SySmox WEB security
Info@sysmox.com

Top seven ColdFusion Security Issues

This installment discusses the most prevalent security issues with server configurations and application implementations for ColdFusion. Future articles will discuss other security-related topics, both for other sysmox products specifically as well as for general security concepts that should be helpful for developers, server administrators, and others involved in web implementations.

+ Why Should You Care About Security Issues?

1. **Coldfusion Directory traversal**
2. **FCKeditor bug**
3. **ColdFusion Administrator on Production Servers**
4. **Unvalidated Browser Input**
5. **Sample Applications and Documentation on Production Servers**
6. **CFFILE, CFFTP, and CFPOP**
7. **ColdFusion Studio and RDS with Production Servers**

Please keep in mind that keeping computer security issues at bay can be a full-time job. While these columns seek to provide general education and point out common security issues in implementations, they are not meant as a substitute for a full-time security specialist group or individual in your organization.

Please also remember that when links are provided for reference, they may be advisories not applicable to your server or configuration. Be sure to carefully check whether the fixes and workarounds suggested apply to your configuration before implementing them. Also, be sure to test any patch in a testing environment prior to applying to a production environment.

Why Should You Care About Security Issues?

Security should be everyone's concern. Over time, the professional security community has learned that the best-implemented security is carried out in a thorough and prevalent manner. If a developer, architect, or designer is not thinking explicitly about security, it simply won't happen spontaneously.

Additionally, the wild, wild web is literally saturated with motivated attackers just waiting for a juicy target to attack and subvert. To make matters worse, many would-be attackers can find simple, automated "scripts" (automated tools that search for and exploit known security issues) with which to attack and subvert your server(s).

If any standard, well-known security issue is a concern with your server's configuration, it is only a matter of time before an unknown attacker finds that she can, and does, successfully attack and potentially subvert your systems.

1 - Coldfusion directory traversal:

Variation of a classic directory traversal vulnerability it can be used for arbitrary file retrieval ; special encoding the bug will let you grab any file ending in ".xml", but by adding a "%00″ its sophisticated

The exploit:

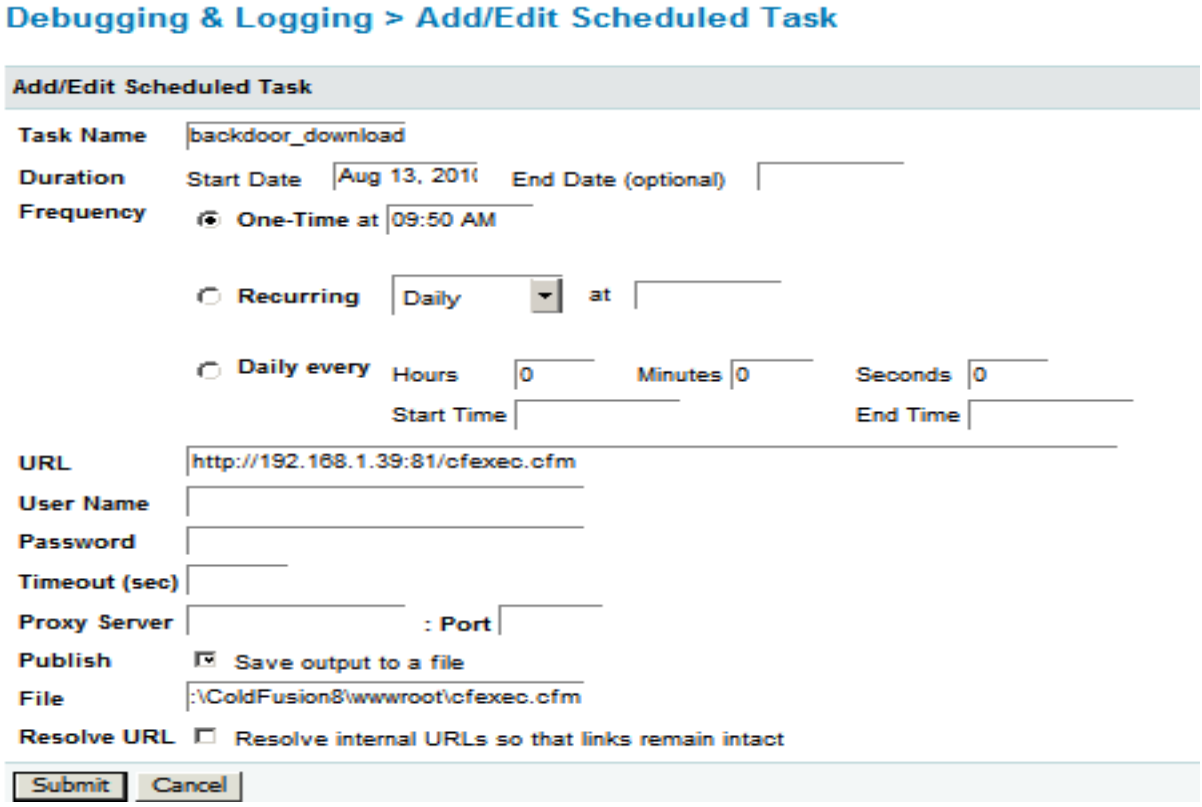
http://server/CFIDE/administrator/enter.cfm?locale=../../../../../../../../../../ColdFusion8/lib/password.properties%00en

If the login admin password was stored hash (Using SHA1 algorithm, similar to CF MX7), the attacker then attempts to crack it via an offline password cracking attack or rainbow table lookup. Note that the default setting in ColdFusion 8 is encrypted=true as per password.properties file. Otherwise.

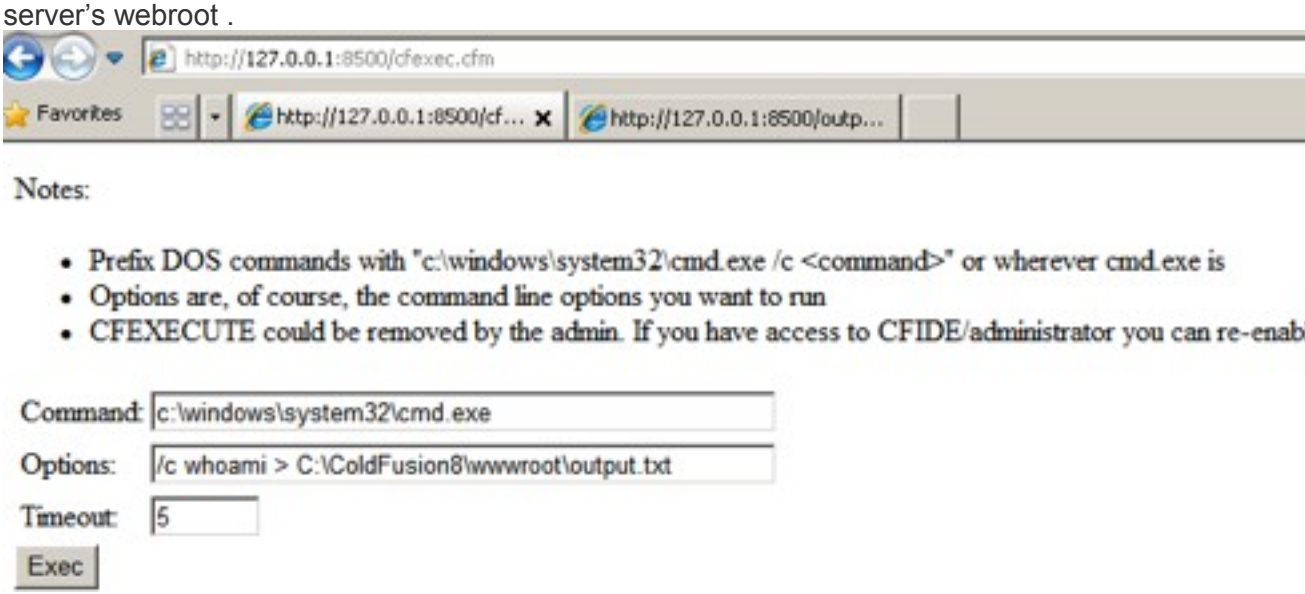
At this point, the attacker would be able to use the decrypted password to login as a CF admin and upload a COLDFUSION BACKDOOR (Trojan horse CFM script that provides an unauthorized remote user with access to a compromised COLDFUSION webserver SYSTEM privileges by default).

Uploading files to a CF server via the administrator console

is a bit counter-intuitive.

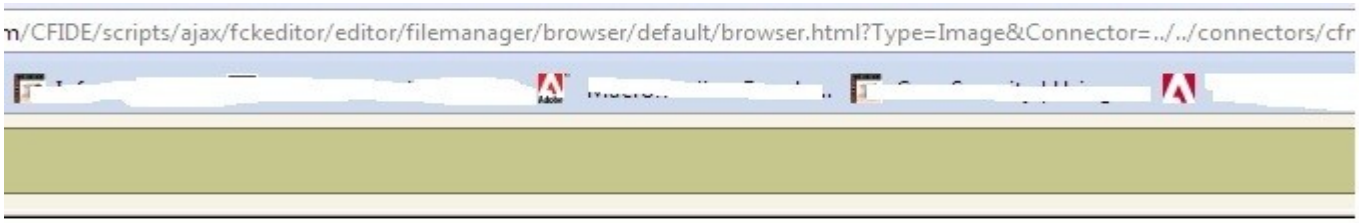


The attacker would basically add a scheduled task that would download backdoor.cfm to the server's webroot .



2- Fckeditor bug:

Last year a vulnerability in some ColdFusion installations. It involves the richtext feature found in cftexarea ; Actually uses an open source app called FCKEditor. The FCKEditor has functionality to handle file uploads and file management but this feature should be disabled in the version embedded in CF server. The problem lies in that in some cases the connector that runs this feature is actually turn on.



.gif
.gif
_logo.gif

Checking config.cfm if conector is on this mean hackers can upload backdoors and files to your web server .

3 -ColdFusion Administrator on Production Servers:

Coldfusion server configuration issues is on of the most important production environments is the default installation . Macromedia advises to use a strongly password that the Administrator login interface requires for login .

Alternatively, if (and only if) you must use the Administrator for remote administration (i.e. for some reason you do not have physical access or file system access to the server itself), we recommend that you install at least one additional layer of security on top of the default provided by the ColdFusion Administrator. Some additional form of authentication is in order, like installing web server-level directory-based authentication or Virtual Private Network (VPN) access via an internal IP address. (A VPN uses encryption to virtually create a private network on otherwise possibly non-secure networks.)

4 - Unvalidated Browser Input :

Unvalidated browser input is an issue that just keeps coming back, not only for users of Macromedia products, but also all over the Internet (see the end of this section for relevant links). For instance, a widely published issue called "Cross-Site Scripting" got a lot of news . This issue as a whole applies to a large portion of the security risks inherent in designing web applications, especially with CFML and Java, which are not generally vulnerable to "buffer overruns," a common security vulnerability .

For more discussion about some advanced kinds of browser input validation, see the section about CFFILE, CFFTP, and CFPOP later in this article.

Unvalidated browser input is a risk wherever the application you develop hands off information gathered from the browser to any other resource. explains that some ODBC drivers potentially allowed users to execute Visual Basic for Applications (VBA) commands on the hosting server without permission.

For the users, doing so was simply a matter of inserting a special character into the HTML form data, and knowing VBA syntax. The offending special character was a '|' character, combined with single quotes. It rested on the shoulders of ColdFusion developers to make sure that any browser data that was going into such a query filtered out the '|' character. Otherwise, there was a potential for hostile users to attack the server by manipulating the data they submitted to the server.

For example, theoretically, if a hostile user changed a URL to look like:

```
http://myserver/page.cfm?x='|shell("cmd /c 1 > c:\temp\foo.txt")|'
```

and the template, page.cfm, contained a query like:

```
SELECT * FROM USERS WHERE lname = '#URL.x#'
```

The URL would cause Microsoft Access to create a file at c:\temp\foo.txt on the server.

In this case, the most appropriate approach to stopping this kind of attack would be to filter URL.x and remove the character '|' before passing the URL.x data on to the query.

In general, browser input validation is a subject you, as a web application architect, designer, or developer, will need to keep in mind as the industry grows and as you encounter more and more situations where it applies. Be on the lookout for any security advisory with the products you use that involves "malformed input". "Malformed", in the security field, loosely translates to something the programmer wasn't expecting, but formally applies only to data or commands that don't follow standard protocol.

Also, be sure to keep in mind that any user could be malicious. There's no way to tell if an attacker may have subverted someone's password without your knowledge. Don't assume that any user has your best interests in mind.

5 -Sample Applications and Documentation on Production Servers :

One important security maxim is that a secured or "hardened" server should have available only the services and resources it needs to do its job and no more. The fewer services or resources a server has installed and available, the fewer details an administrator has to worry about being non-secure, and the fewer possible entrance points a potential attacker has.

Keeping this and the assumption that any production server should also be a hardened server in mind, it becomes obvious that sample applications and documentation have no legitimate place on a production server. The sample applications and documentation have a place in a development environment. However, once your projects and applications reach the testing phase, the servers they reside on should not have either the sample applications or documentation installed.

Another aspect of this issue that is also a risk for many production environments is the propensity for development or test servers to be rolled into production. Often, unless there is sufficient double-checking, the on-server sample applications and documentation will be rolled into production with the development server. If you know that your test or development servers will be rolled into production after testing is complete, think about what you're doing before you enable the option to install sample applications and documentation during ColdFusion Server installs.

6 -CFFILE, CFFTP, and CFPOP

For programmers, it's a good idea to keep in mind that one common method of attack through web applications is finding ways to save unauthorized files to a web server's file system. This is useful in many two-step attacks: the first step is to get an unauthorized file saved to the server's file system, and the second step is to get it executed in some way.

If an attacker could get, for example, her copy of "application.cfm" saved to your application's root directory, all sorts of heck could break loose. Even saving a file that was any regular CFML template could be very dangerous. If the file were saved to a web root directory or lower, the attacker would only need to figure out the URL to the template in order to have it executed. In addition, if the server under attack is running Windows NT and the Windows NT Registry is not properly secured, a template like this could read your entire Registry, which often contains passwords, license keys, and other restricted information, out to the attacker.

The reason CFFILE, CFFTP, and CFPOP are important is that they all allow developers to determine whether files will be uploaded to your server. Developers should keep in mind the security issues relevant to having potentially unauthorized or unvalidated files saved to the server.

CFFILE allows users to upload a file to the server. CFFTP allows the server to send files to or retrieve files from FTP servers on the Internet. CFPOP allows the server to act as a POP3 client and download mail from POP3 servers on the Internet and save them to your server. All three of these tags potentially allow unvalidated and unauthorized files onto the server and should be used and implemented with great care.

If you must use any of these tags in your application, a good way to deal with the risk is to manage it. The best way to manage this risk is to validate the data retrieved. First, save the file(s) to a safe directory. "Safe" means a directory from which the web server will not serve content. This helps prevent a possible attack from targeting a directory from which the web server might execute an uploaded template. In addition, a safe directory should be protected from execution by any other service running on the server.

Second, be sure to validate the data either before it is saved to the safe directory, or immediately afterwards. This means that if you can find or create a utility, CFX tag, or other mechanism to do so, it's a very good idea to make sure that the data is what it says it is. For example, if your application allows uploading graphics to your server, the uploaded files should be checked to make sure they truly are graphics before they're incorporated into any other process. For example, some existing CFX tags are image loaders that will return errors if the file specified is not a valid graphics file.

Finally, validate the file with a virus scanner. If you're going to use the file again, the file may be executed in some way by a word processor, mail client, web server, or some other process. If this is a possibility, some attacker may upload a virus-infected file to try to compromise the security and/or functionality of your server. Configuring a virus scanner to do regular scans of the particular directory where the files are stored is a great way to add an extra layer of security to the process.

Also, server administrators have the option to lock out the use of some of these tags completely using Basic Security. If using Advanced Security and security sandboxing, administers can lock out the use of these tags by some developers and allow their use by others. They can also lock down the directories to which the retrieved files are saved with a secure sandbox. Particularly in a hosting environment where developers from different organizations might share the same server and where the server administrator may not be 100 percent sure of the motivation or intentions of the developers, it would be a good idea for the hosting company to investigate use of security sandboxing.

7 -ColdFusion Studio and RDS with Production Servers

ColdFusion Studio and Remote Data Services (RDS) should only be used with secure Intranets or with internal development servers. Without additional measures of protection, RDS should not be relied upon for secure access to production servers or other mission-critical resources.

If you must use RDS in a secure environment or with a protected resource, be sure to use a non-default password. You should also consider setting up another secure layer to protect the transaction(s), such as Virtual Private Networks (VPN) or some kind of file-drop secured relay from the staging server to the production server.

With VPN implementations, it would be possible to give users a protected, encrypted, virtual secure network on which to use RDS without fear that the RDS session could be hijacked or otherwise compromised.

A file-drop secured relay is an older technology, which may mean that the solution would be less packaged. Still, file-drop relays are related to full-cycle implementations, which you should consider using to develop your web applications if you aren't already assuming you have the resources to do so.

A full-cycle implementation includes project phases like requirements gathering, design, development (implementation), unit testing, regression testing, and rollout (moving the application into production). With a file-drop relay, a developer or tester could signal that her package was ready for the next stage and an automated, secure process (e.g., FTP over secured network pathways, a supervised manual process, etc.) would copy the templates and supporting files from the development or testing server to its next logical destination within the process.

For instance, a developer, once finished with her own development and unit testing, could signal the file-drop process to copy the files from her development server to the testing server for more unit and regression testing.

The Macromedia leaves final implementations to your network and systems groups. Ideally, RDS should not be involved in unprotected remote connections to production servers. At the same time, compromise may be required. Remember that compromise can often mean taking security risks. There's usually a clear tradeoff between convenience and security.

Ammara reda

SySmox operations manager

info@sysmox.com

Http://www.sysmox.com