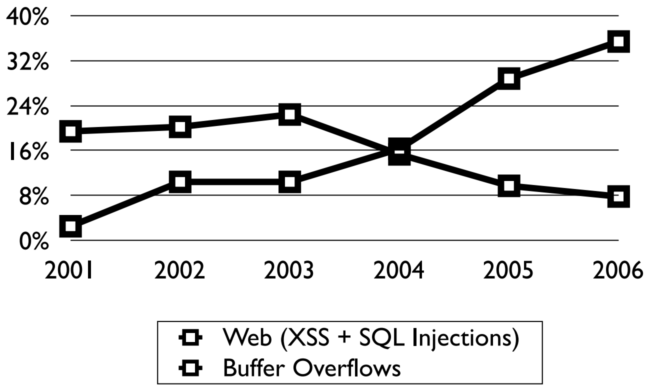
Building Secure ColdFusion Applications

Presented By Pete Freitag
Principal Consultant, Foundeo Inc.

The Plan:

1. Unchecked Input
2. File Uploads
3. XSS - Cross Site Scripting
4. SQL Injection
5. Cross Site Request Forgery
6. CRLF Injection
7. Session Hijacking

A Hot Topic



Source: http://cwe.mitre.org/documents/vuln-trends.html#table1

Unchecked Input

- The Cause of Most Security Problems
- Server Side Validation
  - IsValid Function
  - Regular Expressions

What Are The Inputs?

- URL Variables
- FORM Variables
- Cookies
- HTTP Request Headers (CGI Scope)
  - User Agent
  - Referrer
  - Other Headers

Uploading Files

- A common task that can be very dangerous.

Example: File Uploads

Best Practices for File Uploads

- Upload to a directory outside the web root or to a static content server.
- Always Check the File Extension
  - cffile.serverFileExt
- Use the “accept” attribute, but never trust it.
- Check File Names as well

Cross Site Scripting

- Attacker crafts a request that executes a client side script.
  - Usually JavaScript
  - Flash
  - Applet
  - IFRAME
  - ActiveX

What’s So Bad About XSS

- Stealing Cookies
- Phishing

XSS Examples

ScriptProtect

- ColdFusion MX 7 Introduced ScriptProtect feature.
  - Catches many but not all XSS attacks.
  - Enabled globally or at the application level.
  - Configurable Regular Expressions
    - `WEB-INF/cfusion/lib/neo-security.xml`

Preventing XSS

- Escape HTML Tags and Quotes and more.
  - XMLFormat()
    - Escapes double quotes, single quotes and <tags>.
  - HTMLEditFormat()
    - Escapes <tags> and double quotes but not single quotes.
- Make Your Own Function
  - Escape: < > ‘ “ ( ) ; #

Preventing XSS

- Validate Inputs
- Enforce Maximum String Length

SQL Injection

- Very Dangerous
  - Execute ANY SQL Statement
  - Or ANY Program!
    - xp_cmdshell
- Very Easy to Prevent

Classic SQL Injection Example

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

/news.cfm?id=8;DELETE+FROM+news

Preventing SQL Injection

```
<cfquery datasource=”db” name=”news”>
   SELECT title, story
   FROM news
   WHERE
     id = <cfqueryparam value=”#url.id#”
                 cfsqltype=”cf_sql_integer”>
</cfquery>
```

CFQUERYPARAM

- Can and should be used in
  - WHERE Clauses
  - INSERT Statements
  - UPDATE Statements
  - All variables in your query
    - Where allowed

Cross Site Request Forgery

- How “samy”, a MySpace user made 1 million friends in less than 20 hours.

Cross Site Request Forgery

- Samy found a clever way to execute javascript on his MySpace profile page.
  - Whenever a MySpace user visited his profile samy’s script would add himself as a friend on their profile.
  - For a few hours Samy caused MySpace to shut down for “maintenance”.

Cross Site Request Forgery

- Takes advantage of a logged in user.
  - Performs a privileged action on their behalf.

CSRF + XSS

- You don’t need an XSS hole to perform a Cross Site Request Forgery (CSRF).
  - However, with an XSS hole, HTTP POST requests can be executed behind the scenes with AJAX.
- CSRF could be performed by an IFRAME on a malicious web site.

Cross Site Request Forgery Example

Mitigating CSRF Attacks

- Server Side Confirmations
- Require HTTP POST when performing operations.
- Don’t allow foreign HTTP referrers.
- Require password for sensitive operations.
- Include a hash in the form based on authenticated user’s credentials.

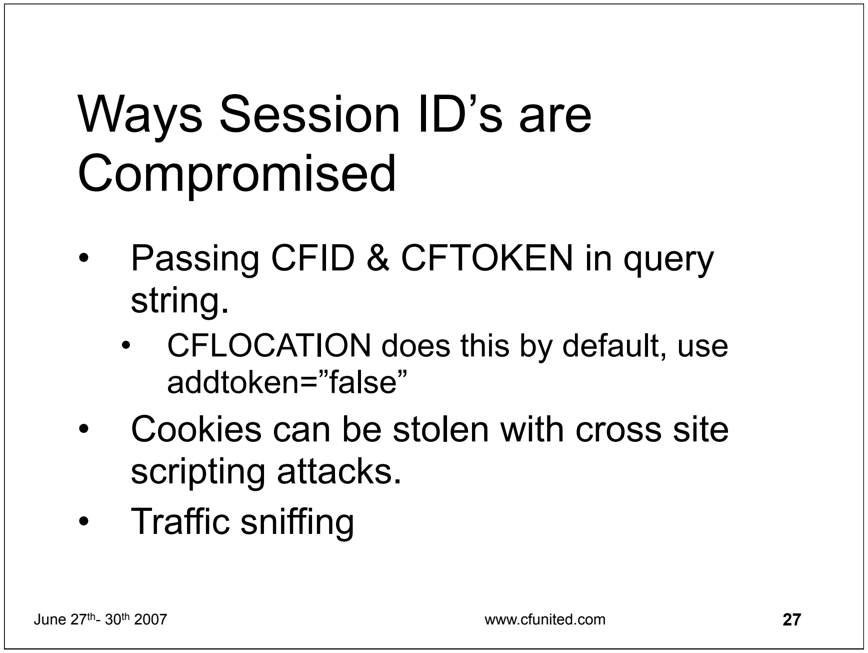
CRLF Injection

- CRLF = Chr(13) & Chr(10)
- CFHEADER

```
<cfheader name=”Content-Type” value=”#url.type#”>
```

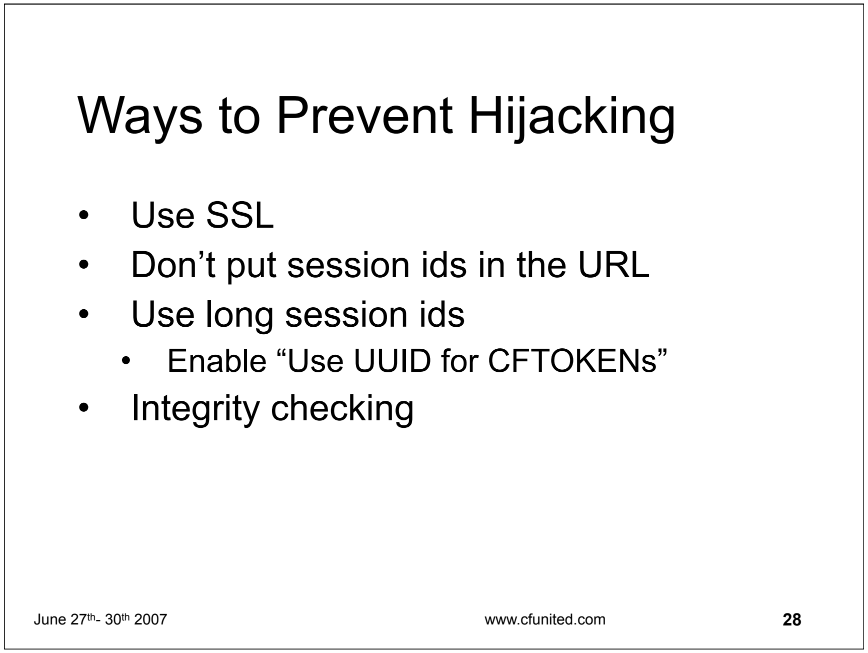
Session Hijacking

- If an attacker knows a user’s session id(s) (CFTOKEN & CFID) they can impersonate the user.



Ways Session ID’s are Compromised

- Passing CFID & CFTOKEN in query string.
  - CFLOCATION does this by default, use addtoken=”false”
- Cookies can be stolen with cross site scripting attacks.
- Traffic sniffing



Ways to Prevent Hijacking

- Use SSL
- Don’t put session ids in the URL
- Use long session ids
  - Enable “Use UUID for CFTOKENs”
- Integrity checking

Don’t Disclose Server Details

- Error messages may show:
  - File Paths
  - Source Code
  - Database Table and Column Names
- Use a Global Error Handler or CFERROR

Require SSL / HTTPS

- Prevent sniffing
- Browsers run at a higher security level lowering success rates on some attacks.
- Secure cookies
  - <cfcookie secure=”true” ...>

In Short: Validate Everything!!

Thanks.

Questions?

www.petefreitag.com
www.foundeo.com