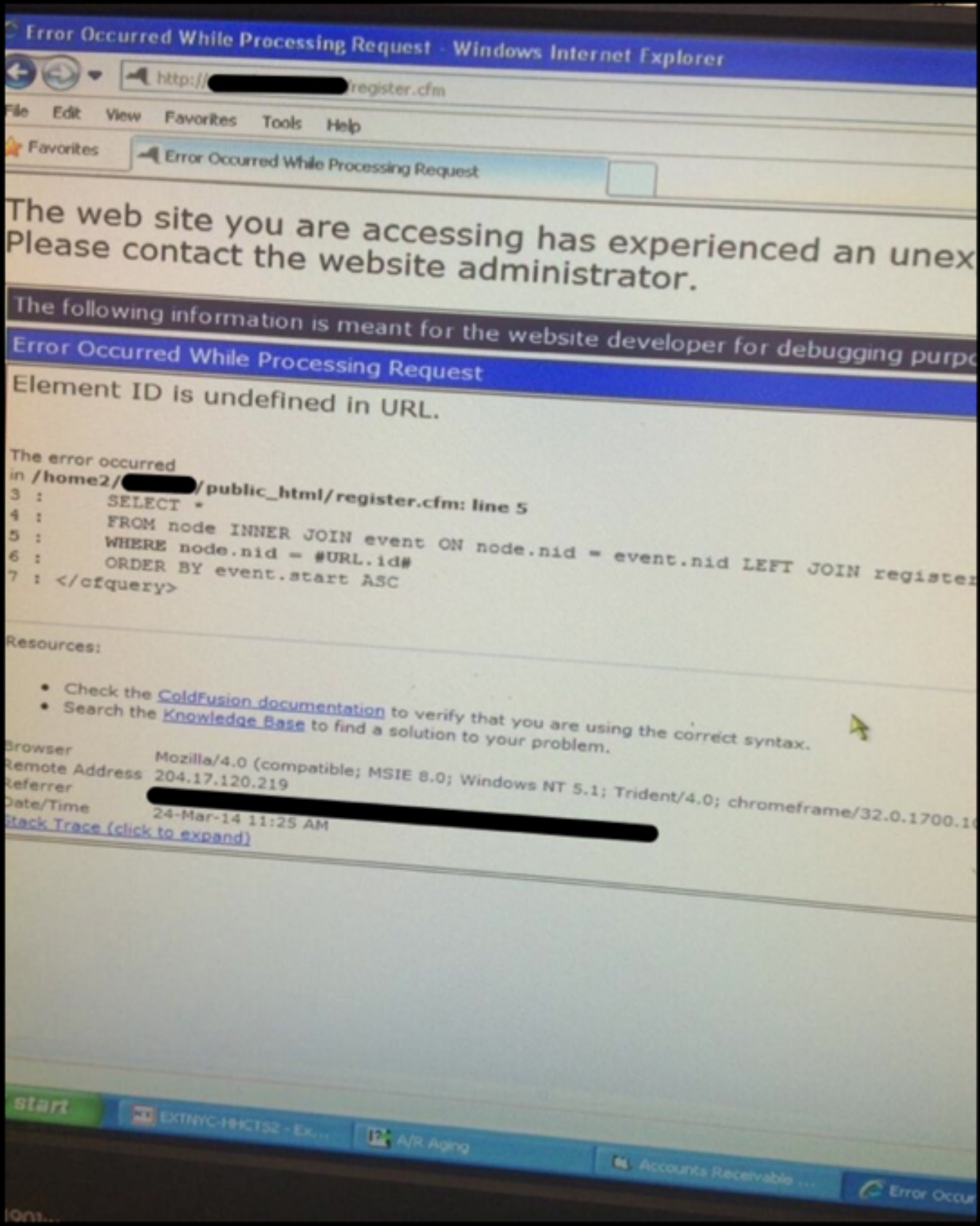
- File can be executed before it's validated.
- Upload outside root, eg GetTempDirectory ram://, s3, etc.
- Upload directly to S3: http://www.petefreitag.com/item/833.cfm

Additional Tips

- Ensure upload directory can only serve static files. Sandbox / file extension whitelist on web server.
- Consider keeping files outside webroot and serve with cfcontent or mod_xsendfile
- Specify mode on unix (eg 640 rw-r——)
- secureupload.cfc: https://github.com/foundeo/cfml-security

SQL Injection

TweetPic from someone that did not responsibly disclose issue to site owner that has SQL Injection


Error Occurred While Processing Request - Windows Internet Explorer
http://[redacted]register.cfm
File Edit View Favorites Tools Help
Favorites
Error Occurred While Processing Request
start
EXTNYC-HHCTS2 - Ex...
A/R Aging
Accounts Receivable ...
Error Occur

The web site you are accessing has experienced an unex
Please contact the website administrator.

The following information is meant for the website developer for debugging purpo

Error Occurred While Processing Request

Element ID is undefined in URL.

```
The error occurred
in /home2/[redacted]/public_html/register.cfm: line 5
3 :         SELECT *
4 :         FROM node INNER JOIN event ON node.nid = event.nid LEFT JOIN register
5 :         WHERE node.nid = #URL.id#
6 :         ORDER BY event.start ASC
7 : </cfquery>
```

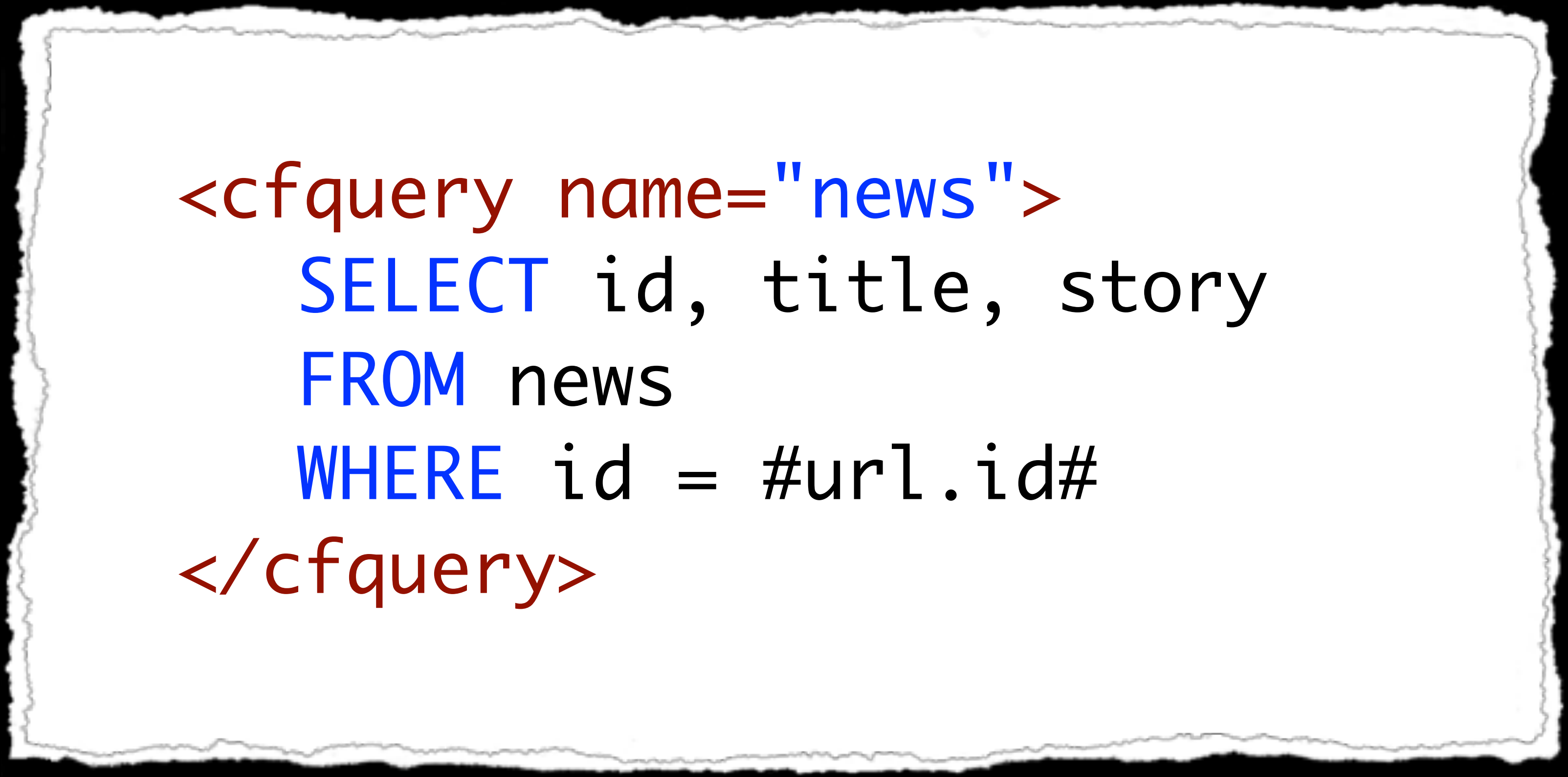
Resources:

- Check the ColdFusion documentation to verify that you are using the correct syntax.
- Search the Knowledge Base to find a solution to your problem.

```
Browser         Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; chromeframe/32.0.1700.1
Remote Address  204.17.120.219
Referrer        [redacted]
Date/Time       24-Mar-14 11:25 AM
```

Stack Trace (click to expand)

SQL Injection



```
<cfquery name="news">
   SELECT id, title, story
   FROM news
   WHERE id = #url.id#
</cfquery>
```

news.cfm?id=1;delete+from+news

SQL Injection

- The solution - use parameters (eg cfqueryparam) whenever possible.
- Validate and sanitize when you can't
    - ORDER BY *column*
    - SELECT TOP *10*
- ORM: make sure HQL statements are parameterized

SQL Injection

Try the lesson

Path Traversal Vulnerabilities

Path Traversal Risk

- Attacker can read any file CF has permission to read
  - Configuration files
  - System Files
  - Logs
- Remote code execution possible in some cases.

HackableType

Try the path traversal lesson

Preventing Path Traversals

- Avoid file paths derived from user input.
- Strip and validate any variables used in paths. Dots and slashes are dangerous.
- Beware of null bytes
- On windows use multiple drive letters to separate application from OS, CF, logs, etc.

Path Traversal Bonus Round

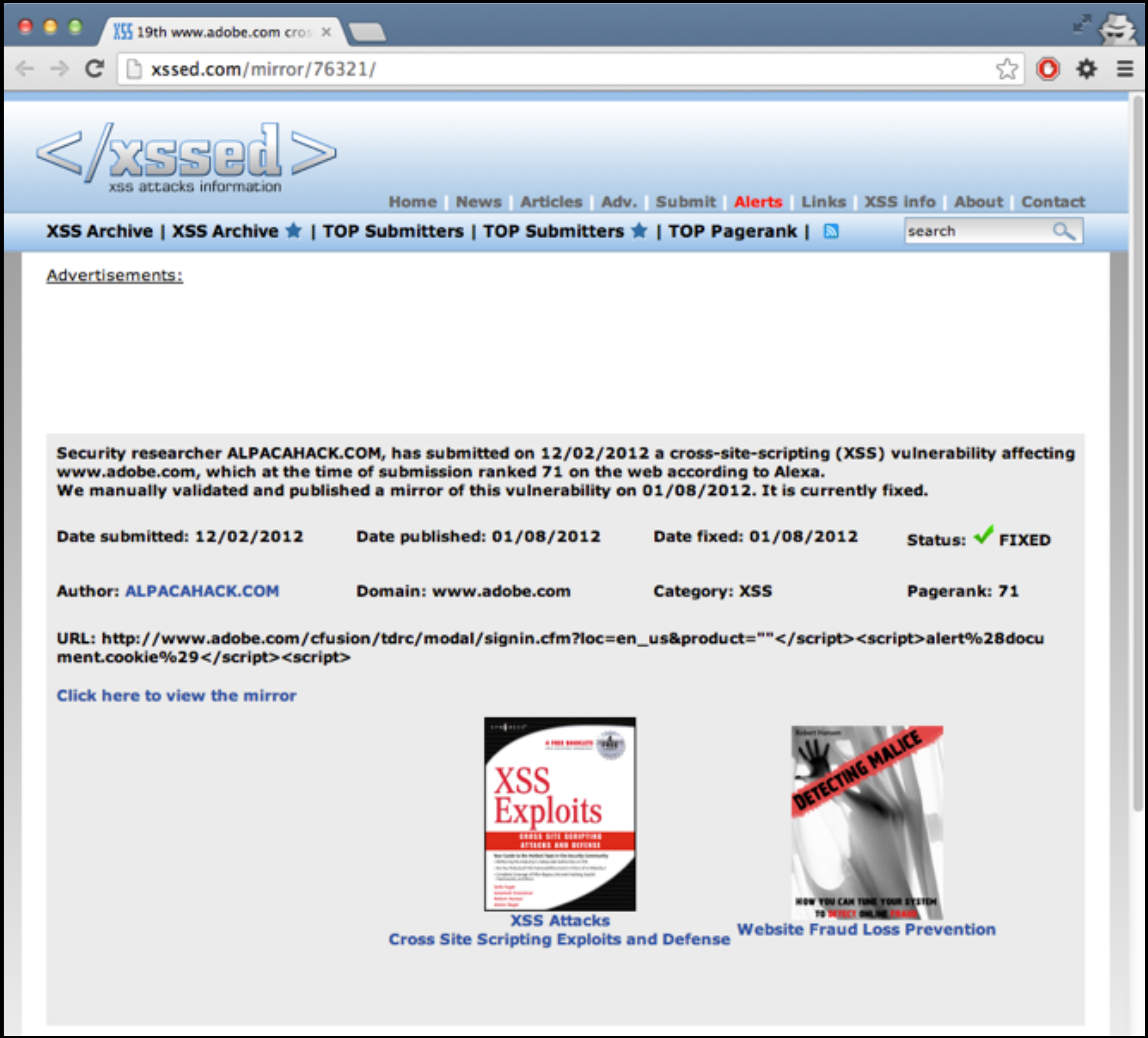
Can you use the path traversal lesson to perform remote code execution?

Path Traversal

- Possible Remote Code Execution via cfinclude
  - CF11+ added Application.cfc and ColdFusion administrator setting:

```
this.compileExtForInclude="cfm";
```

Cross Site Scripting (XSS)


19th www.adobe.com cros
xssed.com/mirror/76321/
</xssed>
xss attacks information
Home | News | Articles | Adv. | Submit | Alerts | Links | XSS info | About | Contact
XSS Archive | XSS Archive ★ | TOP Submitters | TOP Submitters ★ | TOP Pagerank |
search
Advertisements:

Security researcher ALPACAHACK.COM, has submitted on 12/02/2012 a cross-site-scripting (XSS) vulnerability affecting www.adobe.com, which at the time of submission ranked 71 on the web according to Alexa.
We manually validated and published a mirror of this vulnerability on 01/08/2012. It is currently fixed.

Date submitted: 12/02/2012 Date published: 01/08/2012 Date fixed: 01/08/2012 Status: ✓ FIXED

Author: ALPACAHACK.COM Domain: www.adobe.com Category: XSS Pagerank: 71

URL: http://www.adobe.com/cfusion/tdrc/modal/signin.cfm?loc=en_us&product=""</script><script>alert%28document.cookie%29</script><script>

Click here to view the mirror

XSS Exploits
DETECTING MALICE
XSS Attacks
Cross Site Scripting Exploits and Defense
Website Fraud Loss Prevention

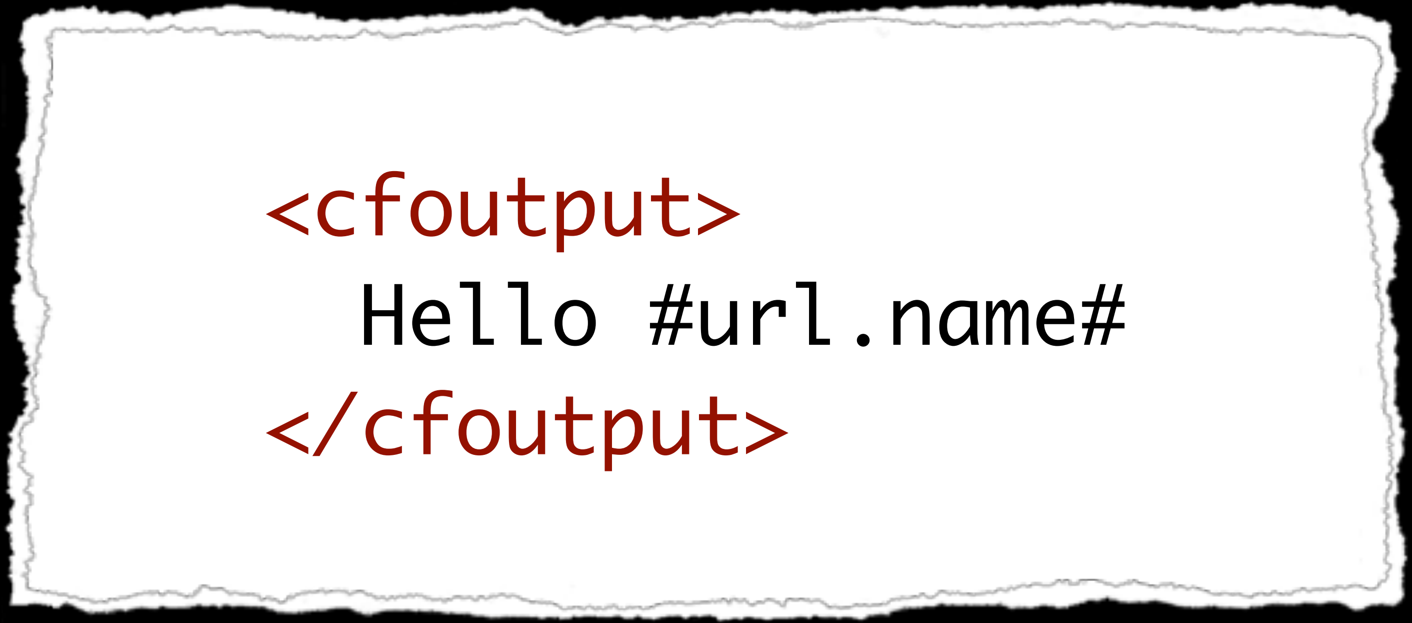
XSS

- XSS holes give attackers a CMS to create any content.
- Can be used to steal sessions
- Phish for passwords or other info.

XSS Types

- Reflected
- Persistant
- DOM

Reflected XSS



```
<cfoutput>
  Hello #url.name#
</cfoutput>
```

hello.cfm?name=<script>...</script>

Reflected XSS

Try the lesson

Preventing XSS

- Strip out dangerous characters
  - < > ' " ( ) ; #
- Escape dangerous characters
  - CF10+ EncodeForHTML, etc.

Preventing XSS

Context	Method
HTML	`encodeForHTML(variable)`
HTML Attribute	`encodeForHTMLAttribute(variable)`
JavaScript	`encodeForJavaScript(variable)`
CSS	`encodeForCSS(variable)`
URL	`encodeForURL(variable)`

XSS in HTML

- Preventing XSS when allowing users to enter HTML is difficult.
    - AntiSamy -> isSafeHTML getSafeHTML
    - ScrubHTML

XSS Utils

- **Encoders**
  - ESAPI: http://www.petefreitag.com/item/788.cfm
  - OWASP Encoder: http://owasp-java-encoder.googlecode.com

- **Sanitizers**
  - AntiSamy: http://www.petefreitag.com/item/760.cfm
  - ScrubHTML: https://github.com/foundeo/cfml-security

OWASP ZAP

- An easy to use web application penetration testing tool
- Completely free and Open Source
- OWASP flagship project
- Included in major security distributions
  - Kali, Samurai WTF, etc.

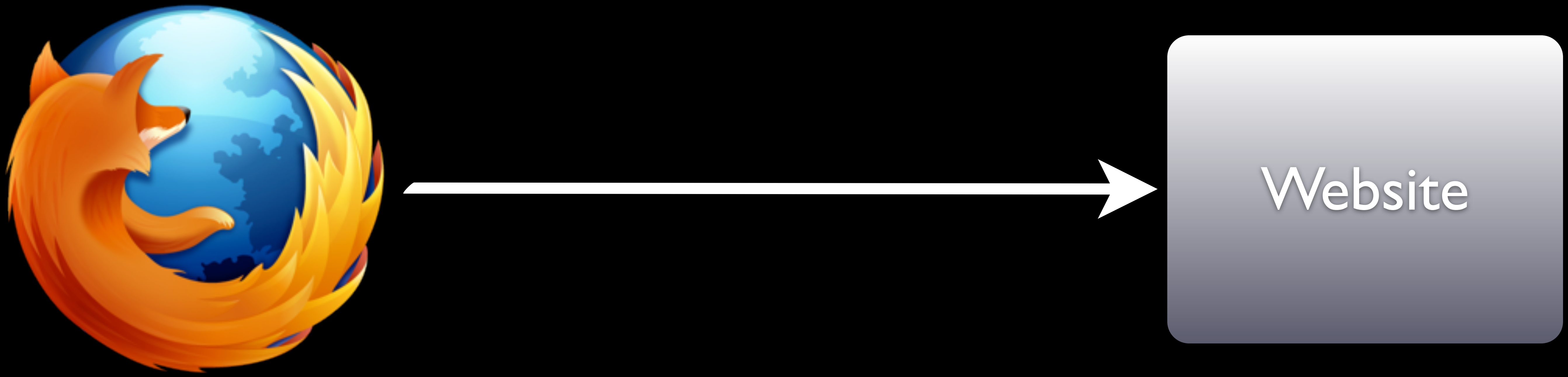
Why use ZAP?

- Ideal for beginners, developers
  - also used by professional pen testers
- Point and shoot via Quick Start Tab
- Manual penetration testing
- As a debugger
- As part of larger security program
  - Automated security regression tests

Main ZAP Features

- Intercepting Proxy
- Active and Passive Scanners
- Traditional and AJAX spiders
- Forced browsing
- Fuzzing
- Cross Platform
    - built on Java (requires 1.7+)

Intercepting Proxy



Using ZAP

Hands on

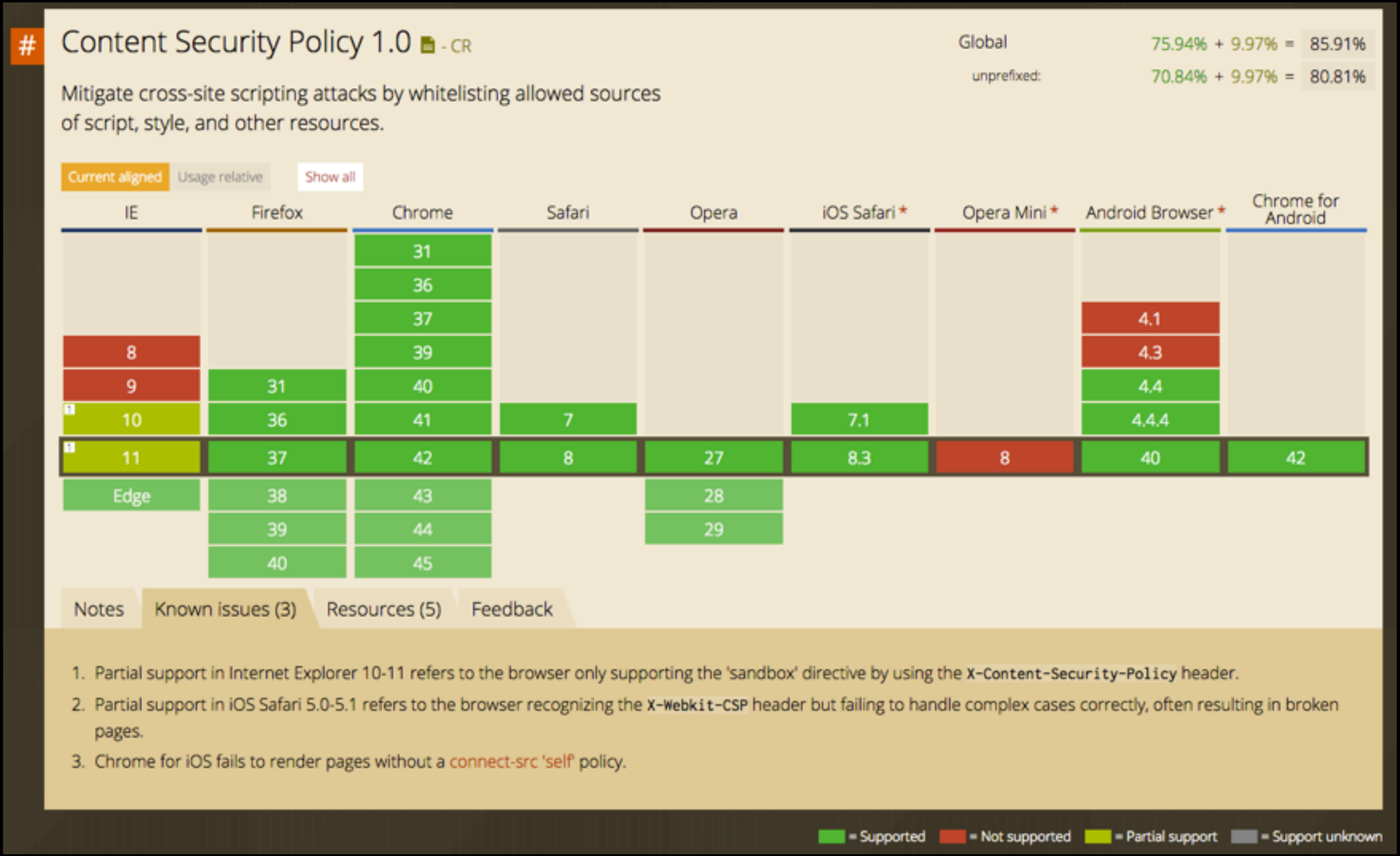
Content-Security-Policy

- HTTP Response Header dictates what assets can be loaded. For example:
    - script-src 'self';
    - script-src 'self' cdn.example.com;
    - script-src 'none';
    - script-src 'unsafe-inline';

CSP Directives

- default-src
- script-src
- style-src
- img-src
- connect-src
- font-src
- object-src
- media-src
- frame-src
- sandbox
- report-uri

CSP 1.0 Browser Support



Content Security Policy 1.0 - CR

Mitigate cross-site scripting attacks by whitelisting allowed sources of script, style, and other resources.

Global	75.94% + 9.97% =	85.91%
unprefixed:	70.84% + 9.97% =	80.81%

IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		🟩 31						
		🟩 36						
		🟩 37					🟥 4.1	
🟥 8		🟩 39					🟥 4.3	
🟥 9	🟩 31	🟩 40					🟩 4.4	
🟨 10 [1]	🟩 36	🟩 41	🟩 7		🟩 7.1		🟩 4.4.4	
🟨 11 [1]	🟩 37	🟩 42	🟩 8	🟩 27	🟩 8.3	🟥 8	🟩 40	🟩 42
🟩 Edge	🟩 38	🟩 43		🟩 28				
	🟩 39	🟩 44		🟩 29				
	🟩 40	🟩 45						

1. Partial support in Internet Explorer 10-11 refers to the browser only supporting the 'sandbox' directive by using the `X-Content-Security-Policy` header.
2. Partial support in iOS Safari 5.0-5.1 refers to the browser recognizing the `X-Webkit-CSP` header but failing to handle complex cases correctly, often resulting in broken pages.
3. Chrome for iOS fails to render pages without a connect-src 'self' policy.

🟩 = Supported 🟥 = Not supported 🟨 = Partial support ⬜ = Support unknown

http://caniuse.com/#feat=contentsecuritypolicy

CSP 1.0 Browser Support

- Chrome 25+
- FireFox 23+
- Safari 7+
- IE Edge 12+
    - Partial Support in IE10+ (sandbox)

CSP Level 2

- Notable Enhancements
  - Nonce
  - Hash
  - form-action directive

CSP Lesson

- Hint: content-security-policy.com

Want More?

- Scope Injection Lesson
- CSRF Lesson

ColdFusion

Raijin/Blizzard

Security Analyzer

Questions?

Thank You!

Pete Freitag
pete@foundeo.com
foundeo.com

David Epler
depler@aboutweb.com
dcepler.net