Boosting ColdFusion Application Security

Pete Freitag, Foundeo Inc.

About Me
Pete Freitag

- 25+ Years ColdFusion Experience
- Company: Foundeo Inc.
  - Products: FuseGuard, HackMyCF, Fixinator
  - Consulting: Code Reviews, Server Review, CFML Security Training
- You might also know me from:
  - Lockdown Guides CF9 - CF2025
  - CFDocs.org, cfscript.me, cfbreak.com
  - blog: petefreitag.com
  - twitter/github: @pfreitag

Security Basics

- Unexpected inputs
  - Use Validation, Parameterization, Encoding
- Logic Flaws
  - Code Review

Photo by Jakub Dziubak on Unsplash

Things
To watch out for

- Files and File Paths
- Network Requests (eg cfhttp)
- Auth Logic (use SSO)
- Untrusted input data
- Crypto

Photo by Rob Martin on Unsplash

Quick Wins

- **Validation** - more validation = less vulnerabilities
- **Logging & Monitoring** - know when problems arise
- **Code Scanning** - continuously look for issues, find known vulnerabilities
    - Scan only changed files

Photo by Klim Musalimov on Unsplash

OWASP Top 10

2021 Edition (2025 version not yet released)

1. **Broken Access Control** - IDOR
2. **Cryptographic Failures** - Weak Algo, Key Storage, Incorrect Implementation
3. **Injection** - XSS, SQLi
4. **Insecure Design** - Error Disclosure, Unprotected Credentials
5. **Security Misconfiguration** - Configuration
6. **Vulnerable and Outdated Components** - KEV
7. **Identification and Authentication Failures** - Logic or Weak Implementation
8. **Software and Data Integrity Failures** - Supply Chain
9. **Security Logging and Monitoring Failures** - Insufficient Logging, Log4Shell
10. **Server-Side Request Forgery** - SSRF

DoD photo by Staff Sgt. Luisito Brooks, U.S. Army/Released

AI + Security

- Prompt Injection
- MCP
- Slop squatting
- Denial of Wallet
- Deterministic / Non Deterministic

Auth & Logic Issues

- IDOR
- Missing Auth
- Insecure Password Storage
- Incorrect Auth Logic
- Timing Attacks
- Use SSO when possible - SAML, Entra, Oauth

File Uploads

- Regularly review all your file upload code:
  - Always checks the file extensions of uploaded files against a list of allowed extensions. Use the **`allowedExtensions`** attribute of cffile.
  - Do not upload directly under the web root (at least not before validation)
  - Don’t rely on mime type checks alone, they can be bypassed!
  - Set **`this.blockedExtForFileUpload`** to full list of executable extensions.

Path Traversals

- Happens when you construct a file path with unsafe variables.
- Example:

```
<cfinclude template="html/#url.name#">
```

Path Traversals

- Be careful whenever file paths are constructed dynamically
  - cffile, cfdocument, cfinclude, cfmodule, cfspreadsheet
  - fileOpen, fileRead, fileWrite, etc.
  - cfdirectory, directoryList, etc.

SQL Injection Example

foundeo inc.

```
<cfquery>
  SELECT story
  FROM news
  WHERE id = #url.id#
</cfquery>
```

Risk: Whenever you have a variable inside a SQL Statement

Fixing SQL Injection

foundeo inc.

```
<cfquery>
  SELECT story
  FROM news
  WHERE id = <cfqueryparam value="#url.id#">
</cfquery>
```

Fixing SQL Injection

Lists

foundeo inc.

```
<cfquery>
  SELECT story
  FROM news
  WHERE topic IN (
    <cfqueryparam value="#url.type#" list="true">
  )
</cfquery>
```

SQL Injection
cfqueryparam

- cfsqltype shortcut:
    - You can simply use integer instead of cf_sql_integer
    - The cf_sql_ prefix is no longer required as of ColdFusion 11
- Add maxlength when possible.
- More tips: Mastering cfqueryparam

SQL Injection

With queryExecute

foundeo inc.

```
queryExecute("SELECT story
  FROM news
  WHERE id = #url.id#”);
```

```
queryExecute("SELECT story
  FROM news
  WHERE id = :id”, {id=url.id} );
```

SQL Injection

With queryExecute

foundeo inc.

```
queryExecute("SELECT story
 FROM news
 WHERE id = :id", {
   id={
     value=url.id,
     cfsqltype="integer"
   }
 });
```

SQL Injection
Other Places to Look

- ORMExecuteQuery
- Stored Procedures - if proc builds SQL statements dynamically from inputs
- new Query() - removed as of CF2025
  - Check of fixinator compatibility scanner

SQL Injection

When Parameters Don’t Work

- Places that parameters *may (depending on DB)* not work:
    - ORDER BY clause
    - SELECT TOP n
    - LIMIT / OFFSET
- Validate!
- Use SELECT TOP #int(url.n)#
- Use cfqueryparam whenever you can

Command Injection

- Take care when using cfexecute, or other APIs that may wrap a native command.
- Avoid untrusted variables in the name and arguments.

```
<cfexecute name="c:\bin\tool.exe" arguments="-n #url.n#">
```

Cross Site Scripting (XSS)

foundeo inc.

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

Fixing XSS

Methods to encode variables

foundeo inc.

```
<cfoutput>Hello #encodeForHTML(url.name)#</cfquery>
```

```
<cfoutput encodefor="html">Hello #url.name#</cfquery>
```

Fixing XSS

Picking the correct encoder

foundeo inc.

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

Fixing XSS
Other Techniques

- Encoding is best, but is not always possible:
  - Sanitize HTML: isSafeHTML, getSafeHTML
  - Content-Security-Policy

CSP

Content-Security-Policy

- HTTP Response Header, tells the browser what resources are allowed to load
- Supported on all Modern Browsers for many years
- CF2025 added support for tags that generate script (cfform, etc)
  - Application.cfc: this.enableCspNonceForScript = true;
  - Or enable in CF Administrator
  - Function: getCSPNonce()
- Learn More: content-security-policy.com

CSP Directives

- default-src
- script-src
- style-src
- img-src
- connect-src
- font-src
- object-src
- media-src
- frame-src
- sandbox
- report-uri
- plugin-types (level 2)
- frame-ancestors (level 2)
- form-action (level 2)

Example CSP Headers

- script-src 'self';
- script-src 'self' cdn.example.com;
- script-src 'none';
- script-src 'unsafe-inline';
- default-src 'none'; script-src 'self'

CSP Disables Inline Scripts

- It will not execute:
    - <script>alert('Hi')</script>
    - Hi
    - <a href="javascript:alert('Hi');">Hi
- Workarounds:
    - Use a nonce to allow execution.
    - You can specify: `unsafe-inline` but it defeats most value of CSP
    - Use unsafe-hashes in CSP level 3 to allow JS event handler attributes

XML External Entities (XXE)

- XML Entities: **&entity;**
  - Can make network requests
  - Read Files
- Avoid parsing untrusted XML
- Disable entities: xmlParse(xml, false, **{allowExternalEntities=false}**)

Remote Code Execution (RCE)

- A few different ways this can happen in CFML, most common:
  - Evaluate
  - IIF
  - cfinclude

Fixing RCE

Replace IIF with Ternary Operator

foundeo inc.

```
iif( len(name), de("#name#"), de("Anonymous") )
```

```
len(name) ? name : "Anonymous"
```

Fixing RCE
Fixing Evaluate

foundeo inc.

```
evaluate("url.#name#")
```

```
url[name]
```

Rid your code of evaluate() - terrible for both performance and security

Fixing RCE
Fixing Evaluate

foundeo inc.

```
#evaluate("x+y")#
```

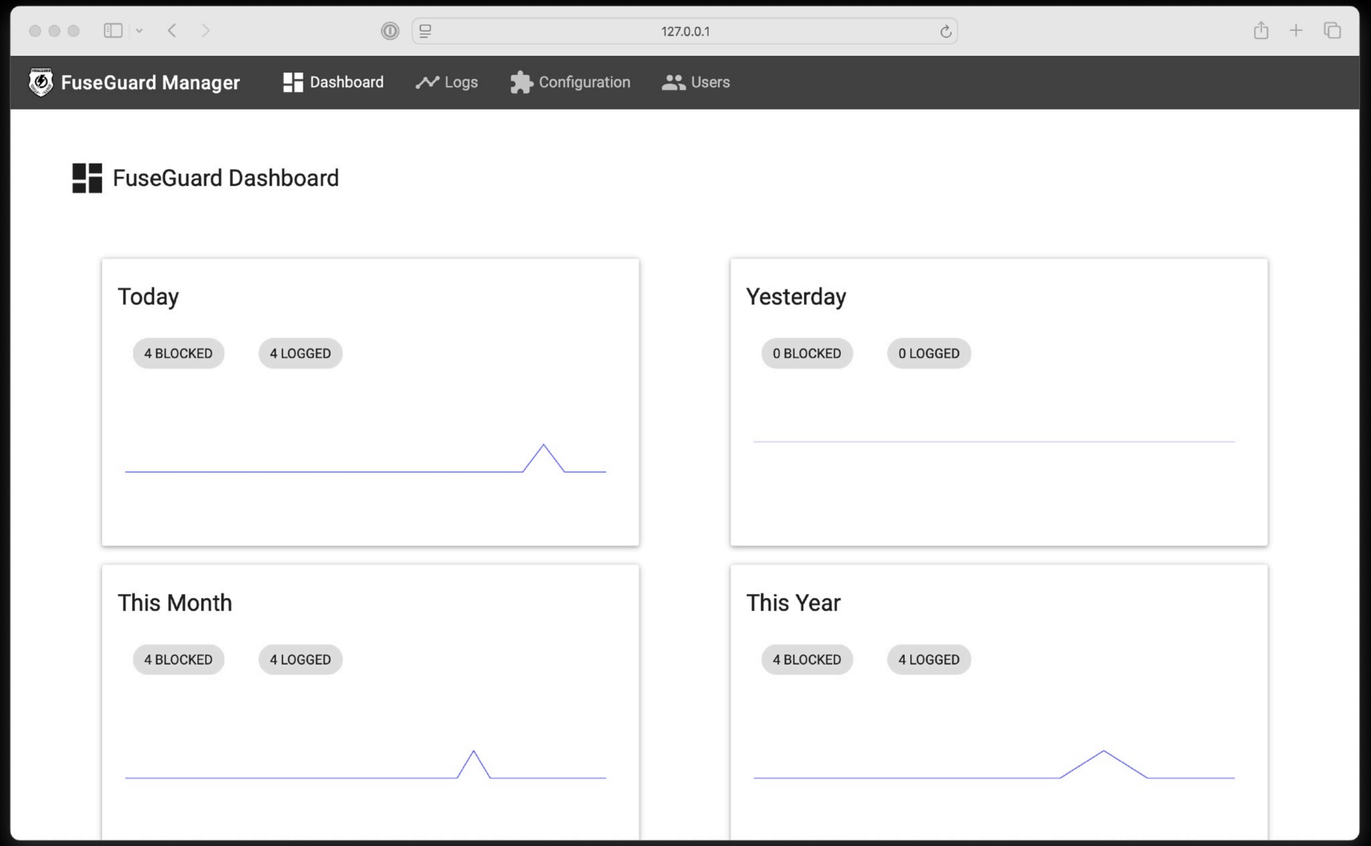
```
#x+y#
```

Rid your code of evaluate() - terrible for both performance and security

FuseGuard

Web App Firewall for CFML

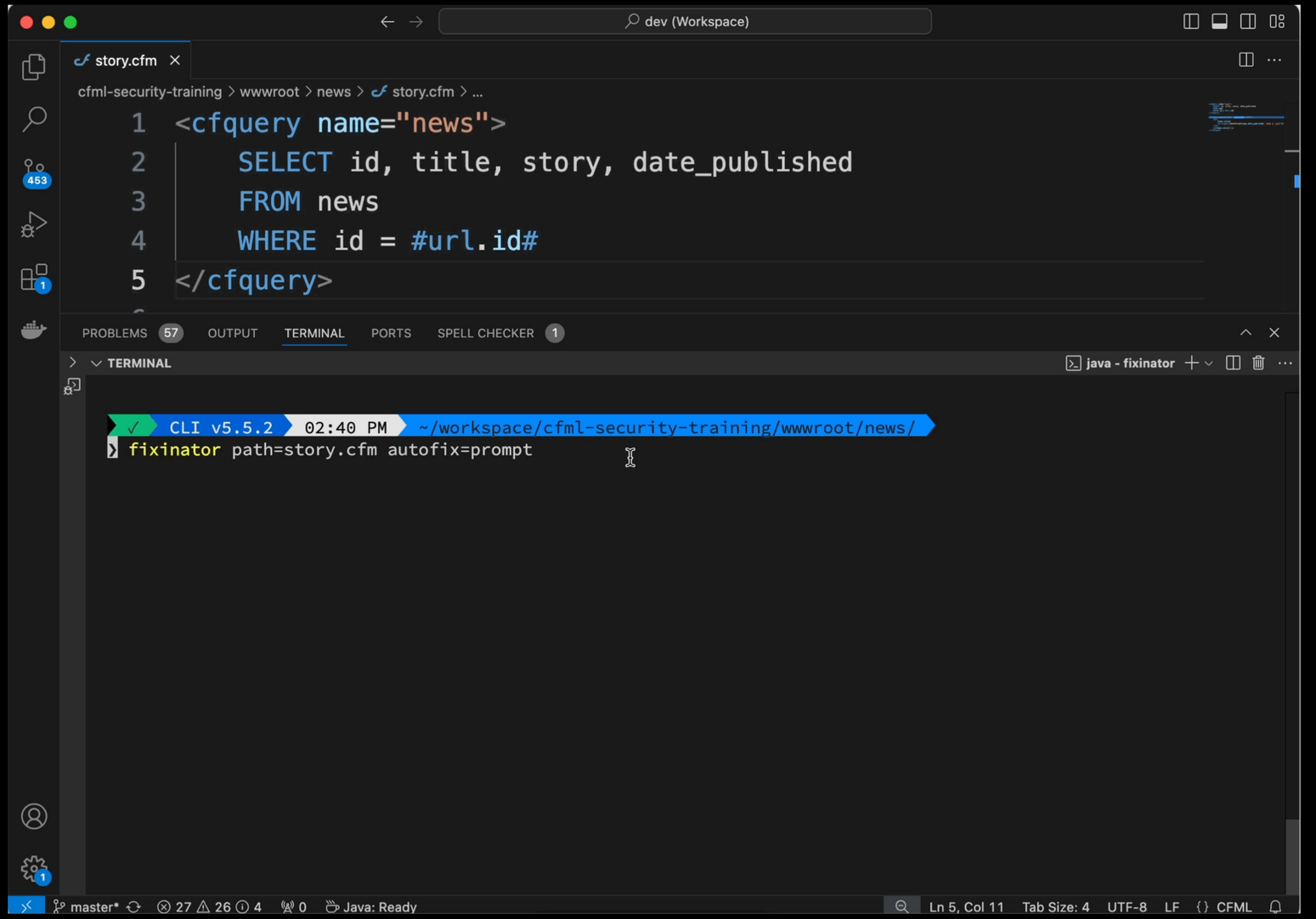
- Blocks Malicious Requests
- Runs onRequestStart



Fixinator

foundeo inc.

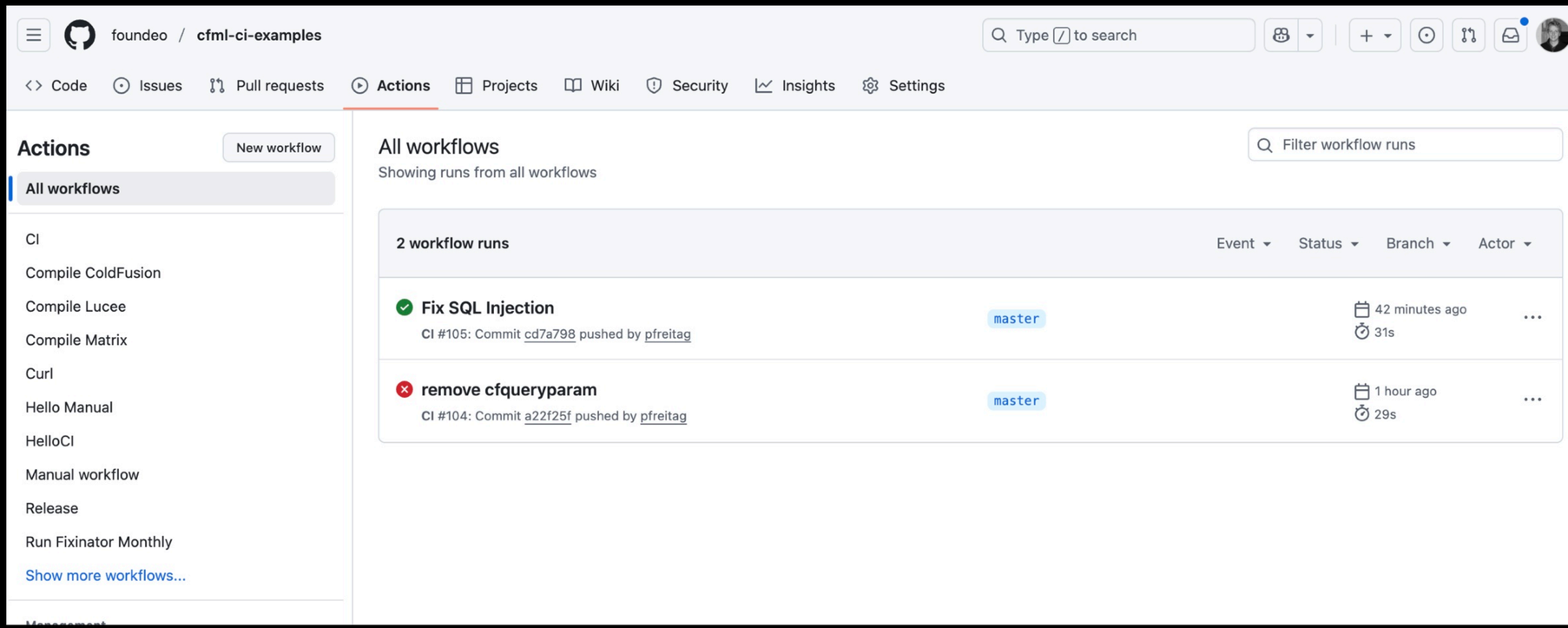
- Fixinator can scan your code and fix certain vulnerabilities



Fixinator

foundeo inc.

- Fixinator supports several CI platforms to scan code



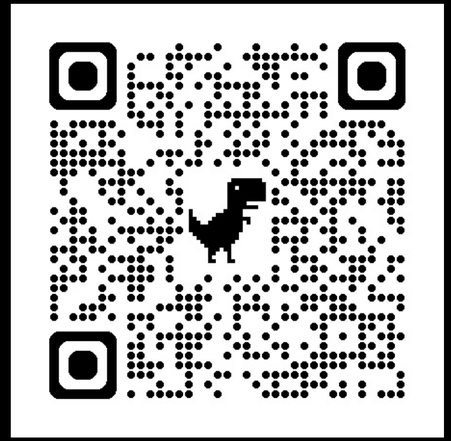
Learn More

foundeo inc.

- ColdFusion Security Guide
  - https://foundeo.com/security/guide/
- Security Training Class (Next Class December 2025)
  - https://foundeo.com/consulting/coldfusion/security-training/
- Ask Me
- Resources: OWASP

ColdFusion Developer Security Training

- December 9-10 2025
- Online Class
- More in-depth coverage of what we looked at today
- Many more Vulnerabilities Discussed
- foundeo.com/consulting/coldfusion/security-training/



Thank You!

foundeo inc.

pete@foundeo.com

Weekly CFML Community Newsletter: cfbreak.com