



Best Practices for Securing your Secrets and Settings in ColdFusion

Presented by Gavin Pickin
Adobe CF Summit 2023

<https://github.com/gpickin/2023-cf-summit-secrets-and-settings>

Who am I?

- Software Consultant for Ortus Solutions
- Work with ColdBox, CommandBox, ...Box every single day
- Working with Coldfusion for 25 years
- Working with Javascript just as long
- Love learning & sharing the lessons learned
- From New Zealand, live in Bakersfield, Ca
- Loving wife, lots of kids, and countless critters

@gpickin on twitter

<http://www.ortussolutions.com>





MODERNIZEORDIE
CFML NEWS EDITION

Episode 151 - June 7th, 2022



BLOGS, TWEETS & VIDEOS
O F T H E W E E K

Watch Now



Search



2022 VS Code Hint Tip and Trick of the Week

Beginner



As seen on the CFML News Podcast - Your Hosts will show off a VS Code Hint Tip and Trick of the week.

8 Videos

24 minutes



2022 ForgeBox Modules of the Week

Beginner



As seen on the CFML News Podcast - Your Hosts highlight a new ForgeBox Module of the Week

7 Videos

29 minutes



Publish Your First ForgeBox Package

Beginner



Learn alongside Gavin Pickin, how to publish your first ForgeBox package.

14 Videos

1 hour 16 minutes

Thank You Sponsors!



HOSTEK



Other Awesome Ortus Presentations!

Battlefield ORM : Learn the strategies and tactics to win with ColdFusion ORM powered by Hibernate!

10/3 @ 9am in Grand Ballroom D

Luis Majano



Automating CFML from the CLI with Task Runners

10/3 @ 4:15pm in Grand Ballroom BC

Brad Wood



Reason I submitted this topic for CF Summit

- Handling credentials, secrets and settings is a crucial aspect of any project.
- Developers must ensure that sensitive data is kept safe and secure from unauthorized access.
- Ensuring safety shouldn't compromise local development convenience.
- Essential to adopt an approach that provides both security and ease of use.
- A lot of my Client Work involves improving procedures and workflows to ensure developer productivity and efficiency while maintaining security.

Goals of this talk

- Discuss some of the Problems related to settings and secrets / credentials
- Discuss some solutions to these problems
- Discuss pros and cons of these solutions
- Discuss the decision process for selecting your solution to these problems
- Identify tools to help you on this journey

There is no RIGHT way

- There are always pros and cons with each decision you make
- The choices you make should take those pros and cons for you, your team, your company and your environment
- Just because one decision is right for one person or company, doesn't mean it is not right for you

BUT looking at the reasons why they made the decision, will help you decide what is best for you.

Hopefully I give you some reasons and tools to make your life easier and more secure!

Who can this talk help?

I think the lessons in this talk can help everyone.

- No matter what type of development setup you have
- No matter what type of deployment strategy you have.

Types of Development Setups

- We develop on Production Server
- We develop on a shared Dev server
- We develop on a local Dev server

Why do we not develop on Production?

- We were bitten by bad experiences
- So we changed our practices to reduce the risk & pain
- From something which was pretty developer friendly?!

Types of Deployment Setups

- No need to push code, we're on the server already
- Copy files from one folder to another
- FTP or Rsync files from one machine to another
- Push to source control and manually pull files to the server
- Push to source control with an automated pull to the server
- Push to source control, build files with CI and deploy the site over the existing site
- Push to source control, build files with CI and deploy the site to a new folder and update Web Server to point to it
- Manual or Automated Docker image creation and deployment.

Why change Deployment Setups?

We have been bitten by bad experiences, or we are smart and want to avoid bad experiences others have been bitten by:

- Deployed a file too soon
- Not updating all of the files needed for an update
- Removing some files, but not all files needed to be removed
- When updating files, file locks blocked a file update
- While replacing files, the traffic hitting the site caught us when a file was in the middle of being written and errored

Big Pro for early versions of ColdFusion

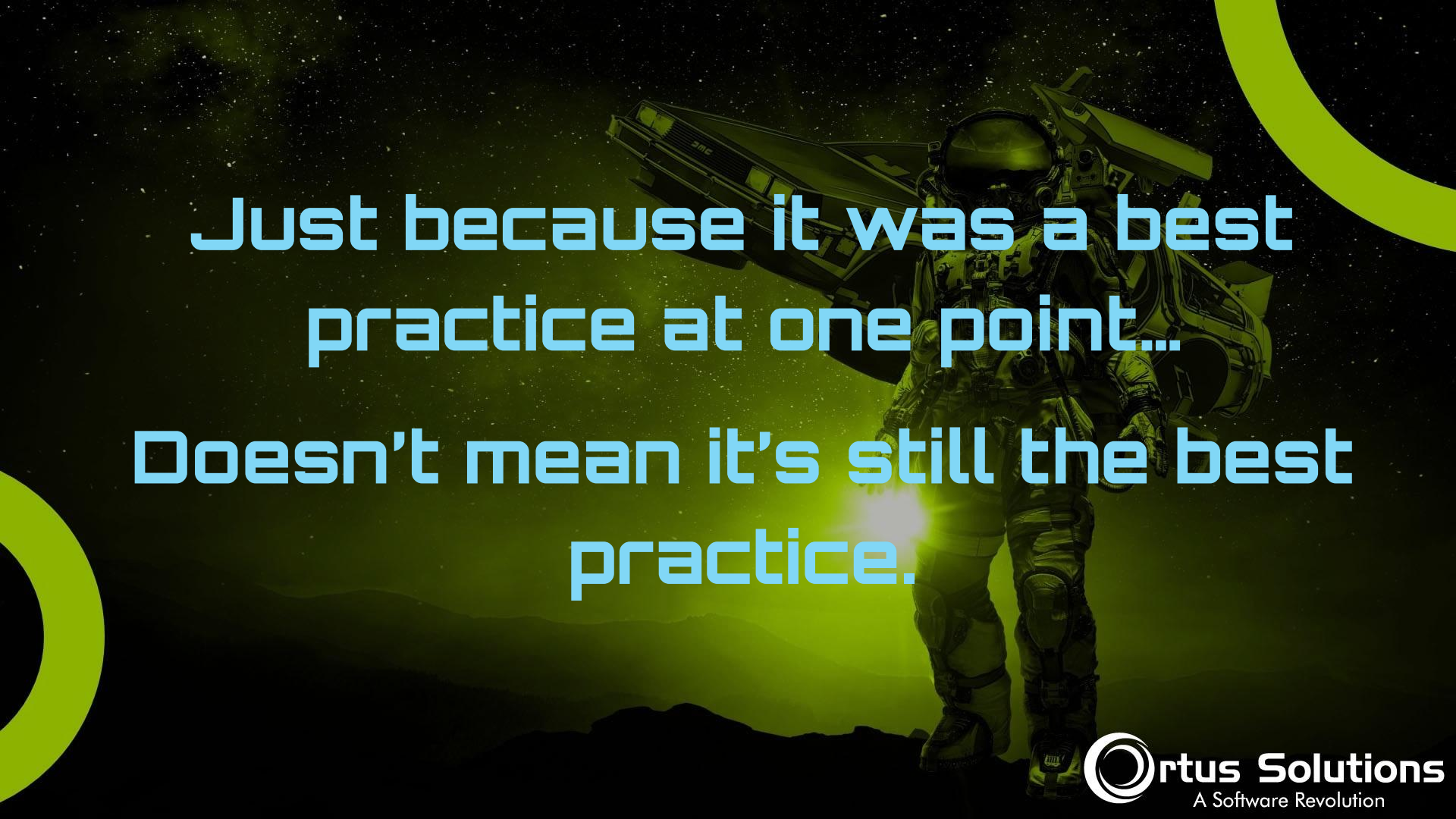
Separating your
CODE
from your
CONFIGURATION

Evolution of Settings and Secrets



In the good old day, Settings are configured through the ColdFusion Administrator

- This is one of the main reasons I fell in love with ColdFusion
NO MESSY CONNECTION STRINGS ALL OVER MY CODE
- No SMTP Mail Settings in my Code, I could set that up 1 time in my server, and it was done.
- Only those with access to ColdFusion admin could add or edit those settings
- They were all in one location



**Just because it was a best
practice at one point...
Doesn't mean it's still the best
practice.**

Remember - Best Practices - Avoid Risks

An astronaut in a full space suit is floating in space, holding a long, thin object that looks like a telescope or a tool. The background is a dark, starry space with a bright light source on the right, creating a lens flare effect. The astronaut is positioned on the right side of the slide, partially overlapping the text.

What happens when Steve adds a new Datasource to his local CF Server, and pushes the code, but it breaks on Production because no one added the datasource to Production. It was in the email he sent.

What happens when someone adds a cache to their local server, but your app crashes locally because you didn't know they added a new cache. You yell at Mary, but she added it to the readme file.

Welcome to Configuration as Code

A full-page background image featuring an astronaut in a white spacesuit floating in space. In the background, a silver classic car is visible, appearing to be in orbit or floating. The scene is set against a dark, starry space background with a bright light source creating a lens flare effect. Large, semi-transparent blue circular shapes are positioned in the top right and bottom left corners.

There is a great blog post on Configuration as Code, I'm going to highlight a few points from it on the next few slides, read the entire thing here

<https://www.browserstack.com/guide/configuration-as-code>

Why use Configuration as Code

- Scalability
- Standardization
- Traceability
- Increased Productivity

Why use Config as Code: Scalability



- Handling configuration changes as code, like IaC, enables teams to create, update, and maintain config files from a single centralized location while leveraging a consistent deployment approach.
- For instance, you require configuration files for each storage option if you are developing USB devices.
- You may create thousands of configurations by combining these files with the required software.
- To handle these variations, you need a robust, centralized source control that can be accessed from different levels in your CI/CD pipeline.

Why use Config as Code: Standardization



- When the configuration is written like source code, you can use your development best practices, such as linting and security scanning.
- Before they are committed, config files must be reviewed and tested to guarantee that modifications adhere to your team's standards.
- Your configurations can be maintained stable and consistent via a complicated microservices architecture.
- Services function more effectively together when a set process is in place.

Why use Config as Code: Traceability



- Setting up configuration as code requires version control.
- You require a robust system that can conveniently save and track changes to your configuration and code files.
- This could improve the level of quality of your release.
- You can locate its source if a bug still slips through and rapidly identify/fix an issue by comparing the versioned configuration files.

Why use Config as Code: Increased Productivity



- You may streamline your build cycle by turning configurations into managed code. Both IT and end users are more productive as a result.
- Your administrators may incorporate everything into a release or build from a single version control system.
- Developers are confident in the accuracy of their changes because every component of your workflow has been tested in concert.

How do we Implement Config as Code

- Admin API
- ColdFusion Application.cfc
- Adobe's CFSetup Configuration Tool
- Adobe's new Product: CCS
- Ortus Solutions' CFConfig tool























Adobe CF Admin API

Apart from modifying Administrator settings from the ColdFusion Administrator portal, you can also change the settings using the Admin APIs.

ColdFusion provides the APIs that are located in the location (<CF Directory>\cfusion\wwwroot\CFIDE\adminapi).

<https://helpx.adobe.com/coldfusion/configuring-administering/coldfusion-administrator-api-reference.html>

Admin API

	_datasource	5/20/2016 3:27 PM	File folder	
	_servermanager	5/20/2016 3:27 PM	File folder	
	customtags	5/20/2016 3:27 PM	File folder	
	accessmanager.cfc	5/20/2016 3:27 PM	CFC File	27 KB
	administrator.cfc	5/20/2016 3:27 PM	CFC File	218 KB
	Application.cfm	5/20/2016 3:27 PM	CFM File	3 KB
	base.cfc	5/20/2016 3:27 PM	CFC File	44 KB
	collections.cfc	5/20/2016 3:27 PM	CFC File	36 KB
	datasource.cfc	4/25/2017 7:34 PM	CFC File	445 KB
	debugging.cfc	4/25/2017 7:34 PM	CFC File	214 KB
	document.cfc	5/20/2016 3:27 PM	CFC File	53 KB
	eventgateway.cfc	4/25/2017 7:34 PM	CFC File	99 KB
	extensions.cfc	4/25/2017 7:34 PM	CFC File	185 KB
	flex.cfc	5/20/2016 3:27 PM	CFC File	14 KB
	mail.cfc	5/20/2016 3:27 PM	CFC File	59 KB
	office.cfc	5/20/2016 3:27 PM	CFC File	30 KB
	runtime.cfc	6/14/2016 7:38 PM	CFC File	267 KB
	scheduler.cfc	5/20/2016 3:27 PM	CFC File	30 KB
	security.cfc	4/25/2017 7:34 PM	CFC File	432 KB
	serverinstance.cfc	5/20/2016 3:27 PM	CFC File	35 KB
	servermonitoring.cfc	5/20/2016 3:27 PM	CFC File	479 KB
	websocket.cfc	5/20/2016 3:27 PM	CFC File	16 KB

Application.cfc

The Application.cfc allows you to set some more common Settings without logging into the administrator, including but not limited to the following:

- Mappings
- Datasources
- CustomtagPaths
- applicationTimeout
- Cache.querysize
- clientStorage
- sessionTimeout
- scriptProtect

What about all the other settings?

Some tags allow you to set settings inline.

For example, the CFMail tag, which allows arguments of username, password, server, port etc on a per email basis.

For the others, you might need to look at Adobe's CFSetup tool, or Ortus Solutions' CFConfig tool.

CF Setup

- Came out with CF2021 - but works with older versions as well (Charlie says CF10+)
- It in the config/cfsetup folder by default.
- Run as an interactive CLI or imperative script
- Can view and set settings
- Can import/export from/to json files as well as servers
- Can import or export from multiple instances, whether they are running or not
- Great integration for Adobe ColdFusion Docker Images
- Can download the tool separately

CF Setup

I am not expert, but here is more information

Video on CFSetup presented by Charlie Arehart

https://www.youtube.com/watch?v=S_UfNptoz4U

Download the tool separately:

https://download.macromedia.com/pub/coldfusion/updates/14/cfsetup/ColdFusion_2021_CFSetup.zip

Docs:

<https://helpx.adobe.com/coldfusion/using/cfsetup-configuration-tool.html>

CF Setup Demo: Starting CFSetup

Run in terminal mode

#>

```
D:\www\__bin\CommandBoxFiles\server\5D99BD329ED53F53  
F1A1C91D0978425F-app1\adobe-2023.0.4.330500\WEB-INF\  
config\cfsetup\cfsetup.bat
```


CF Setup Demo: Setting an Alias

```
cfsetup> alias app1
```

```
D:\www\...\server\5D99BD329ED53F53F1A1C91D0978425F-ap  
p1\adobe-2023.0.4.330500\WEB-INF\cfusion
```

Successfully registered the alias with the instance provided.

```
cfsetup> select app1
```

CF Setup Demo: Listing Datasources

```
cfsetup:app1> show datasource
```

```
Datasource(s)
```

```
contentbox
```


CF Setup Demo: Export Datasource

```
cfsetup:app1> export datasource  
d:/www/poc/cfsummit2023/app1/datasources.json
```

Exporting DATASOURCE.

Provide passphrase(of minimum length 6) to protect sensitive information:*****

Do you want to replace the existing file (Y/N):y

Export successful.

WARN: d:/www/poc/cfsummit2023/app1/datasources.json contains sensitive information and must be stored safely. Also, after use of this file, destroy this file.

CF Setup Demo: Import Datasource

```
cfsetup:app1> import datasource  
d:/www/poc/cfsummit2023/app1/datasources.json
```

Provide passphrase (of minimum length 6) to protect sensitive information:*****

Backup of files are taken and stored at
D:\www__bin\CommandBoxFiles\server\5D99BD329ED53F53F1A1C91D09
78425F-app1\adobe-2023.0.4.330500\WEB-INF\cfusion\lib\cfsetup_
backup\1696177831802.

CF Setup - Password Notes

- JSON file - neoxml passwords use the seed properties. This affects the password on each of the servers
- passphrase - encrypts the whole file - which is not in the file...
- when you import with cfsetup - you need a passphrase

CFSetup decrypts and encrypts the passwords for each server as it sets them, so you can import and export to different versions.

This is why you can't/shouldn't copy xml files between versions. This is why an encrypted password on one server doesn't work on another, the seed is different per server/instance.

CCS - Centralized Config Server

- New with Adobe CF 2023
- Dedicated Server to manage and propagate/sync CF Settings
- Group CF Instances into sets with shared settings
- CF Package settings, such as core, document, mail etc are stored locally
- Maintain a version history of the changes in the settings

CCS Talk was earlier today

Piyush Nayak spoke at 11:45am

Managing ColdFusion configuration settings with CCS

Introducing our brand new server CCS(Central Config Server), that will help you to create, manage and quickly deploy ColdFusion settings. Imagine a world where with just one click you can propagate one essential setting to hundreds of your CF servers! In this session , we will see how this is possible with CCS . The session will essentially cover how to install and run CCS , add all your ColdFusion servers to CCS , categorize CF servers and then manage your settings.



CFCONFIG

CFConfig is a project aimed to help server admins and developers alike manage the configuration of their favorite CF engine.

<https://cfconfig.ortusbooks.com/>

What can CFConfig do?

CFConfig gives you the ability to manage most every setting that shows up in the web administrator, but instead of logging into a web interface, you can manage it from the command line by hand or as part of a scripted server setup. You can seamless transfer config for all the following:

- CF Mappings
- Datasources
- Mail servers
- Request, session, or application timeouts
- Passwords
- Template caching settings
- Basically any settings in the web based administrator

Use CFConfig on any server

CFConfig will work on any CF server regardless of how it was installed. Since it interacts directly with the config files, the server doesn't need to be running. Heck, the server doesn't even need to be installed yet! CFConfig can be used to write out config files before you even start a CommandBox server for the first time.

But it's not just for CommandBox servers. All you need is the folder path to the CF home in your server installation and you can point the CFConfig library at it. This means CFConfig can be used for syncing config across existing servers, standing up docker containers, or provisioning Vagrant VMs.

Use CFConfig on any server

CFConfig interfaces directly with the XML and property files used by your CF engine to store its configuration. It takes care of translating the config properly so you use the same commands regardless of what engine you're managing the config for. The tool will try hard to figure out the version of ColdFusion that you have installed and there are hints you can give it as well.

You can export to and from servers or files, set single settings, diff settings, and more.

CFConfig exists in two parts:

- A service layer for reading, writing, and storing configuration for all CF engines.
- A set of scriptable commands built on top of CommandBox CLI

CFConfig Service Layer



The heart of CFConfig is a standalone module that provides a set of models and services for interacting with configuration files for all CF engines. This library allows for reading, writing, storing, and diffing configuration. This is an underlying service layer meant to have other tools built on top of it.

The CFConfig services do not require CommandBox, but can be used on its own and provides a nice, fluent API for managing configs. They do not use RDS and don't need the server to be running. The services just needs access to the installation folder for a server to locate its config files.

CFConfig Service Layer

- Generic JSON storage of any CF engine's settings
- Engine-specific mappings for all major engines to convert their config to and from the generic JSON format
- Export config from a server as a backup
- Import config to a server to speed/automate setup
- Copy config from one server to another. Servers could be different engines— i.e. copy config from Adobe CF11 to Lucee 5.
- Merge config from multiple servers together. Ex: combine several Lucee web contexts into a single config (mappings, datasources, etc)

CFConfig CLI

The CLI portion of CFConfig warps up the services layer into a CommandBox module that provides command line access to all the features above, but from your native OS shell, bash scripts, automations, or Docker/Vagrant/Heroku provisioners.

The CFConfig CLI also has a deep integration with CommandBox servers making it very easy to manage their configuration.

CFCConfig CLI

- Provides a native CLI tool for managing server configuration
- Scriptable for automated server setup
- Provides complete command help built in
- Tight integration with CommandBox servers

CFConfig Help

* CommandBox Help for cfconfig

Here is a list of commands in this namespace:

cfconfig diff	cfconfig import	cfconfig show
cfconfig export	cfconfig set	cfconfig transfer

Here is a list of nested namespaces:

cfconfig cache	cfconfig debugtemplates
cfconfig mailserver	
cfconfig cfmapping	cfconfig eventgatewayconfig
cfconfig pdfservice	
cfconfig componentpath	cfconfig eventgatewayinstance
cfconfig task	
cfconfig customtagpath	cfconfig eventgatewaylucee
cfconfig datasource	cfconfig logger

CFConfig Show and Tell

Some examples:

- `cfconfig show datasources`
- `cfconfig diff .cfconfig.json`
- `cfconfig set requestTimeout=0,1,2,3`
- `cfconfig show requestTimeout`

Our Apps have evolved, so must we

It is rare for your project to not use some additional 3rd party service today

- Bug logging tools like BugLog, RayGun, Sentry etc
- AWS Tools like S3, SMS, SNS
- Mail Services like Mailgun, Mandrill, Postmark, Sendgrid
- Other services like Google Maps
- External APIs for pulling or pushing data to

Adobe ColdFusion has more and more CF Admin settings for these, but we need store more settings, config, and secret credentials.

Issues with hard coded settings / config



- Maintainability - scattered values are harder to update
- Hardcoding values limits reusability of code in different scenarios
- Security concerns - sensitive data can be a security risk, especially if code is shared, or breached
- Testing and debugging
 - Hard to test different scenarios
 - Hard to test edge cases

A lot of you are using config files already

Many apps have a file that has all your settings in there.
How many have a file like this?

- IF the URL is 127.0.0.1 use these settings
- IF the Folder location is this, use these settings
- IF my name is Bob, use these settings
- If this file exists, use these settings

What about Security??

An astronaut in a full spacesuit is floating in space, holding a large, dark, rectangular object. The background is a deep blue space with stars and a large, bright yellow sun or planet on the right side. The astronaut is positioned in the center-right of the frame, facing slightly towards the left.

Embedded / Hardcoded Credentials are plain text credentials in source code.

Those developers/teams that hardcode credentials into scripts, applications, and systems, usually do it because it helps to simplify deployments and easier developer workflow.

Although, this is done, with increased risks.

Github Secret Scanner



Find hard coded secrets in your code - Github Checkout

<https://www.youtube.com/watch?v=aoL7pDrXt74>

Github detects over 1 million detected secrets per month.
Github notifies partners based on regex expressions
Now supports private repos.

Recently they scanned 37,000 private repos - only 5,000 of those 37,000 repos had secrets exposed

Hardcoding Security Risks

The background of the slide features a dark space scene with a starry background. A green arc is visible in the top right corner, and another green arc is partially visible on the left side. In the center, there is a faint image of an astronaut in a space suit, floating in space. The astronaut is holding a tool or device, and there is a small, bright light source near the astronaut's hands.

Worse case scenario is code being published to a public location, like github. Scanning tools could detect when a new credential is uploaded, and exploit it.

This is apparently what happened with the Uber breach that exposed information of 57 million customers, plus roughly 600,000 drivers.

An Uber employee published plaintext credentials within source code. This code was, at some point, inadvertently posted on Github, a popular repository used by developers. It likely did not take much technical chops for a watchful hacker to notice the embedded credentials on GitHub, then use them to gain privileged access on Uber's Amazon AWS Instances.

Hardcoding Security Risks

The background of the slide features a space-themed illustration. An astronaut in a full spacesuit is positioned on the right side, standing on a dark, rocky surface. In the upper left, a spacecraft is visible against a starry space background. A large, bright yellow circular shape is partially visible on the right edge, and another yellow circular shape is on the left edge.

Hardcoding default passwords into apps and devices can also allow botnets to probe and try to take over devices, allowing back door access where they can do more harm, like malware infections and more.

Additionally, DevOps tools often have secrets embedded in scripts or files, which potentially jeopardizes security for the entire automation process. Thus, gaining control of embedded passwords and keys is an essential requirement for secrets management and DevOps security.

Managing Hardcoded Credentials



Operational Risk:

Hardcoded credentials are often created with the intention that they never be changed—despite the risk that stale passwords present. Thus, admins may feel wary about trying to change certain types of embedded passwords for fear of breaking something in the system, and potentially disrupting company operations.

<https://www.beyondtrust.com/blog/entry/hardcoded-and-embedded-credentials-are-an-it-security-hazard-heres-what-you-need-to-know>

Managing Hardcoded Credentials



Limited visibility and awareness:

A mid-sized organization may have hundreds of thousands—even millions—of passwords, keys, and other secrets sprawled across devices, applications, and systems.

<https://www.beyondtrust.com/blog/entry/hardcoded-and-embedded-credentials-are-an-it-security-hazard-heres-what-you-need-to-know>

Managing Hardcoded Credentials



Inadequate tools:

Unfortunately, there is no viable manual way to detect or centrally manage passwords stored within applications or scripts. Securing embedded credentials hinges on first separating the password from the code, so that when it's not in use, it's secured in a centralized password safe, as opposed to being constantly exposed in plain text.

<https://www.beyondtrust.com/blog/entry/hardcoded-and-embedded-credentials-are-an-it-security-hazard-heres-what-you-need-to-know>

What are Environment Variables



At their core, environment variables are key-value pairs that are available to a running process or program. Environment variables play a crucial role in configuring applications and defining their behavior. These variables are dynamic values that are set within the operating system or runtime environment, and they can be accessed by programs to customize their functionality.

<https://blog.gitguardian.com/using-env-vars-or-not-environment-variables/>

What are Environment Variables

A background image featuring an astronaut in a full spacesuit standing on a dark, rocky surface. In the background, a rocket ship is visible against a starry space backdrop. Large, glowing green circular shapes are positioned on the left and right sides of the slide.

They provide a flexible way to pass information and configuration data to applications without hardcoding them into the code. This flexibility allows for easier deployment and configuration management, as the same codebase can adapt to different environments without modification. Environment variables are one of the 12 'factors' that influenced much of the cloud-native paradigm.

12 Factors Link: <https://12factor.net/config>

Possible Uses for Env Variables



- Path Definitions - Environment variables are commonly used to define paths to important directories or files.
- Configuration Flags - Environment variables can be used to toggle certain features or modify the behavior of an application.
- API Keys and Secrets - API keys and secrets are sensitive information that should not be exposed in the source code. Environment variables provide a secure way to store and access such credentials.
- Database Configuration - Many applications rely on databases to store and retrieve data. Instead of hardcoding the database connection details into the code, developers can use environment variables to specify the database hostname, port, username, password, and other relevant parameters.

How to use Env Vars

- Usually Env Variables are system settings
- You can refer to runtime settings like JVM Properties as Env Variables

With multiple sites on a single machine, security concerns and possible clashes arise, so most languages have adopted modules called DotEnv which supports a file called `.env` which is loaded at Runtime.

Standardization of Approach and Tools



DotEnv conventions, tooling, and best practices are shared across many languages, Node, Python, Docker, ColdFusion even.

CommandBox has a DotEnv module.

<https://www.forgebox.io/view/commandbox-dotenv>

CommandBox DotEnv Module

Allows for auto loading from a .env file into CommandBox runtime environment for a server as it starts up. Those Env Vars can be used in your box.json, server.json and more.

<https://commandbox.ortusbooks.com/embedded-server/configuring-your-server/env-var-overrides>

```
"web": {  
  "http": {  
    "port": "${MY_HTTP_PORT}"  
  }  
}
```

Or an Environment Variable

```
box_server_web_http_port=8080
```


CommandBox CFConfig Env Support

CFConfig allows you to use Environment Variables instead of .CFConfig.json file for more flexible configuration as well.

```
cfconfig_adminPassword=myCoolPass123
# JSON which will be parsed
cfconfig_mailServers=[{"port":"25","smtp":"localhost2"}]
# dot-delimited keys
cfconfig_datasources.myDSN.password=myPass
# array indexes too
cfconfig_mailServers[1].smtp=mail.server.com
```

Workflow Validation

The background of the slide features a composite image. In the foreground, an astronaut in a full spacesuit is shown from the waist up, holding a bright light that illuminates the scene. In the background, a classic car is visible, appearing to be in space. The overall color scheme is dark with green and yellow accents.

`.env.example` is a convention to specify an example of your secure `.env` file that can be committed to your source control, to be used as a template for your `.env` file.

DotEnv CommandBox has a command to validate your `.env.example` file with your `.env` file

DotEnv check

You can also show, set, load, and populate your `.env` file with this CommandBox Command.

DotEnv Examples

* CommandBox Help for dotenv

Here is a list of commands in this namespace:

dotenv check

dotenv get

dotenv load

dotenv populate

dotenv set

dotenv show

DotEnv Demo

DotEnv check

DotEnv check --reverse

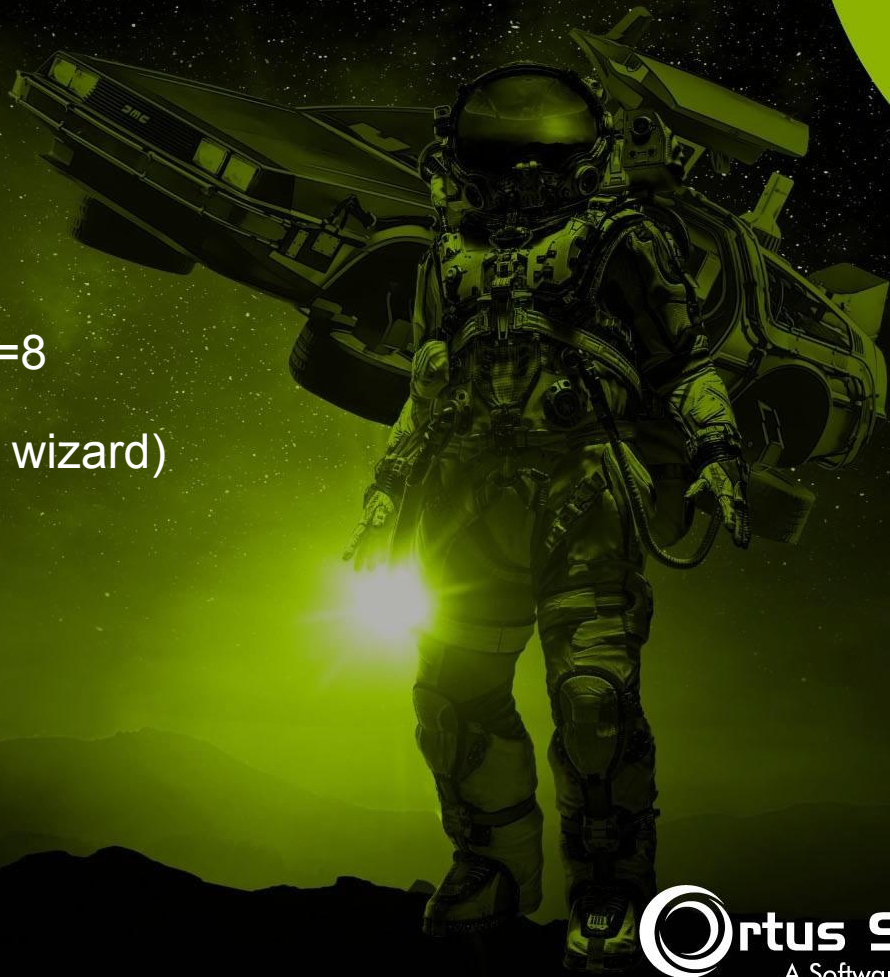
DotEnv set name=test9 value=8

DotEnv set (use the wizard)

DotEnv get DB_DATABASE

DotEnv populate

Echo \${DB_DATABASE}



Workflow Validation - Docker

I built a CommandBox task runner called StackChecker to:

- Validate your Docker Compose file against your .env.example to ensure your environment variables match
- Validate your Docker Compose Stack on your Swarm against your .env.example and local Docker Compose File
- Even checks Docker Secrets, CommandBox Secrets placeholders, and Portainer Secrets

This ensures new env vars are implemented before trying to deploy your code.

<https://www.forgebox.io/view/commandbox-stackChecker>

Check out this Related Talk

**Configuring ColdFusion Docker containers through
environment variables**

Presented by

Guust Nieuwenhuis

Tomorrow, Tuesday October 3, 2023 at 10:15-11:15 (PDT)

In Grand Ballroom D, The Mirage

How to Access Env Vars in ColdFusion

A background image featuring an astronaut in a full spacesuit standing on a dark, rocky surface. In the background, a spaceship is visible against a starry space backdrop. A large, bright yellow arc is on the right side, and a smaller yellow arc is on the left side.

You can use Java to access them

You can check system settings, and java properties

<https://github.com/ColdBox/coldbox-platform/blob/development/system/core/delegates/Env.cfc>

Or

<https://www.bennadel.com/blog/2838-reading-environment-variables-in-coldfusion.htm>

Or look in the server scope.

Environment Variables: SECURE THEM!

- Make sure your .env does not get added to git repo
- Add it to all of your git ignores, NOW! Or else!!!
- Make sure it is not publicly accessible via browser or downloading
- Maybe lock down your .env file for only read access from the tool reading it.

Env Var security concerns



“Every penetration test I know runs [the command] env once they get access to a machine, it is literally the first command attackers run” Alexander Darby

“It is really easy as an attacker to get usernames and passwords because so often they store them as environment variables” Alexander Darby

But wait, why is env vars recommended?

Remember one key thing, to do this, the attacker must already have compromised the system running the application.

They already have access to the code, the RAM, and basically everything. Essentially, to get to this point, the attacker must have already exploited a series of vulnerabilities and will certainly have lots of avenues of attack. Does it really matter then if we are using environment variables?

Alexander was asked this exact question
by the audience.

“You can’t make something 100% secure, but you add
friction” “If developers are lazy, then hackers are lazier”
Alexander Darby

It is a spectrum, so you have to make a decision.

Docker Secrets

The background of the slide features a dark space theme. On the right, a large, bright yellow crescent moon is visible. In the center, an astronaut in a full spacesuit stands on a dark, rocky surface. Behind the astronaut, a white spacecraft with solar panels is floating in space. The overall color palette is dominated by dark blues, blacks, and the bright yellow of the moon and spacecraft.

In terms of Docker Swarm services, a secret is a blob of data, such as a password, SSH private key, SSL certificate, or another piece of data that should not be transmitted over a network or stored unencrypted in a Dockerfile or in your application's source code. You can use Docker secrets to centrally manage this data and securely transmit it to only those containers that need access to it. Secrets are **encrypted during transit and at rest** in a Docker swarm.

Docker Secrets

A given secret is only accessible to those services which have been granted explicit access to it, and only while those service tasks are running.

<https://docs.docker.com/engine/swarm/secrets/>

Docker Secrets

You can use secrets to manage any sensitive data which a container needs at runtime but you don't want to store in the image or in source control, such as:

- Usernames and passwords
- TLS certificates and keys
- SSH keys
- Other important data such as the name of a database or internal server
- Generic strings or binary content (up to 500 kb in size)

How Docker Secrets are shared



When you add a secret to the swarm, Docker sends the secret to the swarm manager over a mutual TLS connection.

When you grant a newly-created or running service access to a secret, the decrypted secret is mounted into the container in an in-memory filesystem. The location of the mount point within the container defaults to `/run/secrets/<secret_name>` in Linux containers, or `C:\ProgramData\Docker\secrets` in Windows containers. You can also specify a custom location.

Reading Secrets with ColdFusion



Once the Docker Container is running, you can read the contents of the file, matching the path in the previous slide.

Example:

```
/var/run/secrets/MASTER_DB_PASSWORD
```

That's not very friendly, so we tried to make it easier with CommandBox

CommandBox Docker Images and Secrets



CommandBox has a convention, when you create a Environment Variable with this format

```
dbPassword=<<secret:MASTER_DB_PASSWORD>>
```

Your Env Var has no secret or credential in it. When CommandBox loads the server in the CommandBox Docker image, it replaces the placeholder for the secret called MASTER_DB_PASSWORD with the value from Docker's secret tooling.

So if you check the Env in your system, it doesn't have the secret expanded, but it will inside of your CFML application.

What is more secure than Env Vars?

External tools that only provide you access to credentials / secrets when you ask for them.

If you load them straight into memory, not in Env Vars, not in System Properties or Java Properties, there are less ways for someone to access them.

But if you load those variables into memory, and they have access to your files, they could see where in CF you are storing them, `application.top.secret.creds` and still output them.

Alternatives to Docker Secrets



Vault, by HashiCorp

Manage secrets and protect sensitive data with Vault

Secure, store, and tightly control access to tokens, passwords, certificates, and encryption keys for protecting secrets and other sensitive data using a UI, CLI, or HTTP API.

Alternatives to Docker Secrets

The background of the slide features a dark space theme. On the right, an astronaut in a full spacesuit is shown from the waist up, holding a glowing blue light. In the upper left, a satellite or space station is visible. A large, bright yellow arc is on the right side, and a smaller yellow arc is on the left side. The overall color palette is dark with yellow and blue highlights.

AWS Secrets Manager

AWS Secrets Manager helps you protect secrets needed to access your applications, services, and IT resources. The service enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle.

Related: Google Cloud Secret Manager, Azure Key Vault, and other cloud options

Alternatives to Docker Secrets

Doppler

Stop struggling with scattered environment variables, hacking together home-brewed tools, and avoiding access controls. Keep your team and servers in sync with Doppler.

Alternatives to Docker Secrets

The background of the slide features a dark space theme. On the right, a large, bright yellow crescent moon is partially visible. In the center, an astronaut in a detailed space suit is shown from the waist up, facing forward. To the left of the astronaut, a satellite or space station module is visible. The overall color palette is dominated by dark blues and blacks, with the yellow of the moon and the astronaut's suit providing contrast.

Keywhiz

Keywhiz is a secret management and distribution service that is now available for everyone. Keywhiz helps us with infrastructure secrets, including TLS certificates and keys, GPG keyrings, symmetric keys, database credentials, API tokens, and SSH keys for external services — and even some non-secrets like TLS trust stores. Automation with Keywhiz allows us to seamlessly distribute and generate the necessary secrets for our services, which provides a consistent and secure environment, and ultimately helps us ship faster.

Alternatives to Docker Secrets

The background of the slide features a space-themed illustration. An astronaut in a detailed spacesuit stands on a dark, rocky surface in the foreground. In the mid-ground, a large, multi-segmented spacecraft or space station is visible. The background is a deep black space filled with stars, and a large, bright yellow-orange planet or moon is partially visible on the right side. The overall color palette is dominated by dark blues, blacks, and the warm tones of the celestial body.

Torus CLI

Torus simplifies the modern development workflow enabling you to store, share, and organize secrets across services and environments. With Torus, you can standardize on one tool across all environments. Map Torus to your workflows using projects, environments, services, teams, and machines.

Alternatives to Docker Secrets

The background of the slide features a composite image. In the foreground, an astronaut in a full spacesuit stands on a dark, rocky surface, looking towards the viewer. In the background, a classic car is visible, seemingly floating in space. The entire scene is set against a dark, starry background with a bright light source creating a lens flare effect. Large, semi-transparent circular shapes are visible on the left and right sides of the slide.

Confidant

Confidant is an open source secret management service that provides user-friendly storage and access to secrets in a secure way, from the developers at Lyft.

More secure?

You could rotate secrets frequently

You could add manual intervention to allow secrets to be loaded, maybe 2FA even, so your app has to have someone manually type in a secret to unlock the secrets.

You cannot make it 100% secure, but you can make it harder.

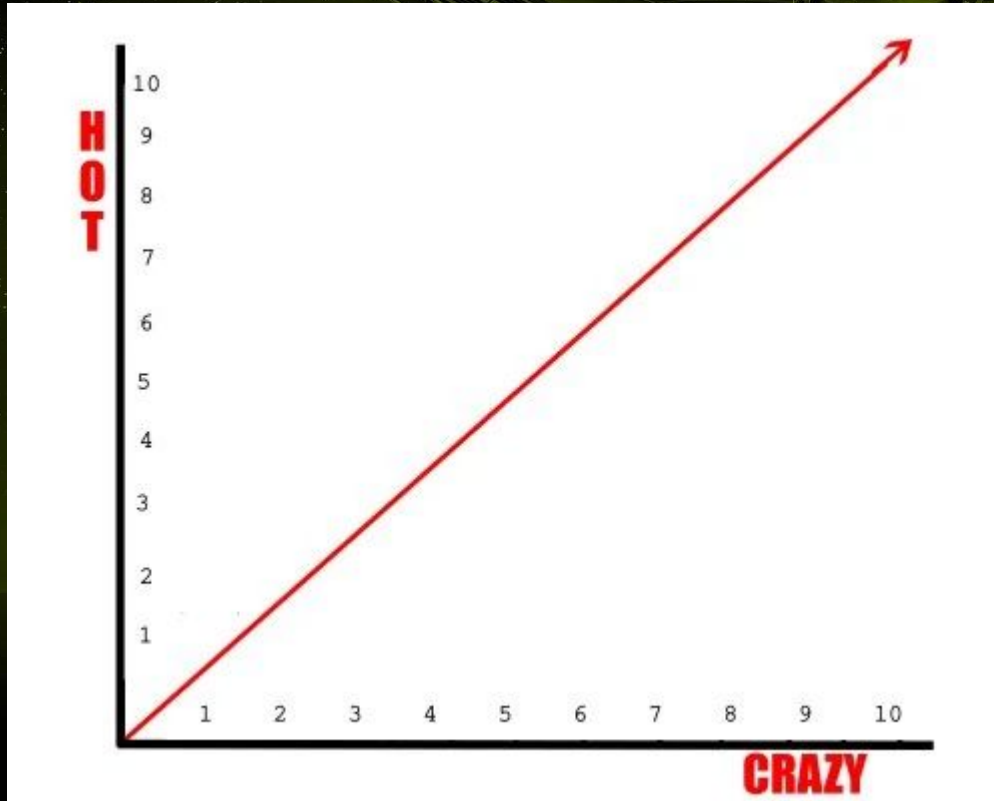
But - at what cost?

Recap of Config as Code Advantages



- Separate code from config - should improve maintainability and readability - especially the idea of MAGIC numbers or MAGIC values.
- Flexibility allows quick changes for your app, for different scenarios, tuning and environments
- Source control of config files (non sensitive) - track changes over time
- Config file setup allows Deployment changes to happen independently from code changes, which can be a time saver, and get for debugging.
- You can time your changes so your code and your config can be ready at the same time, removing the need for complex deploy checklists

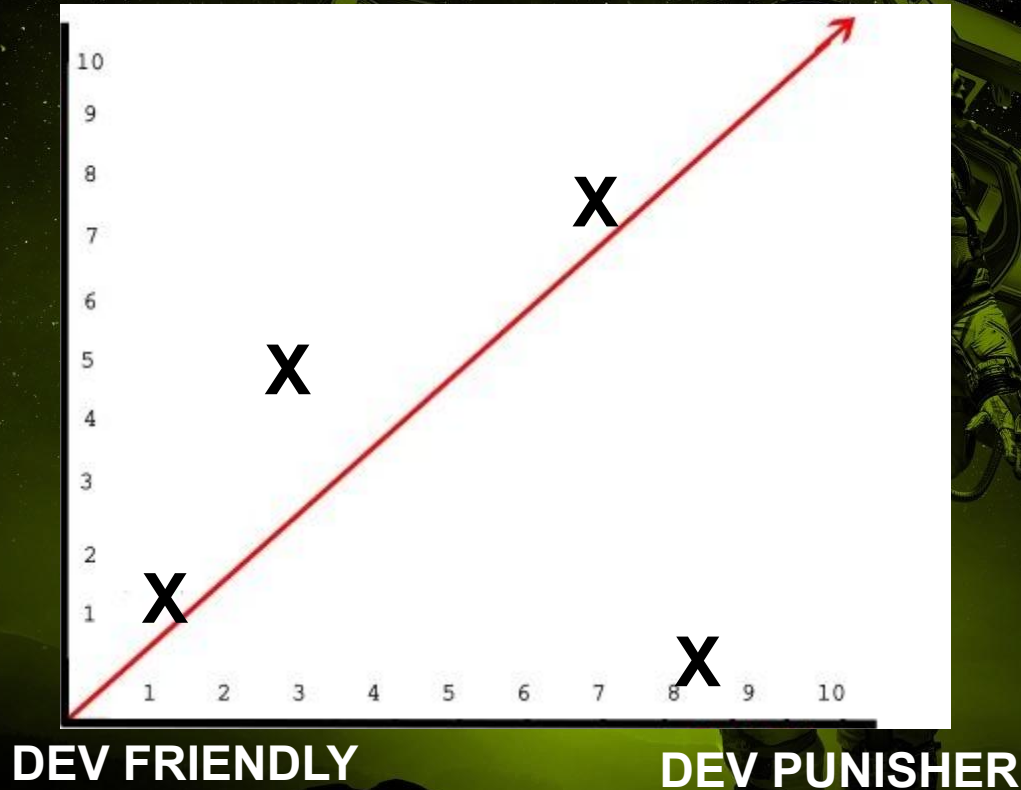
How do you choose?



How do you choose?

MORE SECURE

LESS SECURE



An astronaut in a full spacesuit stands on a dark, rocky surface, possibly a moon or planet. In the background, a car is visible in space, suggesting a surreal or futuristic theme. The scene is set against a starry space background with a bright light source behind the astronaut, creating a silhouette effect. Large, stylized text is overlaid on the image.

Questions ???



INTO THE BOX 2024

THE NEW ERA OF MODERN DEVELOPMENT

"Dive into the fascinating world of modern CFML technologies alongside web development experts and leading innovation companies."

MAY 15th - 17th of 2024
Washington, DC

Scan to read more



Thanks



<https://github.com/gpickin/2023-cf-summit-secrets-and-settings>

- <https://www.ortussolutions.com/>
- <https://intothebox.org/>
- <https://cfcasts.com/>
- <https://www.youtube.com/ortussolutions>
- <https://cfmlnews.modernizeordie.io/>