

ColdFusion Security

Configuration and Implementation

Presented to the Philadelphia CFUG

<http://www.phillycfug.org/>



Not Just Buzzword Compliance

- Threats will always be there
- Developers are key players
- Server administrators are equally important

Didn't you already do this talk?

- Yes, but...
 - CF8 has some great new tools
 - We all need a refresher
 - It's that important.

- Over 10 years old
 - Not a single major exploit
- Most issues are the result of:
 - Infrastructure security
 - Poor configuration
 - Inadequate administration
 - Bad/sloppy code

Three ingredients

- Configuration
- Implementation
- Luck 😊

Configuration

- Default settings after installation are not tight!
- In 30 minutes you'll have a much safer server
- As the “ColdFusion person” you have to instruct your server folks!

Four Basic Rules

1. Don't tip your hand
2. Lock it up
3. Use protection
4. Get tested

Rule #1: Don't tip your hand

- Disable debugging output
 - This is probably a “duh” to most of you
- Specify a missing template handler
 - Like a 404, but not handled by the web server
- Specify a default sitewide error handler
 - Will only execute if your application doesn't do its job

Rule #1: Don't tip your hand

- Create a useful 404 page
 - Humor not necessary, but intelligence is
- Authentication
 - Don't tell the user if it was the username or password that was bad
 - Never submit login info in a GET request
- Clean up your installation
 - Remove cfdocs from web directory

Four Basic Rules










1. Don't tip your hand
2. Lock it up
3. Use protection
4. Get tested

Rule #2: Lock it up

- Application timeout: 8 hours or less (max and default)
- Session timeout: 20 minutes or less (max and default)
- Now you can set these per application in CF8!

Rule #2: Lock it up

- Choose session persistence carefully

	CF	J2EE
	cfid/cftoken	jsessionid
Compatibility		
Serializable		
Termination	Explicit	 On browser close
Client vars	 compatible	
Shareable with jsp/servlets?		

- If you use CF session management, use UUIDs for session tokens
 - Lower likelihood for collisions (and brute force session hijacking)

Rule #2: Lock it up

- Remove anonymous access to /cfide/administrator
 - Even better, restrict IP addresses allowed to access that folder

- RDS
 - Disable it in production
 - Run it over SSL in development
 - Create named user accounts for RDS access on shared dev boxes

Rule #2: Lock it up

- Create multiple administrative users (new in CF8)
 - Long awaited feature!
 - Roles defined here control degree of admin API access too
 - Works in conjunction with sandbox security

Rule #2: Lock it up

- Run ColdFusion as its own user account
 - Confine access to /cfusionmx and relevant site files

- Shut down all irrelevant services on the app server
 - FTP, NNTP, a bajillion Windows services

Four Basic Rules

1. Don't tip your hand
2. Lock it up
3. Use protection
4. Get tested

Rule #3: Use Protection

- “Enable global script protection”
 - Helps to guard against XSS attacks
 - Still a rather novel feature for an application server
 - Don’t rely on this exclusively
- Disable access to certain keywords in your DSNs
 - CREATE, DROP, GRANT, REVOKE shouldn’t be part of your application’s vocabulary (you can add ALTER and TRUNCATE to that list too)
 - Reinforce with access controls on the database user account

Rule #3: Use Protection

- Limit resource use
 - Maximum number of simultaneous requests (variable)
 - Maximum size of POST data (variable)
 - Timeout requests after 30 seconds (or less)
 - Override on a case-by-case basis with cfsetting

Four Basic Rules

1. Don't tip your hand
2. Lock it up
3. Use protection
4. Get tested

Rule #4: Get tested

- I use ScanAlert for quarterly (and ongoing) vulnerability scans
 - They beat the pants off your application
 - Common and not-so-common threats
 - Great for verifying PCI compliance too
 - \$150/yr. (no excuses...play like a champion)

- Tell your network guys you're getting scanned before it happens

Implementation

- There's only so much your server can do for you
- Think about “how could someone break this”
- Code defensively, even if it takes longer (and it will!)

Application settings

- New feature in CF8: override settings on a per-application basis

```
<cfscript>
```

```
    this.name = "myApplication";  
    this.applicationTimeout = createTimeSpan(1,0,0,0);  
    this.sessionmanagement = "true";  
    this.sessiontimeout = createTimeSpan(0,0,20,0);  
    this.scriptProtect = "form,url,cgi,cookie";
```

```
</cfscript>
```

- You should still set secure/tight defaults!

Garbage in...

- Client side (Javascript, Actionscript)
 - Not to be relied on 100%, but certainly nicer to the user
 - Many, many libraries exist to help (qForms, CFFORM)
- Ajax
 - If this is an “Ajaxified” application, use Ajax to do server side validation in real time
 - Prefix JSON structures with a custom string (cfadmin)
 - VerifyClient() in all ajax-facing functions and pages
- Form structure
 - Set maxlength on all text inputs (and validate on the backend too)

- Server-side validation
 - `<cfparam>` as much as possible
 - POST data
 - Custom tag fields
 - `<cfargument type="">` on every CFFUNCTION tag
 - Use `<cfinvokeargument >` too if possible
 - `trim()`, `left()` and `right()` strings before they get too far and trip up the database!

- Database

```
<cfqueryparam cfsqltype="cf_sql_integer" value="...">
```

- Stored procedures (cfprocparam)
- NEVER leave a query exposed to SQL injection attacks!
- Implement constraints on your database

```
<cfquery name="qBad" datasource="#variables.dsn#">
```

```
select * from user where
```

```
Username = '#form.username#' and password='#form.password#'
```

```
</cfquery>
```

- Watch your access attribute
 - Private, Public, Package, Remote
 - Choose wisely!

- Var your variables
 - <http://code.google.com/p/var-scope-checker-fb/downloads/list>

- Type checking

Encryption

- Use SSL for transmission
- Always encrypt passwords if you need to store them
 - Store the keys outside of the web root
 - Same goes for application configuration information
- CF8 has enhanced encryption capabilities
 - AES/DES and more in CF enterprise edition
 - Based on RSA BSafe Crypto-J libraries

- Cross site scripting is nasty, but easy to defeat
 - Assume the user is malicious
 - Based on that assumption, never output values entered by users
 - Sanitize all inputs first
 - RegEx can help with this significantly

Dumb little things

- Always comment with `<!-- word -->` NOT `<!-- word -->`
 - But comment liberally just the same!
 - Don't neglect `hint=""` attributes...CFML is wonderfully self documenting
- Don't just remove `session.user...` `structClear(session)` is safer
- When emailing exception information to yourself
 - Sanitize the email content (no passwords, no credit card numbers, etc!)
 - Include dump of CGI variables (browser info)

Useful links

- <http://www.adobe.com/devnet/coldfusion/security.html>
- CF7-CF8 admin changes
 - http://carehart.org/blog/client/index.cfm/2007/7/3/cf8_admin_changes
- CF7 Security article
 - http://www.adobe.com/devnet/coldfusion/articles/cf7_security.html
- CF8 Security whitepaper
 - http://www.adobe.com/devnet/coldfusion/articles/dev_security/coldfusion_security_cf8.pdf
- CF8 Admin – User Manager
 - Docs are available in the cfadmin help only