



Approaches to Secure CFML Code
Pete Freitag, Foundeo Inc.

About Pete

- Guy who wrote the ColdFusion Lockdown Guides CF9-CF2018
- My Company: Foundeo Inc.
 - **Consulting:** Code Reviews, Server Reviews, Development
 - **FuseGuard:** Web App Firewall for CFML
 - **HackMyCF:** Server Security Scanner
 - **Fixinator:** Code Security Scanner
- Blog (petefreitag.com), Twitter (@pfreitag), #CFML Slack
 - I will post these slides on my blog
- Using CFML since late 90s

2020 Security

- Twitter: Accounts of several well known people were hacked in July [\[link\]](#)
- Zoom: 500,000 zoom passwords up for sale in April 2020 [\[link\]](#)
- Microsoft: 250 million customer support logs from misconfigured elasticsearch servers [\[link\]](#)
- MGM Resorts: 10.6 million customer records including names, addresses, dob posted to a hacking forum. [\[link\]](#)
- Tupperware: Hackers added code to checkout page to collect payment info. [\[link\]](#)
- Marriott: 5.2 million customer records including names, addresses, phone numbers, dob. [\[link\]](#)

Takeaways

- We're all impacted
- Even the biggest, wealthiest, smartest companies still have security vulnerabilities.
- Absolute or Perfect Security does not exist
 - And probably never will!
- We can't ignore it

2020 every second



Today we'll look at

Ways to improve security of your ColdFusion apps

Where do I start?

I'm not given time to "improve security"

But you haven't seen my code!

There are too many possible security issues to consider

Our Goal Today

- How to start
- Identify the low hanging fruit
- Prioritize the dangerous stuff / easy wins



**DO YOU HAVE AN
OLD & LARGE
CODEBASE?**



MATURE CODEBASES

- Have **thousands** of source code files
- Has code you hope you don't have to see again.
- Can take weeks, but often months of work to properly secure.
- Can be hard to fix, brittle
- Probably uses outdated techniques

Pick your Game Plan

- **Focus Mode** - Spend several weeks dedicated to identifying & fixing vulnerabilities.
- **Prioritize** - Spend time identifying the most critical vulnerabilities and patch less critical vulnerabilities as you see them.
- **As you go** - As you work on files fix vulnerabilities as you see them. You may miss some vulnerabilities with this approach.
- **Hire Someone to Find or Fix issues** - Can work well when you are too busy to find or fix issues.

How Do You Start?

Step 1: Delete the code!



You Might Have a Lot Of Code that Never Runs



OLD CODE OFTEN FULL OF SECURITY HOLES

Most of that code may never run

You Might Be Using...

“Home Made Version Control”

- index_2.cfm
- index.old.cfm
- index-backup.cfm
- index-2007-03-04.cfm
- index-copy.cfm
- folder_backup2009/

Version Control

- Those backup folders and files are probably full of vulnerabilities.
- Version Control Server keeps backups of all your code and all changes you have ever made to it.
- Sync server source code with version control.
 - Identify if someone changed something on the server.

Version Control

- Spend some time to identify unused code.
- Delete it!
- Version control has your back, if you deleted something you can recover it from the repository.

“There are Lots of Fads in Software Development, Version control is not one of them.”



@levelsio

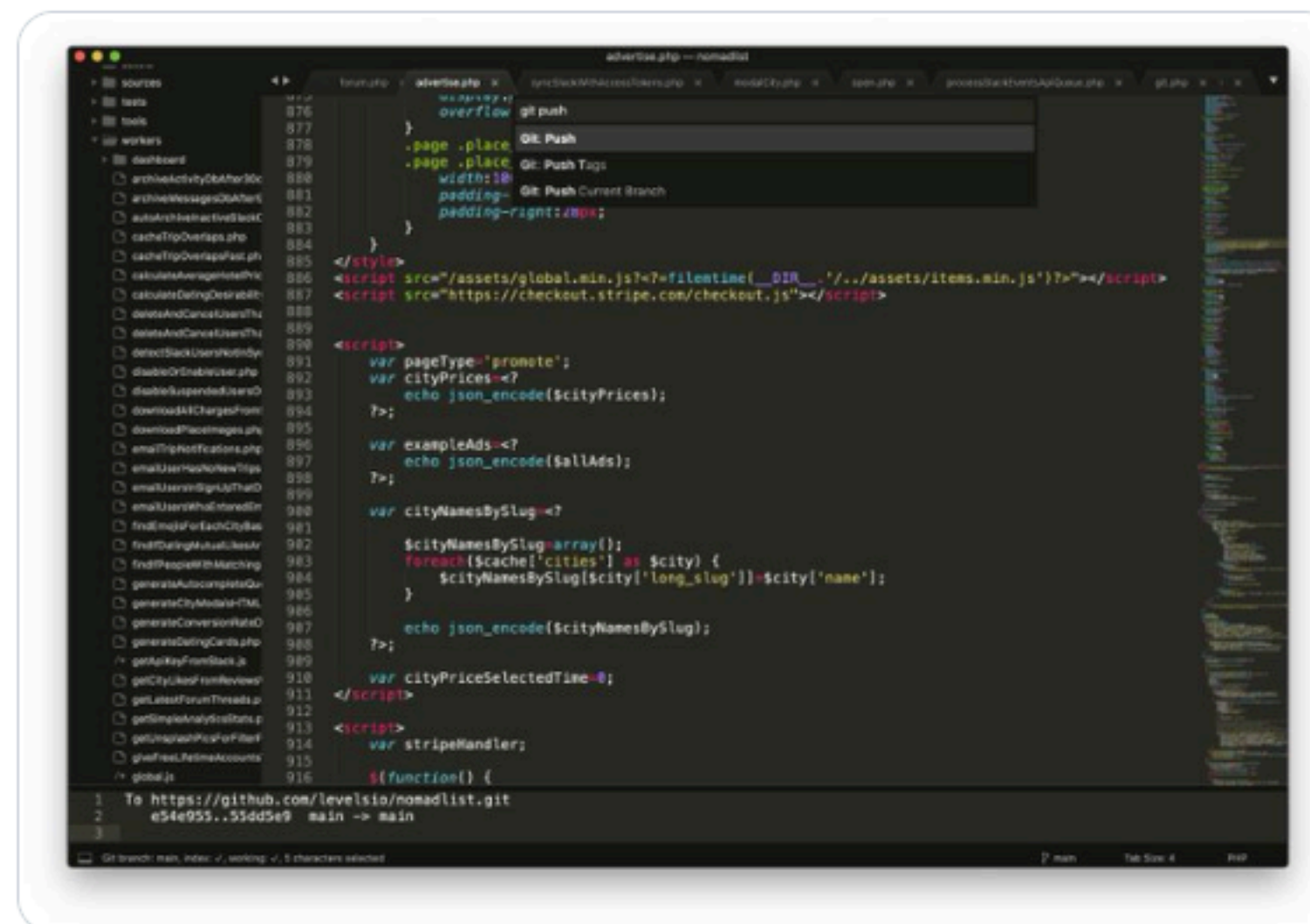


I got Git integration now in Sublime Text, I feel like a wizard now 🧙

I edit my code, then inside editor I push to server and it's live on the site a second later

Twitter: we been telling you this for 5 years

Me: true, sry lol



9:43 AM · Nov 18, 2020 · Twitter Web App

77 Likes



@levelsio · 6m



Replying to @levelsio

I never understood it because nobody told me this was possible, I always thought I had to write git commit -m in the CLI and it seemed inferior to FTPing. If anybody told me I could edit my code, and upload to server from my editor....

Ways To Identify OBSOLETE Code

- Unix / Linux / Mac

```
$> find /wwwroot/ -mtime +365
```

- Windows

```
C:\>forfiles -p "C:\web" -s -m *.* /D -365 /C  
"cmd /c echo @path"
```

Ways To Identify OBSOLETE Code

- Unix / Linux / Mac

```
$> find /wwwroot/ -atime +365
```

- The `atime` (last accessed time) timestamp may be disabled on your server for performance (if drive was mounted with `noatime` flag).
- RHEL mounts drives with the `relatime` flag by default, which is not real time but may be sufficient for these purposes.

Patch That Server

- Use ColdFusion 2016* or greater. CF11 Core Support Ended Apr 2019, CF10 Ended May 2017, CF9 have had no security patches for many many years.
- Windows 2008 (EOL 2015)
- Java 8+, Java 7 (EOL 2015), Java 6 (EOL 2013)



* Core Support for CF2016 Ends February 2021

Patch That Server

- Multiple Denial of Service Vulnerabilities in old versions of Java
- Path Traversal via Null Byte injection JVM (< 1.7.0_40)
- CRLF Injection (CF10+)
- File Uploads “somewhat” more secure (CF10+)
- TLS / SSL Protocol Implementations
- Java 8 Not supported on CF9 and below
- Use HackMyCF to help keep you on top of all this

Lockdown The Server

- What user is the JVM running as?
 - If your CFML server is running as SYSTEM or root then the attacker can do a lot more harm.
- What permission does the user have?
 - If CFML server user only has readonly access to web root and CFML server install directory then less harm can be done (easily).
 - Does CFML server need full write access to web root? or just one or two directories?

**“Nearly 60% of Breaches due to
Un-patched Vulnerability”**

ServiceNow Survey

Equifax Breach

- The Equifax breach was caused by using a vulnerable java library: Apache Struts
 - Struts was patched on March 7th 2017
 - Equifax discovered breach on July 29th 2017
 - Equifax applied the patch on July 30th, 2017

One year after the Equifax breach:

"As many as 10,801 organizations—including 57% of the Fortune Global 100—have downloaded known-to-be-vulnerable versions of Apache Struts"

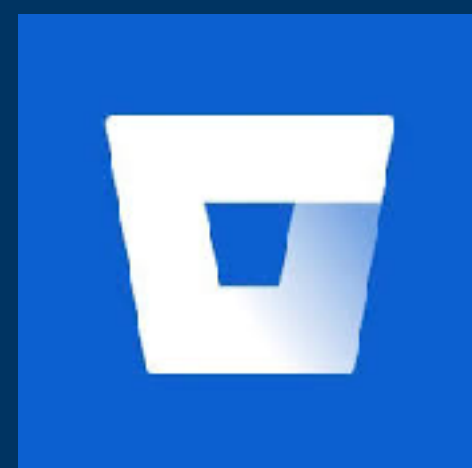
Source: <http://fortune.com/2018/05/07/security-equifax-vulnerability-download/>

UPDATE Known Vulnerable Components

- **Fixinator** - (CFML, JS, JAR) Looks for known vulnerable CFML libraries (eg FCKeditor file upload vulnerability, old custom tags, etc) [commercial]
- **OWASP Dependency Check** - (Java, C, Ruby, Python, NodeJS)
- **RetireJS** - (JS)
- **npm audit** - (JS)

Continuous Security

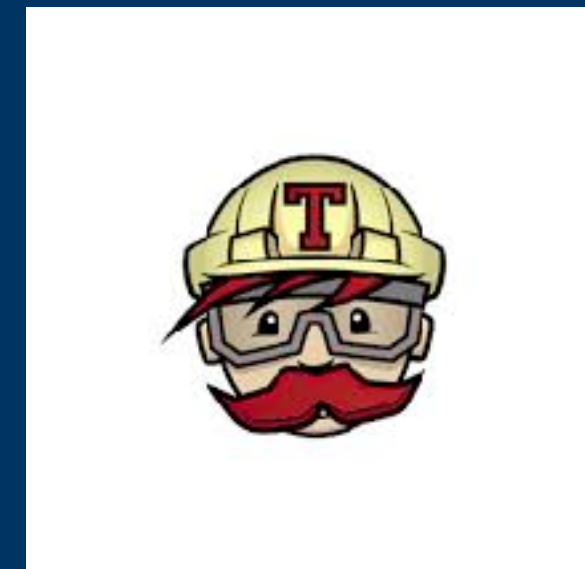
You need Version Control



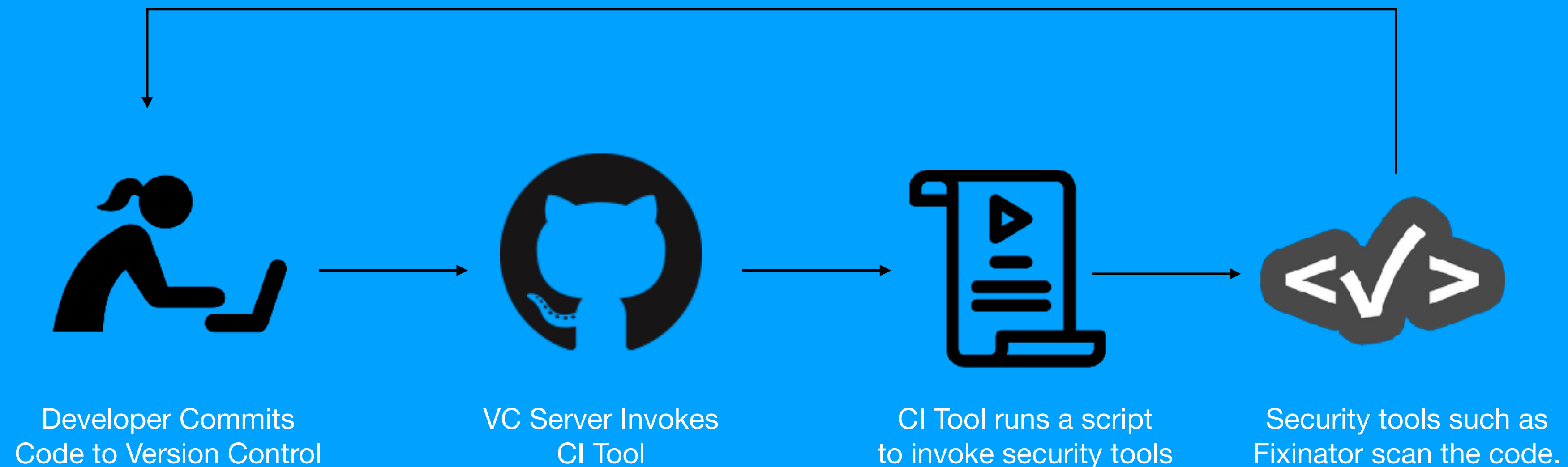
A CI Platform



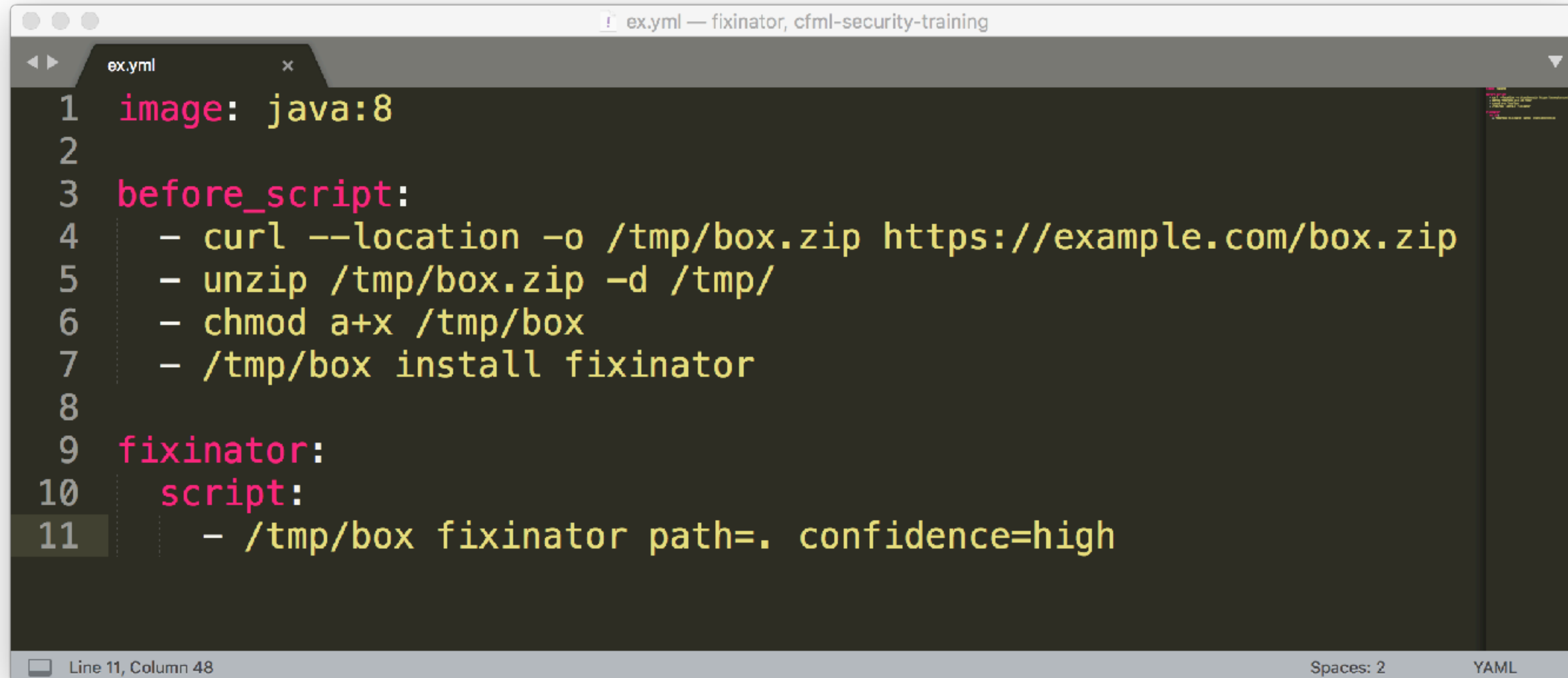
or



Continuous Security Workflow



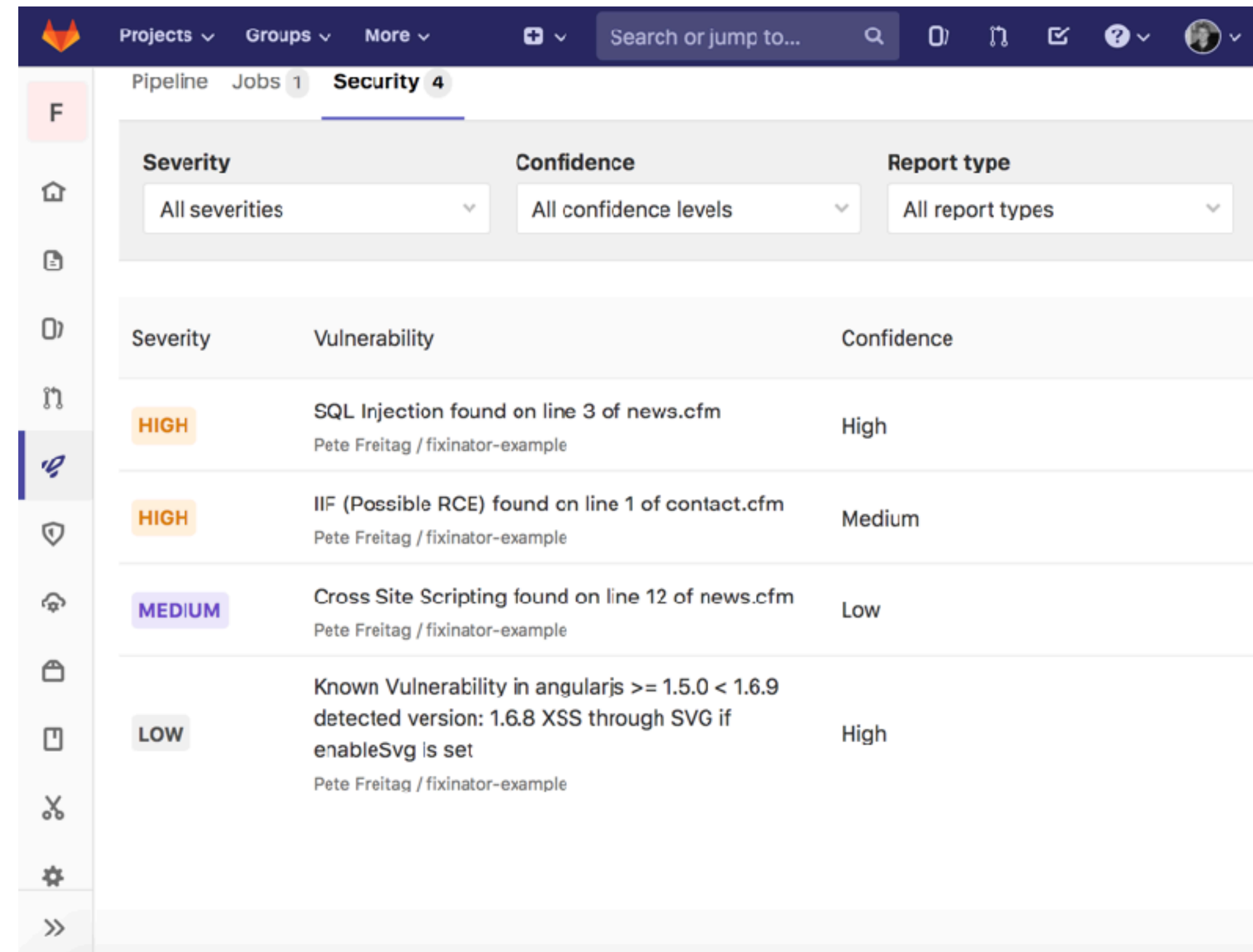
STOP, YAML TIME



```
ex.yml — fixinator, cfml-security-training
ex.yml
1  image: java:8
2
3  before_script:
4    - curl --location -o /tmp/box.zip https://example.com/box.zip
5    - unzip /tmp/box.zip -d /tmp/
6    - chmod a+x /tmp/box
7    - /tmp/box install fixinator
8
9  fixinator:
10  script:
11    - /tmp/box fixinator path=. confidence=high
Line 11, Column 48 Spaces: 2 YAML
```

This is a script that tells the CI pipeline what tools to use.

GitLab Example



The screenshot shows the GitLab Security interface. At the top, there are navigation tabs for 'Pipeline', 'Jobs 1', and 'Security 4'. Below these are three filter dropdowns: 'Severity' (set to 'All severities'), 'Confidence' (set to 'All confidence levels'), and 'Report type' (set to 'All report types'). The main content is a table with four rows of vulnerability data.

Severity	Vulnerability	Confidence
HIGH	SQL Injection found on line 3 of news.cfm Pete Freitag / fixinator-example	High
HIGH	IIF (Possible RCE) found on line 1 of contact.cfm Pete Freitag / fixinator-example	Medium
MEDIUM	Cross Site Scripting found on line 12 of news.cfm Pete Freitag / fixinator-example	Low
LOW	Known Vulnerability in angularjs >= 1.5.0 < 1.6.9 detected version: 1.6.8 XSS through SVG if enableSvg is set Pete Freitag / fixinator-example	High

➡ <https://gitlab.com/pfreitag/fixinator-example/pipelines/>

SHIFT LEFT

A DevOps term:

Requires continuous integration / testing

Developers find / fix problems early in development

Yields higher quality software, better development

It's easier to fix a
bug the same
day you wrote the
code that caused it

GETTING STARTED WITH CI

- Michael Born's Course: Five days of CI with ColdFusion and Bitbucket: <https://learncf.teachable.com/>
- Fixinator Continuous Security Guides: <https://github.com/foundeo/fixinator/wiki/Continuous-Integration-Guide>
 - How to setup CI on GitLab, Bitbucket, TravisCI, Azure DevOps, AWS CodeBuild, CircleCI, and Jenkins.
 - Example: <https://gitlab.com/pfreitag/fixinator-example/pipelines/>
- My CI Presentation ITB2020: <https://www.petefreitag.com/item/902.cfm>

Delegate Risks

Delegate Risky Areas

When possible

- Payment Processing
 - Delegate to payment gateway hosted payment page
 - JS SDK (Braintree, Stripe, etc), may lessen PCI compliance requirements
- Authentication
 - Enterprise: Active Directory, LDAP, etc
 - Social: Google, Apple, Facebook, Twitter, MS, etc.
- Other Areas: Encryption, Secrets, Key Management

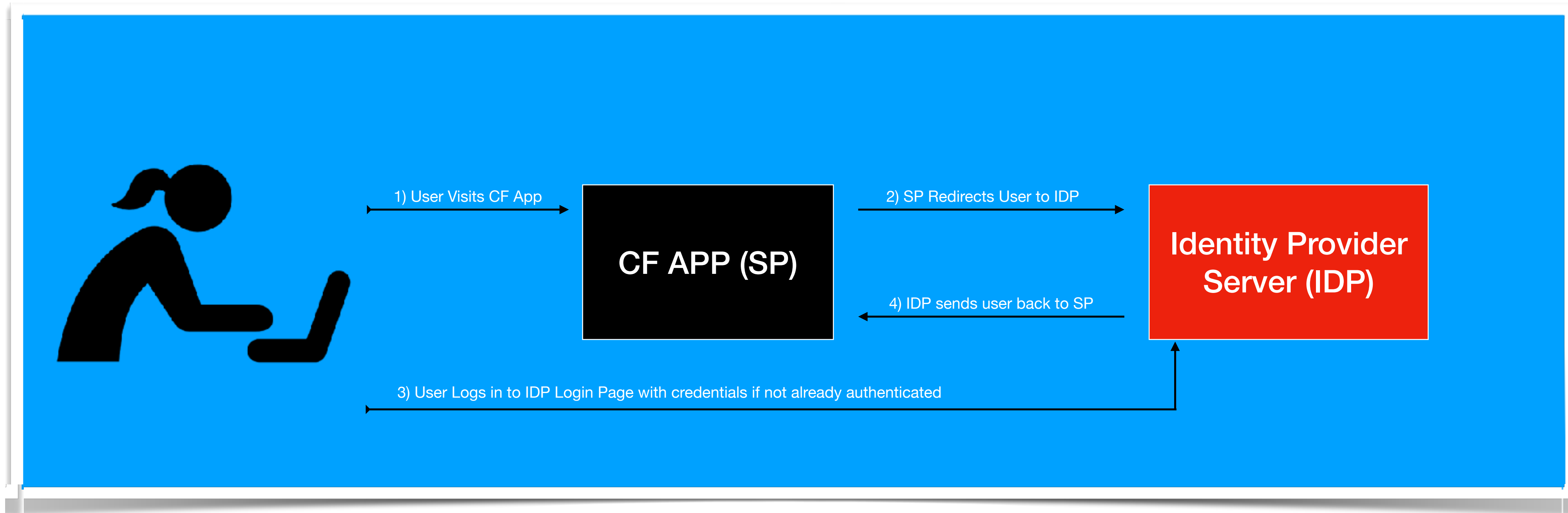
Authentication

Single Sign On (SSO)

SAML, OAUTH, etc

- DELETE your authentication code!
 - DIY Auth code notoriously riddled with security weaknesses
 - Eliminate storing passwords, forgot passwords, etc.
- Users will notice and will like it.
 - Users do not need separate passwords for your app
- Both a Security Improvement and User Experience Improvement

How does SAML work?



SSO Implementation

- ColdFusion 2021 SAML Integration
 - `InitSAMLAuthRequest()`, `ProcessSAMLResponse()`,
`InitSAMLLogoutRequest()`
 - <https://helpx.adobe.com/coldfusion/using/saml-coldfusion.ug.html>
- Java Libraries such as java-saml, or opensaml
- Web Server Extensions act as a facade, provide authenticated user id as CGI variable.

Implement a WAF

- Inspect HTTP Request or Response
 - Block or log malicious requests
 - Provides Defense in Depth
- Several options
 - Hardware Based
 - Software Based / Application Level
 - FuseGuard

How Do you Start

Step 2: Identify high risk vulnerabilities in your code.

HIGH RISK Vulnerabilities

- File Uploads
- Remote Code Execution / Dynamic Evaluation Issues
- SQL Queries (SQL Injection)
- File System Access / Path Traversals
- Dynamic Process Execution (CFEXECUTE)
- Anything that can fully compromise server

How I Classify Vulnerabilities

**Compromises
the server(s) directly**

**Compromises
users**

Both are important but where do you start?

How I Classify Vulnerabilities

**Compromises
the server(s) directly**

Examples:
SQL Injection
File Upload / Access
Remote Code Execution

**Compromises
users**

Examples:
XSS
CSRF
Session Hijacking

Both are important but where do you start?

A low-angle shot of a hot air balloon with a vibrant rainbow color scheme (red, orange, yellow, green, blue, purple) against a clear blue sky. The balloon's canopy is fully inflated and its suspension lines are visible, converging towards the top center.

Evaluate

Remote Code Execution Via

Common Legacy Evaluate

```
<cfset day_1 = "Monday">  
<cfset day_2 = "Tuesday">  
<cfset day_3 = "Wednesday">  
  
<cfoutput>  
    #Evaluate("day_#url.day#")#  
</cfoutput>
```

Evaluate Example

Fixing Legacy Evaluate Example

```
<cfset day_1 = "Wednesday">
<cfset day_2 = "Thursday">
<cfset day_3 = "Friday">

<cfoutput>
    #variables["day_#url.day#"]#
</cfoutput>
```

Fixing Evaluate Issues

- Search Code for "Evaluate"
- In most cases you should not need to use Evaluate at all, use brackets.
 - If the variable is a query you may need to use `queryName[row][columnName]` notation.
 - Not all cases are super simple to fix, but most are.
- Remove all Evaluate calls from your code.
- Also look at PrecisionEvaluate

Do Any other Functions Evaluate Dynamically?



```
Hi #iif(len(url.name) EQ 0, de("Friend"), de(url.name))#
```

The second and third arguments are evaluated dynamically!

IIF Example

Fixing IIF

```
Hi #(!len(url.name)) ? "Friend" : url.name#
```

ELVIS OPERATOR (CF11+)

```
Hi #url.name?:"Friend"#
```

Elvis Operator tests to see if url.name is defined / not null



File Uploads

Common Yet Dangerous

File Uploads

3 CORE Rules

File Uploads



Never trust a MIME!

File Uploads Rule #1

- CF10 added strict attribute to cfile action=upload
 - Instead of validating the MIME type that the browser sends it validates the the file content (eg fileGetMIMEType()).
 - Can we still get around this?

FILE UPLOADS Rule #2

- Always validate the **file extension**
 - CF10+ allows you to specify file extensions in **accept** attribute
 - You can also specify a file path in the destination with a hard coded extension

```
<cfile action="upload" accept=".png,.jpg">
```

```
<cfile action="upload" destination="/path/to/#int(some.id)#.pdf">
```

FILE UPLOADS Rule #3

- The upload **destination** must be outside of the web root

```
<cffile action="upload" destination="#expandPath("./uploads/")#">
```

```
<cffile action="upload" destination="#getTempDirectory()#">
```

File Uploads

- Inspect file content: fileGetMimeType, isImageFile, isPDFFile, etc
- Upload to static content server (s3 for example)
 - Upload directly to s3: <https://www.petefreitag.com/item/833.cfm>
- Make sure directory serving uploaded files cannot serve dynamic content.
- File Extension Allow List on Web Server (eg IIS Request Filtering)
- secureupload.cfc: <https://github.com/foundeo/cfml-security/>

New File Upload Features

- New in CF2018 update 3, CF2016 update 10 & CF11 update 18
- Application.cfc setting: `this.blockedExtForFileUpload`
 - Comma separated list
 - Set to "*" to block all (empty string allows all)
- Set server wide in ColdFusion Administrator



Path Traversal

File System Access &

Path Traversal

```
<cfinclude template="path/#fileName#">
```


Path Traversal example

Fixing Path Traversals

- Avoid variables in paths
 - If you really need to use a variable strip out everything except a-z0-9
- Use the CF11+ Application.cfc setting `this.compileExtForInclude` setting.

Finding FILE ACCESS ISSUES

- As you can see any code that accesses the file system can potentially be exploited.
- Review all function calls / tags that access file system
 - cfile, cfdocument, cfinclude, cfmodule, cfspreadsheet
 - fileRead, fileWrite, fileOpen, etc



SQL Injection

It still happens

Classic SQL Injection

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

news.cfm?id=0;delete+from+news

Fixing SQL Injection

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

Script Based

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

Vulnerable

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

Not Vulnerable

Finding SQL Injection

- Search codebase for `cfquery`, `queryExecute`, `ormExecuteQuery`, `new Query()`
- Use Static Code Analyzer (CFBuilder 2016+)
- Fixinator can find, and fix them for you (quick demo)
- Fix when you see one as you work

Securing Legacy CFML

**Step 3: Fix Additional vulnerabilities
in your code.**

Scope Injection

Taking advantage of scope cascading in CFML

- Suppose you have a variable `session.userID`
 - Now assume the variable is not yet defined in the `session` scope (user not yet logged in)
 - If I request `/admin/page.cfm?session.userID=1`
 - CF will check `session` scope first, but if not defined will check all other scopes: `url`, `form`, `cookie`, etc.
 - Here `url.session.userID` is defined and would be used
- Demo

Preventing Scope Injection

- Can happen with any variable that is not defined, not just `session`
- Ensure that variables are properly defined
 - Set defaults in `onSessionStart`, `onApplicationStart`
 - Use `structKeyExists`, or `session.keyExists("userID")`
- Application.cfc setting `this.searchImplicitScopes = false`
- Learn more: <https://www.petefreitag.com/item/834.cfm>

What's Next

- Session Handling (sessionRotate, sessionInvalidate)
- Authentication / Authorization / Forgot / Remember Me Code
- Cross Site Scripting
 - CF2016 `<cfoutput encodefor="html">`
- Cross Site Request Forgery
- Timing Attacks
- And More

Where can I learn more

About Web Application Security?

- [OWASP.org](https://owasp.org) - tons of info about web application vulnerabilities
- Foundeo Security Training Class - 6-8 hour class goes more in depth on everything we discussed today.
- <https://foundeo.com/consulting/coldfusion/security-training/>

Thank You!

Questions?

Pete Freitag pete@foundeo.com
or [@pfreitag](https://twitter.com/pfreitag) on Twitter

foundeo

foundeo.com | fuseguard.com | hackmycf.com | fixinator.app