ColdFusion 10 Security Enhancements

by Pete Freitag, Foundeo Inc.

petefreitag.com | foundeo.com | hackmycf.com

foundeo

inc.

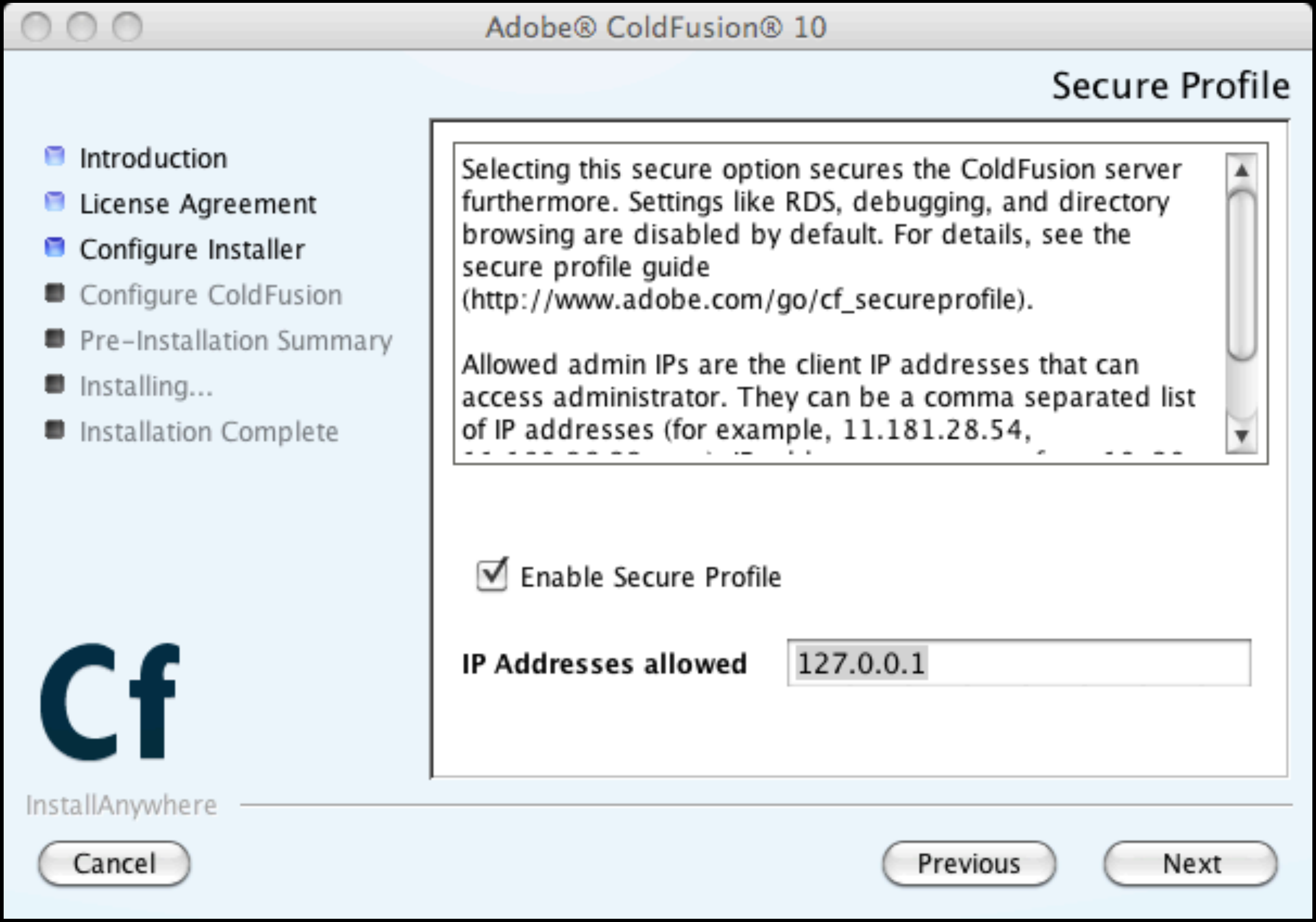
Who am I

- Owner Foundeo Inc.
    - ColdFusion Consulting
    - Products: FuseGuard, HackMyCF
- Adobe Community Professional
- 14 Years ColdFusion Experience
    - Author
    - Blog: petefreitag.com
    - Twitter: @pfreitag

Agenda

- ColdFusion 10 Server Security Enhancements
- ColdFusion 10 Language Enhancements to increase Security
  - New Functions
  - Application Settings

Secure Profile



Selecting this secure option secures the ColdFusion server furthermore. Settings like RDS, debugging, and directory browsing are disabled by default. For details, see the secure profile guide (http://www.adobe.com/go/cf_secureprofile).

Allowed admin IPs are the client IP addresses that can access administrator. They can be a comma separated list of IP addresses (for example, 11.181.28.54,

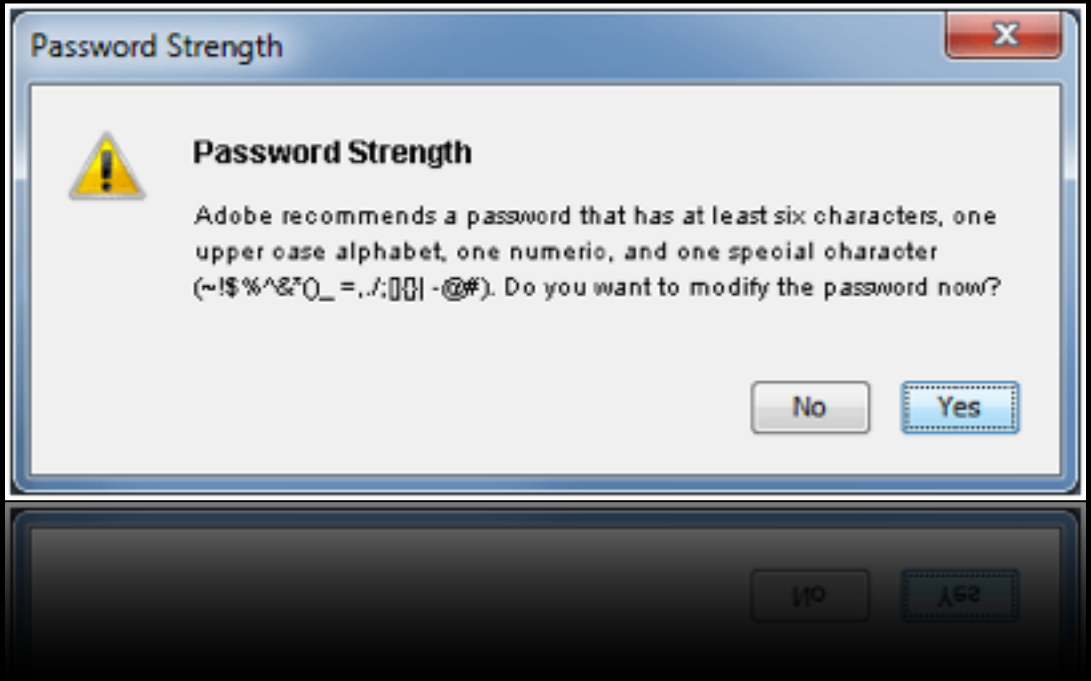
☑ Enable Secure Profile

IP Addresses allowed 127.0.0.1

Secure Profile

- Disables RDS, Flash Remoting, Web Sockets
- Various CF Admin Settings
- Full List Here:
  - http://www.adobe.com/go/cf_secureprofile

Server Passwords



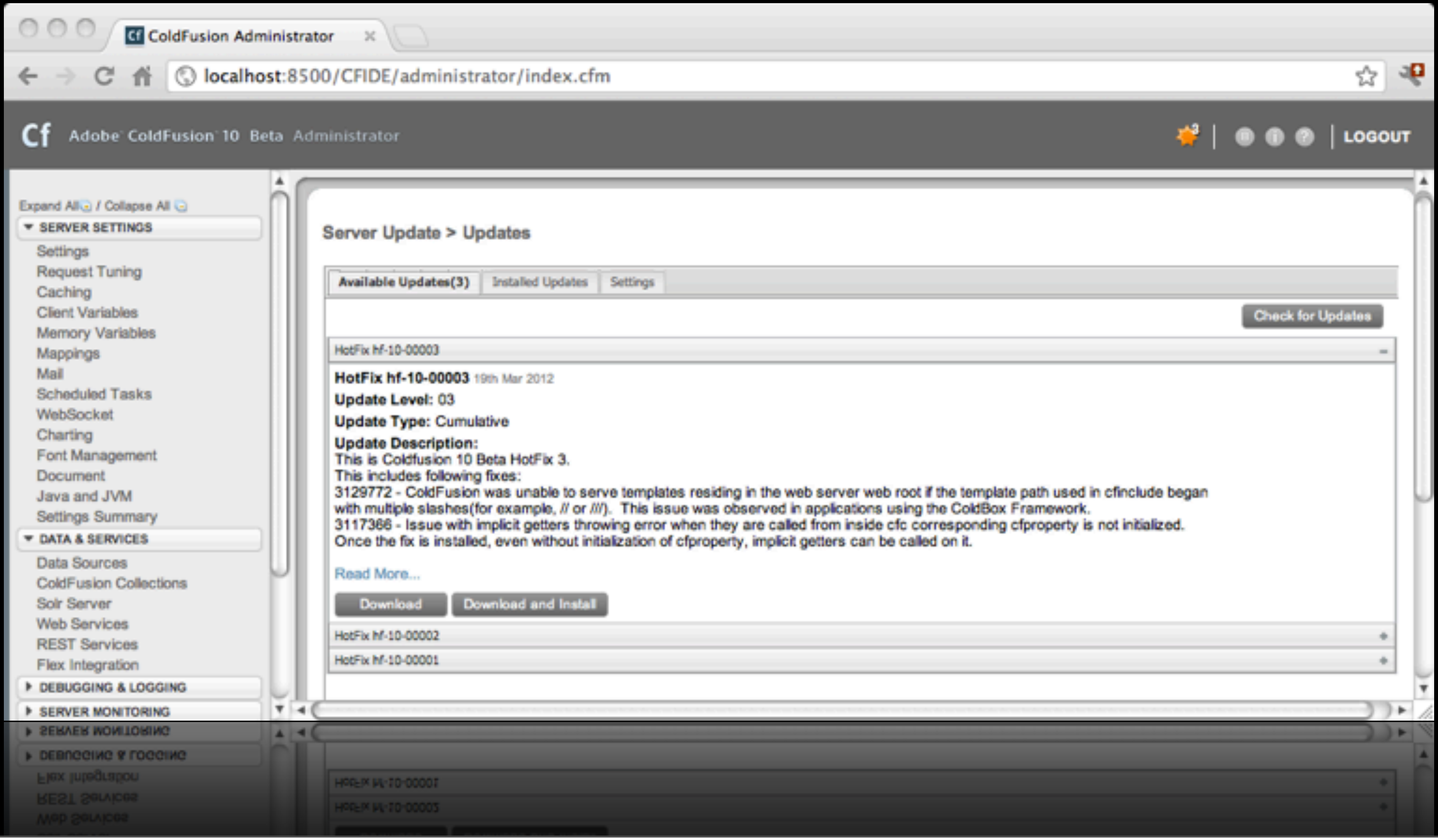
Password Strength

Password Strength

Adobe recommends a password that has at least six characters, one upper case alphabet, one numeric, and one special character (~!$%^&*()_ =,./:[]{}| -@#). Do you want to modify the password now?

- Warns of weak passwords
- All service passwords encrypted

Hotfix Installer



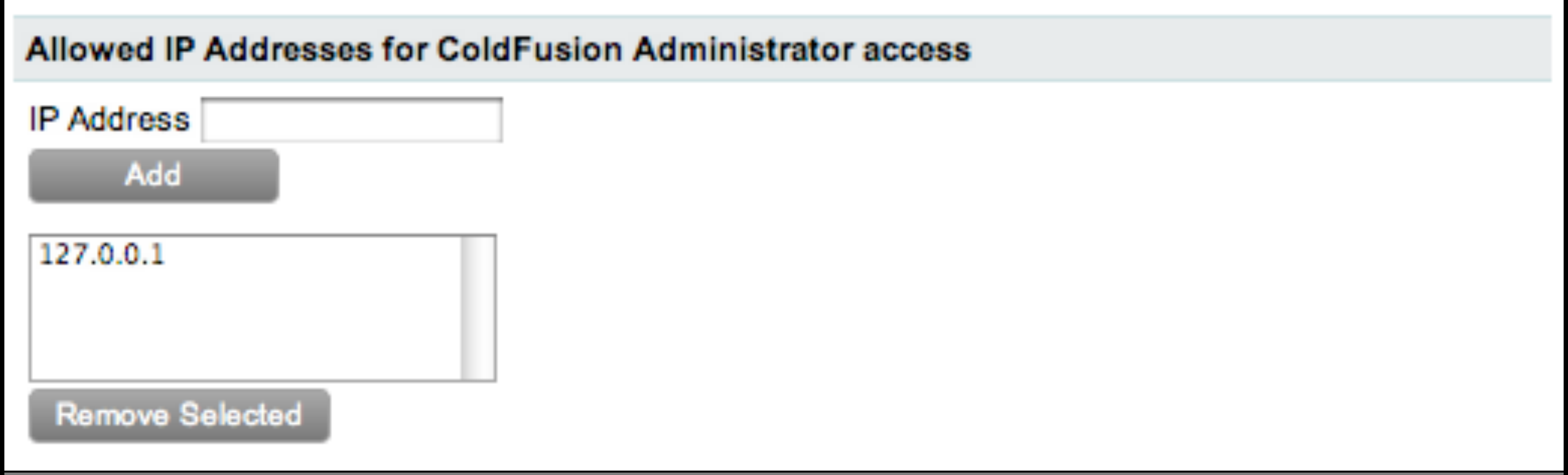
HotFix hf-10-00003 19th Mar 2012

Update Level: 03

Update Type: Cumulative

Update Description:
This is Coldfusion 10 Beta HotFix 3.
This includes following fixes:
3129772 - ColdFusion was unable to serve templates residing in the web server web root if the template path used in cfinclude began with multiple slashes(for example, // or ///). This issue was observed in applications using the ColdBox Framework.
3117366 - Issue with implicit getters throwing error when they are called from inside cfc corresponding cfproperty is not initialized.
Once the fix is installed, even without initialization of cfproperty, implicit getters can be called on it.

CF Administrator IP Restrictions

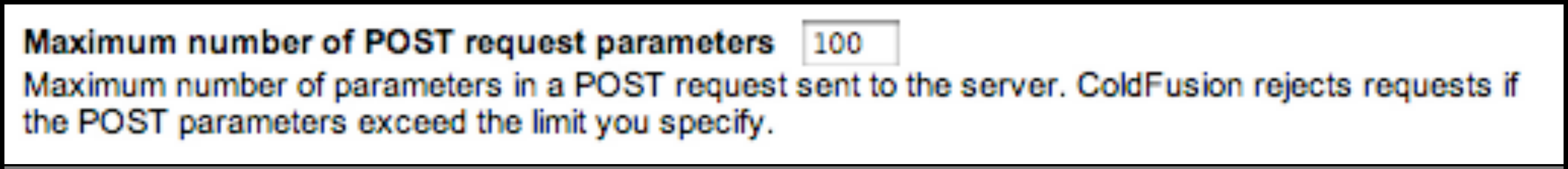


Allowed IP Addresses for ColdFusion Administrator access

IP Address

127.0.0.1

Limit Number of POST Variables



Maximum number of POST request parameters 100

Maximum number of parameters in a POST request sent to the server. ColdFusion rejects requests if the POST parameters exceed the limit you specify.

Also added to CF 9.0.2

Secure Defaults

- Enable UUID For CFTOKEN on by default
- ScriptProtect on by default
    - Note: scriptProtect has very limited ability to protect from XSS.

Tomcat

- Newer Servlet Specs offer more security controls
- Wider deployment than JRun
- Security Issues Patched Quickly

Session Hijacking

- If I know your CFID / CFTOKEN (or JSESSIONID) values then I can authenticate as you.
  - Session ID's are just as valuable as a password, while they are valid.

Preventing Session Hijacking

- Keep session ids out of the url
    - cflocation addtoken=false
- Use SSL
- Cookies typically best transport mechanism

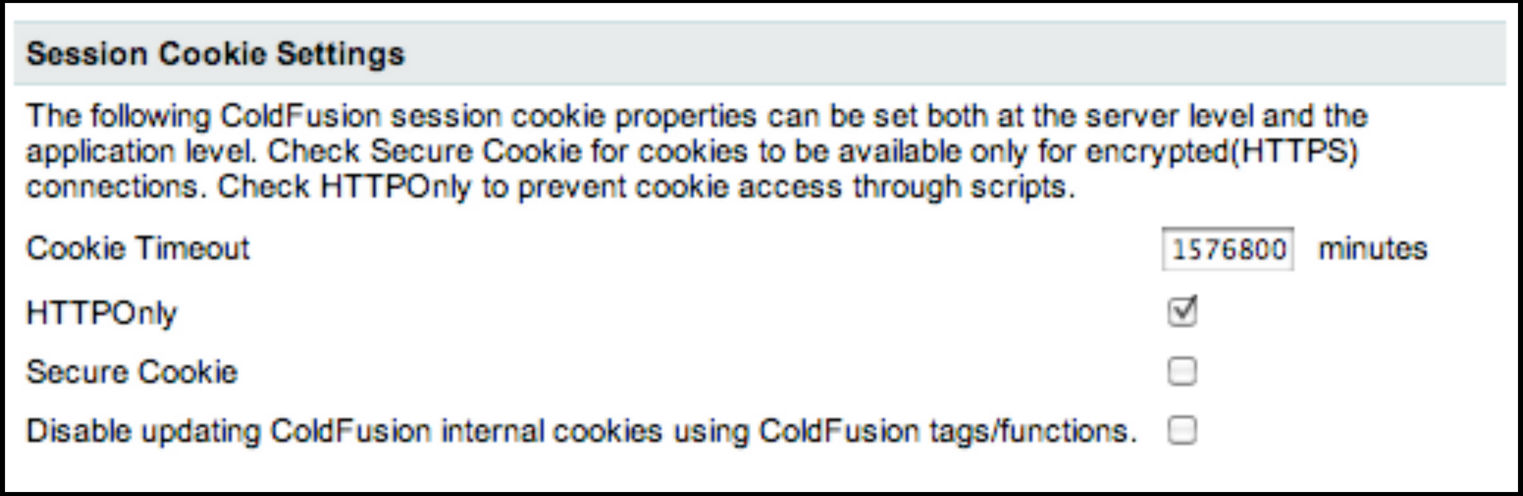
Secure Cookies

- When the **secure** attribute is present the browser only sends the cookie over a *secure* connection (SSL/https).
    - Browser support nearly ubiquitous
- Use **secure** for session cookies

HttpOnly Cookies

- When cookies are set with the **HttpOnly** attribute the browser restricts access to it from "non-http API's" (JavaScript)
  - Supported on Modern Browsers, but also does not break old browsers.
- Use HttpOnly for Session Cookies to prevent session hijacking via XSS

New Session Cookie Settings in ColdFusion Administrator



Session Cookie Settings

The following ColdFusion session cookie properties can be set both at the server level and the application level. Check Secure Cookie for cookies to be available only for encrypted(HTTPS) connections. Check HTTPOnly to prevent cookie access through scripts.

Cookie Timeout 1576800 minutes

HTTPOnly ☑

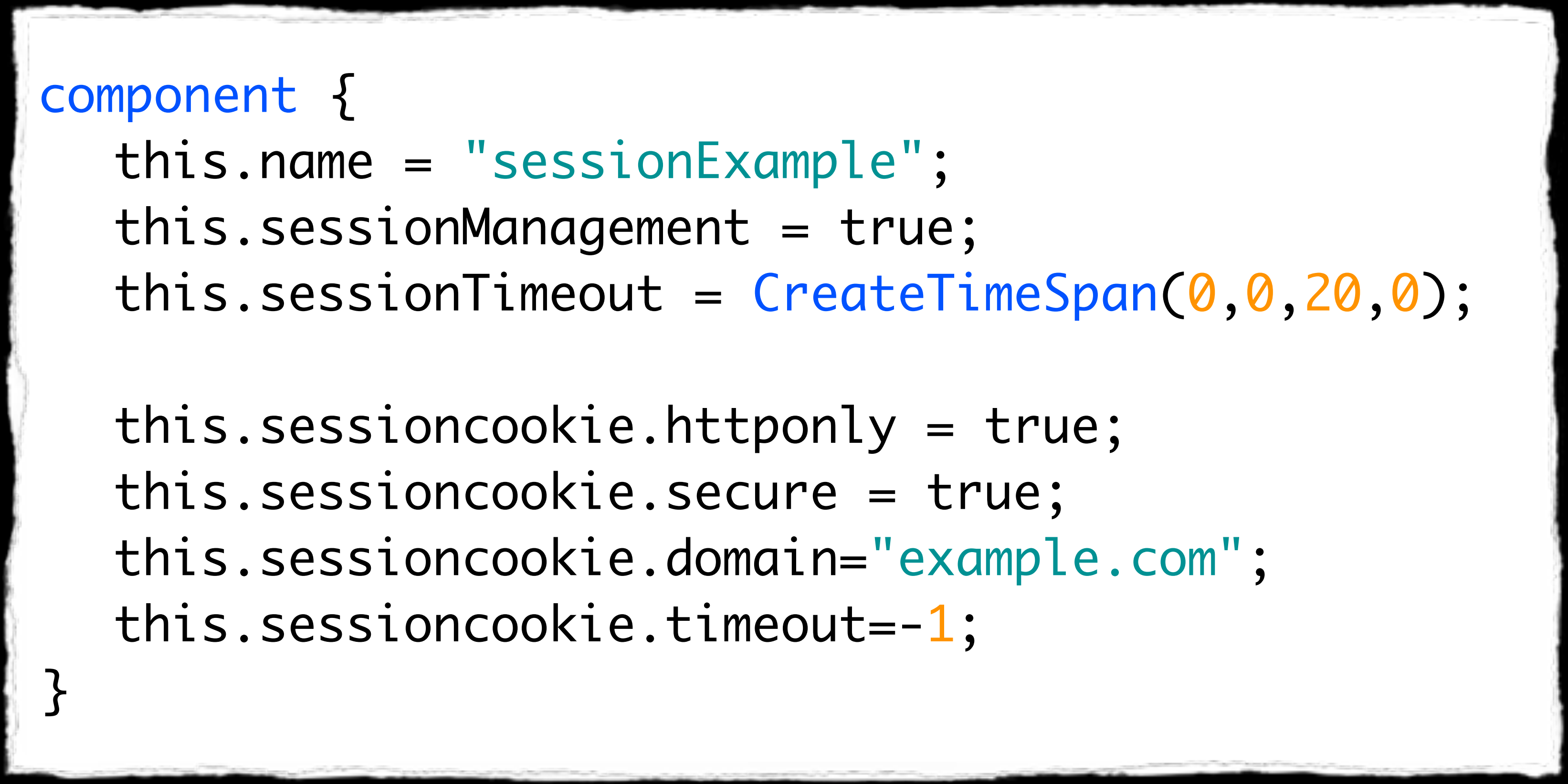
Secure Cookie □

Disable updating ColdFusion internal cookies using ColdFusion tags/functions. □

Session Cookie Settings

- **Cookie Timeout** - Defaults to 3 years, you should lower this.
- **HttpOnly** - Defaults on, keep it on.
- **Secure** - Defaults off, turn on globally if all sites on server require SSL.
- **Disable Updating ColdFusion internal cookies using tags & functions** - defaults off

Session Cookie Settings in Application.cfc



```
component {
   this.name = "sessionExample";
   this.sessionManagement = true;
   this.sessionTimeout = CreateTimeSpan(0,0,20,0);

   this.sessioncookie.httponly = true;
   this.sessioncookie.secure = true;
   this.sessioncookie.domain="example.com";
   this.sessioncookie.timeout=-1;
}
```

SessionRotate()

- New Function SessionRotate()
    - Invalidates Current Session
    - Generates new Session ID, sets new cookies.
    - Copies old session vars into new session
    - Does not invoke onSessionStart()

Why Rotate Sessions?

- Call **SessionRotate** after successful authentication to prevent session fixation attacks.

SessionInvalidate

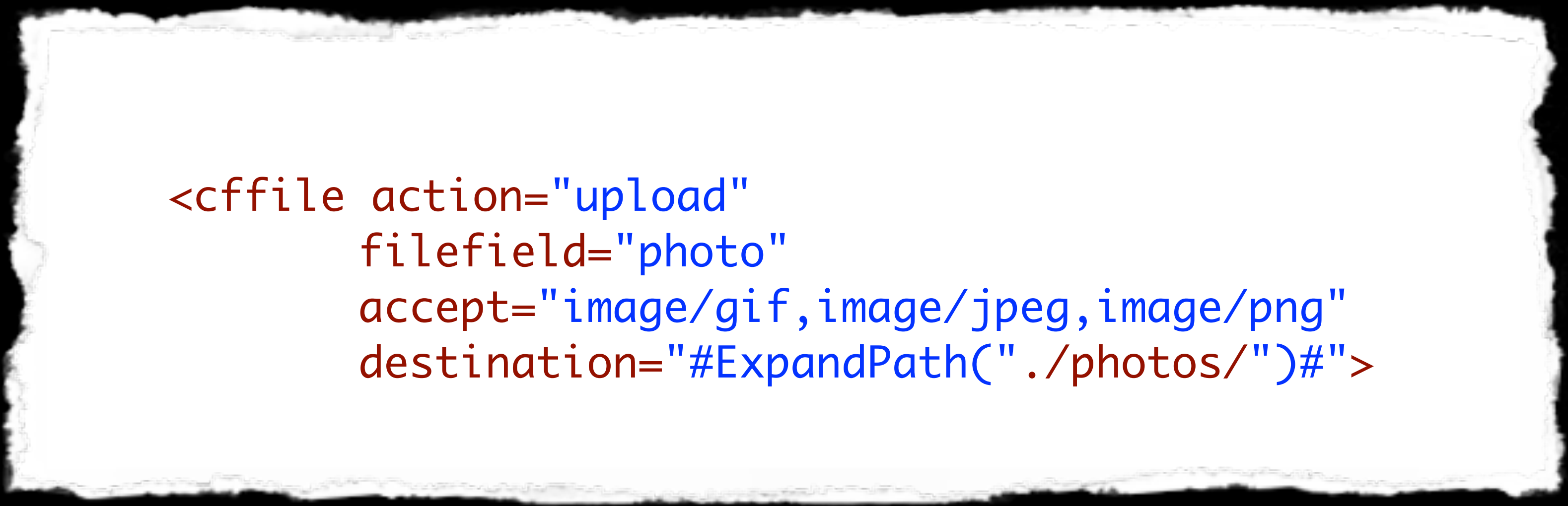
- Destroys a session
- For J2EE sessions does not invalidate underlying jsessionid.
- Call upon logout

Session Demos

File Uploads

- Very Dangerous yet common requirement
- If careless attacker may upload and execute a file on the server.

Vulnerable Code



```
<cffile action="upload"
        filefield="photo"
        accept="image/gif,image/jpeg,image/png"
        destination="#ExpandPath("./photos/")#">
```

File Upload Demos

File Uploads

- The cffile accept attribute now supports file extensions:
  - accept="*.jpg,*.png"
  - strict="true/false"

fileGetMimeType

- fileGetMimeType(*filePath, [strict]*)
  - Inspects file contents to determine mime type
  - When strict=false just checks file extension.

My Recommendation

- Use File Extensions in **accept** attribute.
- Then Validate Type using fileGetMimeType and/or other methods.
- Don’t mix file extensions and mime types in accept attribute.

Cross Site Scripting

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

Exploiting XSS

- Instead of hello.cfm?name=pete
- Attacker runs:
  - hello.cfm?name=<script>alert(‘pete’)</script>

Is XSS That Bad?

Cross Site Scripting

- The Risks:
  - Session Hijacking
    - POST Forms via AJAX
  - Phishing (steal passwords, credit cards, etc.)
  - Publish Content on your site

Fixing XSS

- One Solution: Strip all harmful characters
    - < > ' " ( ) ; #
- Not always a realistic solution.

Fixing XSS

- Encode variables to escape special characters. (eg < becomes &lt;)
    - Proper encoding depends where you output it, HTML, JavaScript, CSS etc.

Output Context’s

Context	Example
HTML	`<p>Hello #url.name#</p>`
HTML Attribute	`<div id="#url.name#" />`

JavaScript	`<a onclick="hi(#url.name#)" />`
	`<script>#var#</script>`

CSS	`<div style="font-family: #url.name#" />`
	`<style>#var#</style>`

URL	`<a href="hi.cfm?name=#url.name#" />`

In CF9 we can use:

- XMLFormat() or HTMLEditFormat()
  - XMLFormat Escapes < > ' "
  - HTMLEditFormat Escapes <> "

CF10 Gives Us

- New Encoder Methods leveraged from OWASP Enterprise Security API
  - Java API that has encoder methods for each context.
  - http://code.google.com/p/owasp-esapi-java/

Using ESAPI

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

Encoder Method Demos

Canonicalize()

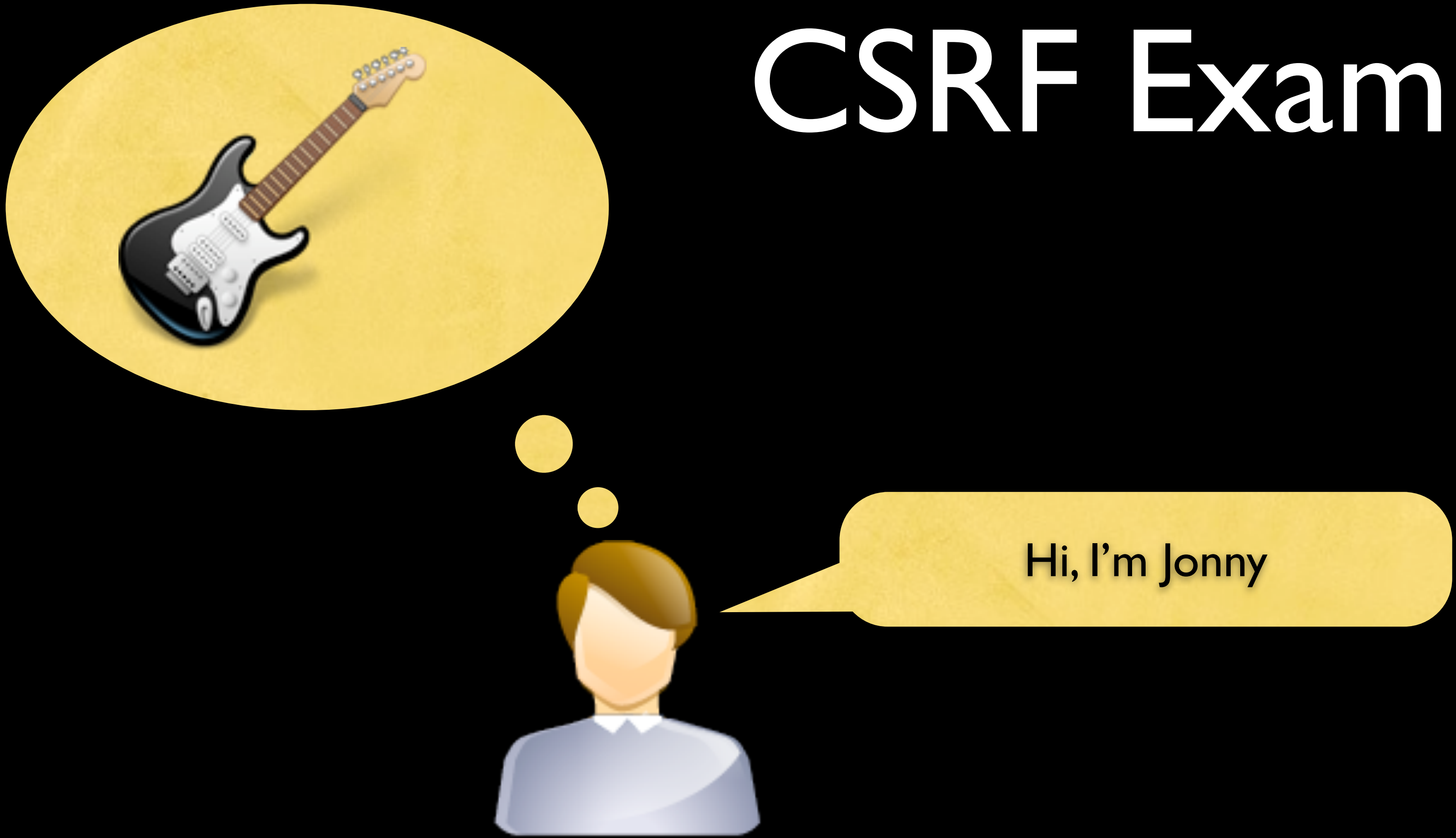
- Pronounced kuh-non-ical-ize :)
- Canonicalization is the operation of reducing a possibly encoded string down to its simplest form
- canonicalize(inputString, restrictMultiple, restrictMixed)
- Call before validation

CFForm

- Restricts characters you can use in the **name** attribute of cfinput, etc.
- No longer populates cfform action attribute if omitted
  - You can re-enable this with a jvm setting however.

Cross Site Request Forgery

CSRF Example



Jonny is currently logged into auction site: hack-bay.com

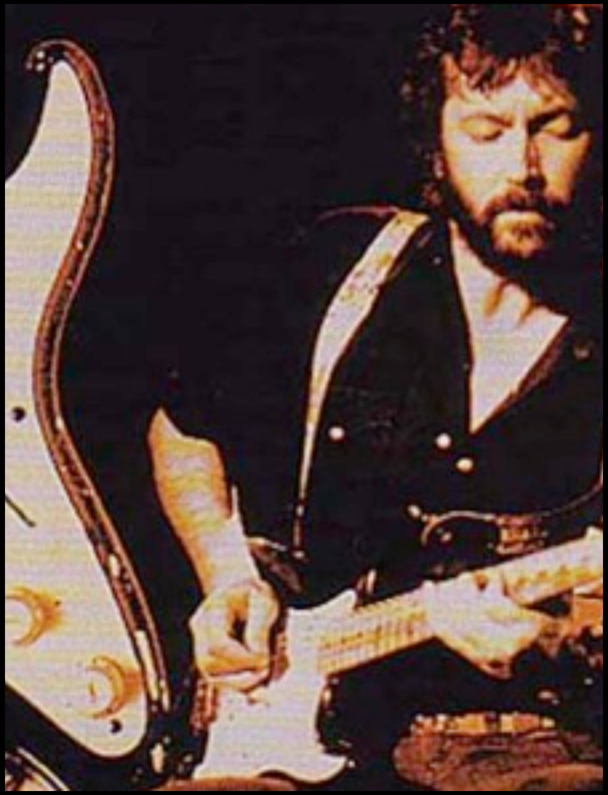
CSRF Example

Jane - is this really Eric Clapton's Strat?

Hi Jonny, Yes, check out this photo:
http://bit.ly/1337

Sweeeet!!

CSRF Example



```
<img src="http://hacker.example.com/clapton.jpg" />

<img src="http://hack-bay.com/bid.cfm?item=123&amount=80000" height="1" width="1" />
```

CSRF Example

- Jonny just bid $80,000 on the guitar, by clicking on the link from Jane.

Fixing CSRF

- Require method = POST
  - CSRF still possible with POST, but more difficult.

Fixing CSRF

- Reject Foreign Referrers
  - Doesn’t fix XSS + CSRF
  - Referrer might not be present / spoofed.

Fixing CSRF

- Require Password or Captcha
  - Not very usable, but sometimes essential.

Fixing CSRF

- Random Token
  - Include a random token as a hidden field.
  - Store the token in a session variable
  - Compare the hidden form field with session variable on form action page.

New CSRF Token Functions

- CSRFGenerateToken([key], [forceNew])
- CSRFVerifyToken(token, [key])
- Must enable session variables
  - tokens stored in session internally

CSRF Function Demo

Hash

- ColdFusion 10 adds the **iterations** argument.
    - Increases hash computation time.

Hashing

- A hash provides a **one way** encoding of a string into a fixed length string.
  - Unlike Encryption which is two way (you can get the original string again if you have the key)
- Use ColdFusion's Hash(string, algorithm, encoding, iterations) function:
  - Hash("password", "SHA-512")

Hash Algorithms

- MD5 - Default Algorithm of the Hash Function, Fast not as secure
- SHA - Secure Hash Algorithm FIPS
  - SHA-1 160 bit Algorithm designed by the NSA
  - SHA-2 (SHA-256 and SHA-512) also designed by the NSA
  - SHA-3 winner will be announced by NIST Q2 2012
- Algorithm support determined by JCE. ColdFusion Enterprise installs RSA BSafe Crypto-J Provider for FIPS-140 Compliance.

Each User Has Same Password

uid	password
1	5F4DCC3B5AA765D61D8327DEB882CF99
2	5F4DCC3B5AA765D61D8327DEB882CF99
3	5F4DCC3B5AA765D61D8327DEB882CF99

No Salt

uid	password
1	8FD974D2D58F875F968AF667994C951B
2	DF982CE25D47C6E8ECA7BEE61AE972C3
3	BE721CAA292A226EA58E8089CF422407

Salted

HMAC

- Hash-based Message Authentication Code
  - Hash + a Secret Key
- Commonly used for authenticating API Requests.
  - Sign request variables and a timestamp using a shared secret key.

HMAC

- HMAC(msg, key, algorithm, encoding)
- Algorithms: HMAC-MD5, HMAC-RIPEMD160, HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512

Misc Enhancements

- RSA Crpyto-J Library Upgraded to Version 5 (from Version 3.6 in 9.0.1)
- Application.cfc setting to make Ram Disk ram:/// isolated to current application.
- CFLogin more secure defaults for authorization cookie.

Thank You!

pete@foundeo.com

petefreitag.com | foundeo.com | hackmycf.com

foundeo

inc.