CFMX Sandbox Security para ambientes compartilhados

Alex Hübner,
http://www.cfgigolo.com
alex@hubner.org.br
Macromedia Certified Professional

1. Introdução

Freqüentemente me deparo com provedores de hospedagem que oferecem ColdFusion em seus planos. Também vejo que muitos destes desconhecem quase completamente esta plataforma, especialmente em ambiente compartilhado. São vários os motivos que levam a esta situação. Em minha opinião pessoal, o principal é a incompetência. Alguns administradores e empresas irresponsáveis, na ânsia de oferecer tudo aqui e agora, acabam deixando a segurança e qualidade em segundo (ou último) plano, não estudando ou procurando dominar uma determinada tecnologia em voga e oferecida por eles, como é o caso do ColdFusion.

Certamente existem empresas e pessoas oferecendo um serviço de hospedagem ColdFusion competente e de qualidade, contudo muitos são inaceitáveis. Costumo separar estes em três tipos: o primeiro, dos totalmente irresponsáveis, que não conhecem nada do produto e o instalam/oferecem assim mesmo. O segundo, dos que instalam e oferecem, mas "limpam as mãos" quando algo sai errado dizendo: "ColdFusion é inseguro, instável, blá, blá", muito comum em locais cuja cultura tecnológica é preconceituosa e estagnada. O terceiro e último é composto por aqueles que conhecem um pouco da plataforma, entretanto não de forma satisfatória ou completa. Desconhecem, por exemplo, a necessidade de se configurar sandboxes. Por isso, num suspiro de bom senso e sanidade, optam simplesmente por desabilitar tags "sensíveis" tais como CFFILE, CFDIRECTORY, entre outras, acreditando que assim estarão seguros. Ledo engano.

Esta situação é embaraçosa para todos os desenvolvedores ColdFusion que se vêm marginalizados e à mercê de provedores de hospedagem ruins e de segunda mão, alguns que sequer possuem uma licença oficial do software.

2. Objetivos

O objetivo deste documento não é ensinar administração de ColdFusion Server, muito menos segurança em servidores ColdFusion MX. Trata-se de uma pequena coletânea de settings que recomendo para servidores de hospedagem compartilhados usando **ColdFusion Enterprise com sandbox security**. Abrange especificamente as versões ColdFusion MX 6.0 e 6.1 **Enterprise** e em modo "stand alone" ou "server configuration". Também pode ser usado para o ColdFusion instalado "em modo J2EE", seja sob JRun ou outro servidor Java em ambiente Windows. Todos os exemplos são baseados na plataforma Windows, todavia também serão válidos para Linux, bastando adaptar as informações.

Também não é objetivo deste documento aprofundar-se em detalhes técnicos e explicações pormenorizadas. Isso certamente daria um livro, não um documento de poucas páginas. Encare-o como uma espécie de "receita de bolo". Lembre-se que este documento não necessariamente será adequado para o seu caso. Atente à descrição do ambiente em que estou trabalhando (parágrafo acima) e adapte/use tais informações de acordo com seu julgamento e experiência.

Estou aberto a responder quaisquer dúvidas e especificidades deste documento na lista de discussão ColdFusion Brasil – http://www.coldfusion.org.br. Críticas, sugestões de melhoria e correções para este também são bem vindas. Cadastre-se na lista e poste sua dúvida/crítica/sugestão/correção para que todos possam se beneficiar das respostas (não apenas minhas).

3. Hospedagem compartilhada

Algo que precisa estar claro desde já: uma hospedagem compartilhada nunca será 100% segura. Por mais que se diga o contrário, o fato é que compartilhando recursos do servidor, riscos inerentes também serão, por assim dizer, “compartilhados”. Isso porquê em tal ambiente não existe isolamento total entre usuários e aplicações. Qualquer aplicação pode causar interferência numa outra aplicação e no servidor todo. Um exemplo clássico são ataques do tipo DoS (não confundir com DDoS), propositais ou muitas vezes não - alguns programadores “mãos pesada” conseguem criar monstros devoradores de CPU e memória sem perceber - onde um script ou uma aplicação inteira põe o processador em apuros, comprometendo a performance de todo o servidor.

Caso você não abra mão de ter um ambiente absolutamente seguro, deverá partir para uma solução dedicada ou semi-dedicada. O ColdFusion MX oferece a possibilidade de se ter, numa mesma máquina física, múltiplas instâncias de seu serviço rodando de forma isolada entre elas. Trata-se de um isolamento tanto em termos de segurança quanto em termos de performance já que nesta modalidade, cada instância tem sua JVM própria. Para os que não conhecem Java, o termo JVM é uma abreviação para Java Virtual Machine, onde o software cria uma abstração própria que, grosso modo, funciona como se fosse uma máquina exclusiva e separada.

Para esta abordagem existe um custo maior tanto operacionalmente quanto em termos de consumo de recursos da máquina. Trata-se, portanto de uma excelente opção para planos “semidedicados” ou mesmo “dedicados”, ainda não oferecidos no Brasil (veja que boa oportunidade!) e em raros casos no exterior. Por isso não iremos tratá-la neste documento.

4. As duas versões comerciais do ColdFusion

Muitos não sabem, mas existem duas versões comerciais do ColdFusion Server, uma mais barata, apelidada de Standard (também conhecida como Professional na versão 6.0) e outra, mais cara, chamada Enterprise. As diferenças? Para quem desenvolve nenhuma uma vez que o suporte a CFML é idêntico. No entanto, para quem hospeda, as diferenças são muito importantes e significativas. O ColdFusion Enterprise oferece suporte a um número maior de plataformas e bancos de dados. Adicionalmente, a versão Enterprise suporta a instalação de múltiplas instâncias (mencionadas no item anterior), sandboxes (contextos) de segurança - indispensáveis numa hospedagem compartilhada - e otimizações especiais para o envio de grande quantidade de e-mails. Inclui ainda uma versão completa do famoso servidor J2EE da Macromedia, o JRun e alguns outros recursos interessantes. A tabela comparativa das diferenças entre as versões pode ser encontrada aqui no seguinte link: http://www.macromedia.com/software/coldfusion/productinfo/product_editions/. Se você está oferecendo ColdFusion MX em ambiente compartilhado, sua escolha natural deve ser pela versão Enterprise. Infelizmente muitos provedores, desconhecendo as diferenças (ou não), optam por comprar a versão "mais barata", Standard. Lembre-se de que o barato sai caro: uma hospedagem compartilhada na versão Standard é arriscada.

Normalmente (repetindo: normalmente) desenvolvedores CFML são bem intencionados e não perdem tempo com tentativas de invasão e ou outras atividades virtuais ilícitas. Além disso, para explorar vulnerabilidades usando scripts CFML é necessário que se tenha uma conta no servidor, o que faz do atacante quase obrigatoriamente um cliente. Isso já é motivo (ou não) para deixar os mal intencionados do lado de fora, uma vez que prejudicando o servidor ele estará prejudicando sua própria conta e aplicação. Porém nunca confie na boa índole (ou capacidade de fazer besteira) de todos eles.

5. Sandboxes

Quem tem gato em casa (especialmente em apartamento) sabe que eles precisam (e gostam) de usar uma caixa de areia (ou de granulado especial) para fazer suas necessidades e enterrá-las de forma higiênica. O conceito de sandboxes no ColdFusion pode ser explicado, de forma rudimentar,

fazendo uma analogia à caixa de areia dos gatos. Nesta caixa isolada seus usuários poderão fazer o que bem desejarem, de forma "higiênica", sem sujar ou estragar o que estiver ao redor. O que isto significa na prática? Significa que tags como CFFILE, CFDIRECTORY e outras que, potencialmente podem causar estragos, podem ser habilitadas sem qualquer tipo de problema. O usuário de uma sandbox devidamente configurada, não terá permissão para "sujar" a conta do vizinho ou qualquer outro local. Ele não vai conseguir fazer upload de arquivos, deletar, copiar ou criar diretórios fora da sua pasta raiz, da sua sandbox. Porém existem algumas pequenas exceções, mais à frente vamos falar delas.

Torna-se claro portanto que toda e qualquer hospedagem compartilhada deve fazer uso de sandboxes. Por isso pergunto: sandbox security já está habilitada no seu servidor? Não? Então vamos habilitar:

a. Entre no ColdFusion Administrator;
b. No menu lateral, encontre a opção "Sandbox security" (a última) dentro da área chamada "Security";
c. Se você não habilitou isto antes, a opção "Enable ColdFusion Security" deverá estar apagada. Selecione-a e clique em "submit changes";
d. Será necessário reiniciar o serviço do ColdFusion Server;

Uma vez completado os passos acima, você deve entrar novamente no ColdFusion administrator e verificar se ele criou os dois diretórios padrões e necessários. Veja a imagem abaixo:

Estes dois diretórios (ColdFusion CFIDE e ColdFusion WEB-INF) são necessários para que o ColdFusion funcione perfeitamente. Por isso eles não podem ser removidos (note a ausência do ícone de deletar lado). É importante saber que a entrada que aponta para o diretório CFIDE deve corresponder ao diretório onde estão os arquivos da aplicação "ColdFusion Administrator". Se a pasta CFIDE estiver em seu local de instalação padrão o CFMX irá configurá-la corretamente. Entretanto, caso você tenha trocado a localização da pasta CFIDE será necessário criar uma nova entrada de sandbox security para esta. Isso pode ser feito de forma simples: na primeira caixa de diálogo ("Add Security Sandbox"), selecione a opção já existente para a pasta CFIDE em "New Sandbox, or pick one to copy from". Feito isso, especifique (você pode usar o botão "browse server") a localização da pasta CFIDE que você está usando e clique em "Add".

6. Criando novas sandboxes

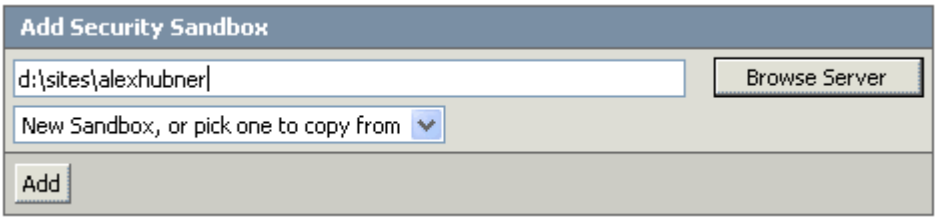
Antes de começarmos, é necessário fazer uma observação sobre uma prática muito comum em provedores de hospedagem compartilhados: muitas vezes o nome de pastas base de usuários são definidas usando uma sintaxe tal como: “d:\Inetpub\dominio.com.br”. Esta sintaxe é desaconselhável, pois utiliza caracteres especiais e há casos onde esta sintaxe será lida e interpretada pelo enginee do CF como sendo “d:\Inetpub\dominio\com\br”.

O processo de criação de sandboxes é bastante simples. Uma sandbox é criada sempre sob uma pasta base ou raiz, normalmente a pasta base do usuário no servidor. Para exemplificar imagine o seguinte cenário:

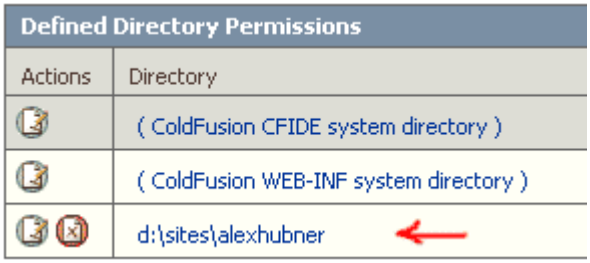
a. Um usuário “alexhubner” possui um site no servidor cuja pasta raiz foi configurada como/em “d:\sites\alexhubner”;
b. Sob a pasta “d:\sites\” existem outras contas de usuários como por exemplo “d:\sites\zezinho”, “d:\sites\mariazinha” etc;
c. Este usuário deve ter controle total (read, write, execute (scripts cfml) e delete) via CFML nos arquivos e pastas existentes sob sua pasta raiz.

Diante de tal cenário, o processo de criação consiste nesses dois passos simples:

a. Em “Add Security Sandbox”, especifique o caminho completo para a pasta base do usuário, no nosso caso “d:\sites\alexhubner”;
b. Clique no botão “Add”.



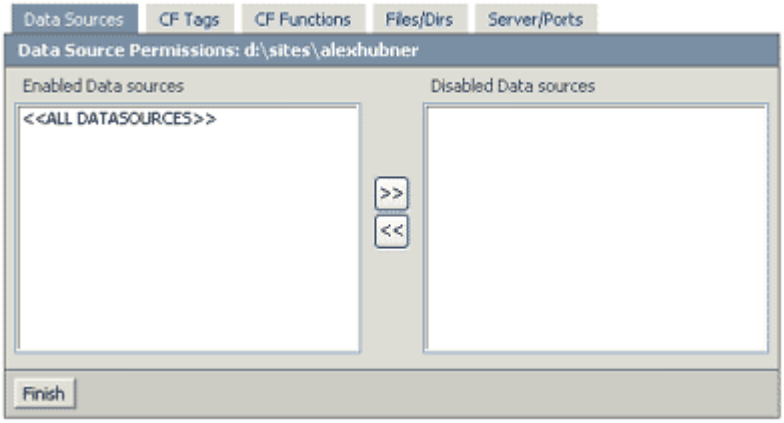
Feito isso, a nova sandbox, representada pelo caminho da pasta base, deverá aparecer na listagem de sandboxes existentes:



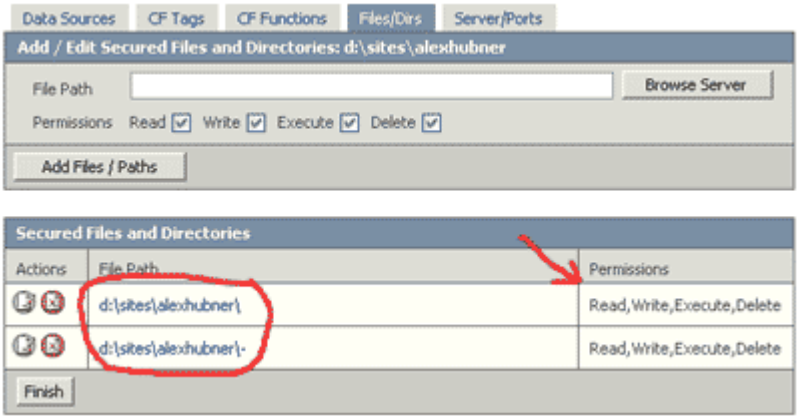
O próximo passo será configurar em detalhes esta sandbox. É aqui que este documento começa a fazer sentido na medida em que as configurações que proponho, daqui em diante, foram pensadas visando a melhor segurança do servidor em ambientes compartilhados.

7. Configurando uma sandbox para ambiente compartilhado

Ao se clicar na sandbox em questão, temos uma interface com cinco “orelhas” a saber: “Data Sources”, “CF Tags”, “CF Functions”, ”Files/Dirs” e “Server/Ports”. Veja a imagem abaixo:



O esquema de funcionamento desta interface é bastante simples. Para todas as orelhas, na janela esquerda têm-se o que está habilitado e na janela direita o que está desabilitado. Por padrão, cada nova sandbox terá permissão de acesso em todas as data sources do servidor e qualquer tag e função disponíveis. Também poderá acessar qualquer recurso externo (webservices ou consultas/acessos com CFHTTP, CFFTP etc) uma vez que não existe nenhuma restrição em "Server/Ports". A exceção é a manipulação de arquivos que, por padrão (obviamente), ficará limitada à pasta base do usuário ("d:\sites\alexhubner"). Um clique na orelha "File/Dirs" evidencia isso:



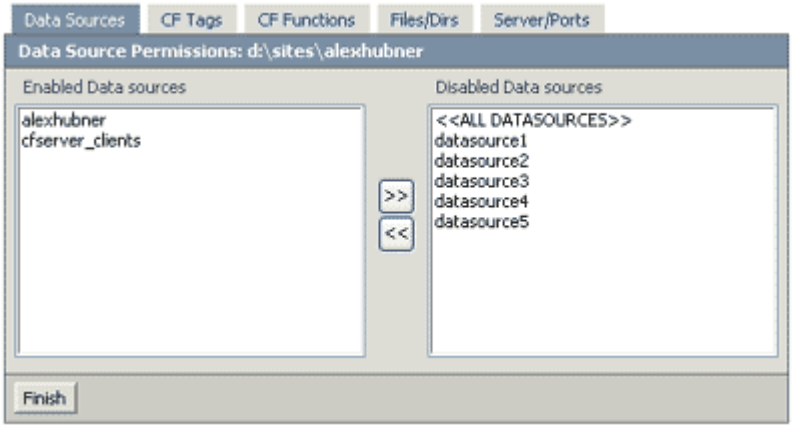
Perceba que existem duas entradas para a pasta do usuário. Em teoria bastaria uma entrada no formato "d:\sites\alexhubner\-". Note o traço, que significa que a sandbox tem autorização para ler, escrever, executar e deletar qualquer arquivo e pasta recursivamente a partir da pasta base. Porém, por algum motivo que desconheço, o ColdFusion precisa de ambas entradas para funcionar corretamente. Não me pergunte por que, eu realmente não sei.

Eu costumo adicionar uma terceira entrada (não obrigatória) apontando para a pasta temporária do ColdFusion. Algumas aplicações em CFML fazem uso desta através da função GetTempDirectory(). Esta pasta e função podem ser usadas para armazenamento temporário de arquivos em aplicações específicas. A pasta temporária do ColdFusion costuma ser "C:\CFusionMX\runtime\servers\default\SERVER-INF\temp\wwwroot-tmp\-" (note que já coloquei o traço), habilite-a sem problemas.

Vamos dar início a um "tour" pelas orelhas e suas configurações recomendadas.

8. Orelha Data Sources

Na orelha “Data Sources” nós indicaremos ao ColdFusion quais data sources - conexões JDBC que permitem a interação com bancos de dados – poderão ser usadas por uma determinada sandbox/conta. Na imensa maioria das vezes habilitamos àquelas pertencentes apenas à conta de hospedagem em questão. Imagine que o nosso cliente tem direito apenas a uma data source no seu plano, a orelha ficaria desta maneira:



Você já deve estar se perguntando: por que a data source “cfserver_clients” também está habilitada? Não deveria ser apenas a do cliente? Bem, no meu servidor, por questões de performance e segurança, costumo armazenar variáveis de cliente (opção “Client Variables” no Administrator) num banco de dados de alta capacidade (mySQL, SQL Server etc) em oposição às alternativas existentes (cookie e registry - mais a frente vamos falar dos problemas com o registro) – saiba mais neste link: http://mysecretbase.com/ColdFusion_Tutorial_01.cfm. Desta maneira a sandbox em questão precisa ter acesso à data source responsável pelo armazenamento de variáveis de cliente no servidor. No nosso caso ela se chama “cfserver_clients”.

Não dediquei um item especifico a esta questão, mas saiba desde já que, para uma configuração segura (e mais rápida também), você deve armazenar as variáveis de cliente num banco de dados separado, mesmo que isso seja feito usando uma base de dados em Access, no improvável caso de não existir um banco de dados mais potente para isso.

Uma informação adicional que julgo importante: muitos oferecem o Access como opção “barata” de banco de dados em hospedagem compartilhada. Lembre-se (de novo): o barato pode sair caro. O Access é um péssimo banco de dados para aplicações web, ainda mais quando localizado na mesma máquina do servidor de aplicações. Prefira oferecer ao seu cliente um banco de dados melhor, nem por isso mais caro. O MySQL em uma máquina separada (Linux de preferência) é uma excelente (e viável) opção em substituição ao problemático Access uma vez que o ColdFusion oferece suporte pleno a este banco de dados. Você certamente poupará dor de cabeça com suporte técnico e eventuais travamentos sem explicação (para o usuário) da base de dados em Access.

9. Orelhas “CF Tags” e “CF Functions”

Chegamos à parte mais importante deste documento. Você deve se recordar que, mais acima no item cinco, eu mencionei o fato de existirem algumas exceções à segurança completa de uma sandbox. Justamente por conta destas exceções algumas tags e funções CFML precisam estar desabilitadas. Recomendo a leitura da documentação presente no seu CD ou download de CFMX sobre as tags e funções em questão antes de prosseguir.

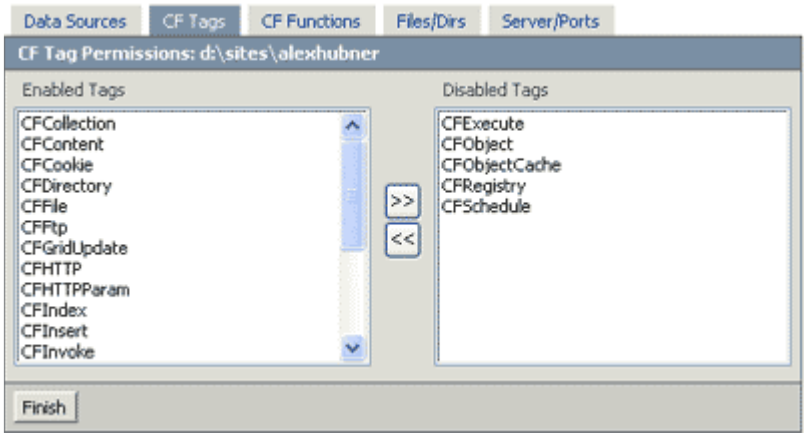
Na orelha de “CF Tags”, recomendo desabilitar as seguintes tags:

- **CFExecute:** permite executar arquivos e programas diversos no servidor. Um exemplo é o

programa “ping.exe” que vêm com o Windows e muitos outros que aceitem execução via linha de comando. Como na grande maioria dos casos, a instalação do ColdFusion server é feita usando-se a conta “System” do Windows. Isso pode se tornar perigoso uma vez que o usuário pode fazer upload de um executável malicioso em sua pasta base e executá-lo usando o ColdFusion, sob a conta “System” com privilégios elevados. Adicionalmente você poderá ter problemas de performance caso o usuário tente acessar um executável problemático, que consuma muitos recursos do servidor. Considere instalar o ColdFusion sob uma conta diferente da “System”, com poucos privilégios;

- **CFObject:** leia a explicação sobre esta tag no item 10;
- **CFObjectCache:** leia a explicação sobre esta tag no item 10;
- **CFRegistry:** raramente uma aplicação em CFML precisa acessar o registro do windows e quando precisa, trata-se de uma ação muito específica e que muito provavelmente estará além do que uma hospedagem compartilhada pode oferecer. A tag CFRegistry também possui acesso em modo “System” ao registro do Windows, o que é arriscadíssimo, por isso recomendo desabilitar esta tag. Lembre-se do que foi dito no item 8 sobre configurar o armazenamento de variáveis de cliente em um banco de dados ao invés de cookies e registro (padrão no CF). Caso esta mudança não seja feita, você **não** poderá desabilitar esta tag pois o servidor ColdFusion precisará que esta esteja habilitada, de forma semelhante ao que acontece no caso da data source “cfserver_clients” do nosso exemplo.
- **CFSchedule:** o agendamento de tarefas é uma das grandes funcionalidades do ColdFusion Server, porém deve ser usada com parcimônia. Uma vez mal utilizada pode vir a trazer problemas de performance para o servidor. Em servidores compartilhados onde esta tag está habilitada é comum vermos tasks agendadas para rodarem a cada minuto ou as vezes muito menos que isso! O agendamento de tarefas em scripts CFML deve ser feito somente mediante solicitação formal por parte do cliente com um intervalo mínimo entre as execuções para não causar problemas de performance e “abusos” de agendamentos com scripts pesados.

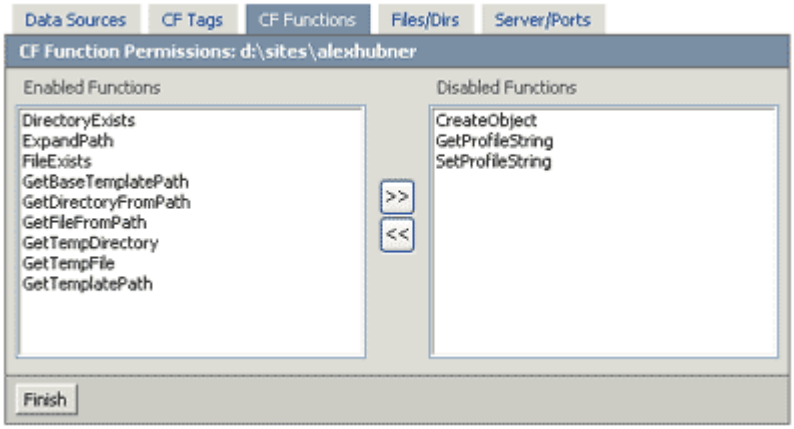
A orelha “CF Tags” ficaria assim então:



Na orelha “CF Functions”, recomendo desabilitar as seguintes funções:

- **CreateObject:** leia a explicação sobre esta função no item 10;
- **GetProfileString e SetProfileString:** estas duas funções são raramente utilizadas em aplicações CFML. São usadas para, entre outras funções, configurar e ler arquivos de inicialização de aplicativos (.ini entre outros), incluindo o próprio servidor ColdFusion. Recomendo desabilitá-la portanto.

A orelha “CF Functions” ficaria assim então:



10. CFOBJECT, CFOBJECTCACHE e CreateObject

No item anterior sugerimos que as tags CFOBJECT, CFOBJECTCACHE e a função CreateObject fossem desabilitadas. Em versões anteriores ao CFMX desabilita-las não seria um grande problema, porém estas ganharam um apelo e utilidade bastante importante nas versões CFMX o que certamente trará dores de cabeça a você e ao seu cliente. Tenha certeza de que ele vai reclamar o fato destas não estarem habilitadas, porém existem questões de segurança e privacidade importantes envolvendo-as, o que as tornam candidatas à desabilitação. Para explicar o porquê desta recomendação, faz-se necessário uma breve explicação sobre as funcionalidades das mesmas.

Com a mudança de plataforma para Java, o ColdFusion oferece aos seus usuários um leque enorme de possibilidades, a melhor delas: total integração com o ambiente Java, incluindo execução de classes, EJBs, custom tags JSP e muitas outras. Esta interação é feita usando-se as tags CFOBJECT, CFOBJECTCACHE e a função CreateObject. Estas tags também são usadas com componentes COM, CORBA e outros. Até aqui nenhum problema, exceto que de forma similar ao que ocorre com a tag CFEXECUTE, é possível acessar componentes COM e/ou objetos Java maliciosos, que não necessariamente estarão isolados dentro da sandox. Também é possível acessar objetos e classes (e métodos) do próprio ColdFusion de uma forma muito íntima e insegura. Podemos citar algumas tais como recuperar senhas de data sources, senha do próprio ColdFusion Administrator (em versões CFMX 6.0 pré Updater 1), variáveis de sessão existentes em outras aplicações do servidor etc, isso tudo mesmo estando dentro de um contexto de sandbox. O problema não se limita à possibilidade de explorar e ler informações sensíveis, mas também de alterar, criar e deletar configurações diversas. Usando estas tags e a função CreateObject, é possível, por exemplo, deletar data sources ou criar novas. Pode-se criar mappings, schedules e uma infinidade de outros recursos inerentes ao ColdFusion Server (lembre-se de que o ColdFusion é um aplicativo Java). Têm-se também acesso à API Java, existente sob o CFMX, o que pode trazer problemas uma vez que se trata de um universo complexo e à parte.

Mas e qual será o motivo da reclamação por parte do cliente se ele não estiver querendo usar Java? Bem, estas tags podem ser usadas como uma forma elegante (e cada vez mais presente entre os desenvolvedores) de se programar em CFML como se fosse uma pseudo-linguagem orientada a objetos. Também são usadas com freqüência em substituição à tag CFINVOKE para utilizar e consumir Componentes ColdFusion (arquivos cfc), usados com freqüência em novas aplicações.

Esta é uma das minhas mais sérias críticas ao ColdFusion Server. Não vou citar exemplos nem demonstrar como se pode obter ou alterar tais informações. Lembre-se de que se o ColdFusion Administrator, uma aplicação inteiramente feita em CFML (criptografada), é capaz de alterar e controlar quase todas as configurações do servidor, você provavelmente poderá fazer o mesmo com a sua aplicação CFML. Desabilitando tais tags você impede que isso ocorra (lembre-se de que

o ColdFusion Administrator - CFIDE - possui sua sandbox, que foi adicionada automaticamente, onde estas tags estão habilitadas).

Espero que as próximas versões do ColdFusion tragam uma solução para este problema. Até lá, desconheço qualquer maneira de se contorna-lo sem sacrificar o uso das tags em questão. Já pesquisei bastante sobre o assunto, contudo sem sucesso. Caso você conheça alguma maneira, por favor, entre em contato.

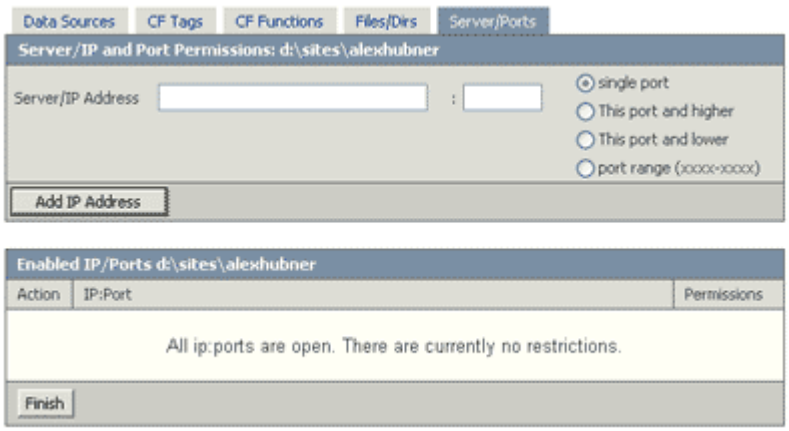
A única forma de se oferecer suporte a estas tags, com total segurança, é através de múltiplas instâncias do ColdFusion (possibilidade de configuração citada neste documento). Importante frisar algo: clientes podem argumentar que sem estas tags habilitadas, não será possível fazer uso dos ColdFusion Components. Isso não é verdade, os mesmos poderão ser utilizados através da tag CFINVOKE.

Todas as demais tags, que não recomendamos desabilitar, não oferecem qualquer risco ao servidor e aos sites/aplicações existentes.

Sou um pouco radical quando questionado sobre o porquê de desabilitar as tags CFOBJECT, CFOBJECTCACHE e a função CreateObject. Respondo de forma bastante simples: “esta é uma hospedagem ColdFusion, não numa hospedagem Java ou qualquer outra tecnologia. CFML não é uma linguagem orientada a objetos, use CFINVOKE”. Sugiro fazer o mesmo para acalmar (ou atiçar) os ânimos de seu cliente.

Bem, até agora cobrimos 4 das 5 orelhas existentes na interface de configuração de sandboxes security. Vimos como proteger data sources, tags, funções e arquivos (nossa primeira explicação). Ficou faltando a quinta “orelha”, chamada “Server/Ports”. As configurações desta não são tão importantes em termos de segurança uma vez que o consumo de recursos externos não é muito diferente de se oferecer uma conta de FTP (o que com certeza todos terão) num servidor. O cliente pode colocar o que bem quiser dentro da sua conta. Entretanto, se você por acaso (questões de consumo de banda ou qualquer outro) resolver aplicar restrições nesta “orelha”, tome ciência de um bug de funcionalidade não documentado pela Macromedia que afeta a tag CFFTP em contas “sandboxeadas” - http://www.cfgigolo.com/archives/000326.html.

Na nossa configuração, a orelha “Server/Ports”, sem restrições, ficaria assim:



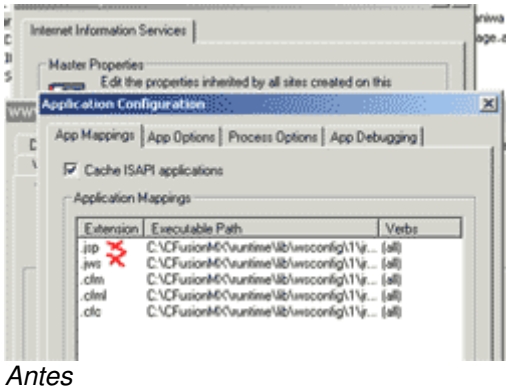
11. Considerações finais

Assim como qualquer tecnologia oferecida numa hospedagem compartilhada, existem muitas considerações a se fazer e detalhes a observar. Não é diferente com o ColdFusion MX Server. Muitas delas, envolvem performance e outras tantas segurança. A imensa maioria **não** foi coberta por este documento, que focou apenas a criação de sanboxes e a necessidade de se usá-las.

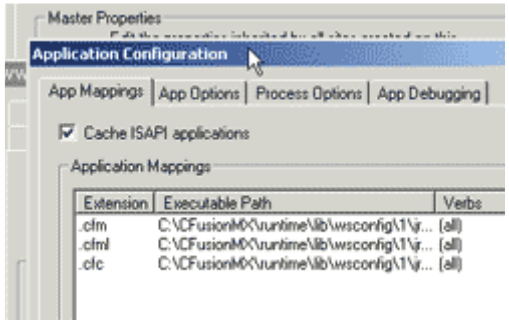
Leia a documentação do produto, participe de listas de discussão, leia os blogs nacionais e internacionais (principalmente), fique ligado em dicas e best-pratices encontrados em sites sobre o ColdFusion (ex: o site do CFUG-SP – http://www.cfugsp.com.br – oferece o download de apresentações sobre segurança em aplicações ColdFusion e também sobre performance neste).

Abaixo seguem algumas últimas recomendações importantes, à luz do que foi apresentado:

- O ColdFusion MX Enterprise permite que se rodem Java Server Pages (jsp) através do Jrun. Como estamos falando de uma hospedagem ColdFusion, é natural que estas extensões sejam desabilitadas no seu servidor web (IIS, Apache, etc). Remova todos os suportes à extensões que não sejam usados especificamente por aplicações ColdFusion. Nos meus servidores mantenho apenas os seguintes: .cfm, .cfml e .cfc e mais nada. Veja a imagem exemplo da configuração de extensões do IIS:



Antes



Depois de eliminar as entradas não utilizadas.

- Não preciso nem falar, mas vale o lembrete: instale todos os patches e hotfixes existentes para o ColdFusion MX. Se você estiver (e deve) usando a versão 6.1, neste link - http://www.cfgigolo.com/archives/000362.html - existe uma maneira rápida e simples de se fazer isso.

- Se você possui uma licença do ColdFusion MX (6.0), que foi a primeira versão da linha MX, saiba que você pode (e deve) atualizar esta versão, gratuitamente, para a mais recente, MX 6.1, que traz inúmeras correções e melhorias tanto em performance quanto em segurança. Faça o download desta nova versão no site da Macromedia, seu serial number do CFMX 6.0 irá funcionar para esta versão.