Segurança do ColdFusion MX 7 em ambiente compartilhado

Alex Hübner
alex@cfgigolo.com
Revisão 4 – 31-05-2005

1. Introdução

É cada vez mais comum encontrar empresas e serviços de hospedagem oferecendo suporte ao ColdFusion Server. Este é um sinal claro de que a tecnologia está crescendo cada vez mais no país. Contudo existe um problema com este crescimento rápido: há pouca documentação disponível no nosso idioma, a começar pelo manual oficial de administração do ColdFusion, que é totalmente em inglês. Tentando preencher esta lacuna da documentação em nosso idioma, este tutorial irá abordar conceitos básicos de segurança do ColdFusion Server funcionando em ambiente compartilhado (em provedores de hospedagem). Não precisamos dizer, mas nunca é demais lembrar que, como profissional, é sua obrigação dominar o idioma inglês para poder estar por dentro das últimas novidades da área. Com o ColdFusion não é diferente: se quiser conhecê-lo à fundo e ser um fera na plataforma, terá que aprender a ler em inglês. A aquisição da Macromedia pela Adobe tende a mudar isso, uma vez que a Adobe tem uma presença muito maior no Brasil. Esperamos que tutoriais oficiais e documentação voltem a ser publicados em nosso idioma, mas não fique esperando por isso, antecipe-se e saia na frente.

É preciso deixar claro que o objetivo deste tutorial não é ensinar administração de ColdFusion Server, muito menos cobrir todos os aspectos de segurança em servidores ColdFusion, pois estes não estão restritos ao produto. Estendem-se ao sistema operacional, ao servidor web, à rede (firewall e afins) e uma série de outros fatores que, somados, dariam um livro, não um tutorial. Trata-se, portanto, de uma pequena coletânea de *settings* que recomendamos para servidores de hospedagem compartilhados rodando o **ColdFusion MX 7** versão **Enterprise** ou versão **Developer** - veja mais adiante porquê excluímos a versão Standard desta lista. Todos os exemplos dados neste tutorial são baseados na plataforma Windows, mas são válidos para o Linux, Mac ou qualquer outra plataforma suportada pelo ColdFusion (são inúmeras - veja http://www.macromedia.com/software/coldfusion/productinfo/systemreqs/fp_frameset.htm) . Basta adaptar as informações (por exemplo, o formato de arquivos e pastas) de acordo com o sistema operacional utilizado por você e seguir em frente. Para este tutorial, é fundamental que você já tenha conhecimentos básicos de gerenciamento do ColdFusion Server. Caso você não conheça o dito cujo, sugerimos começar por aqui - http://www.cffaq.com/index.cfm?language=br.

Para tratar de quaisquer outros assuntos relativos ao ColdFusion Server - incluindo este tutorial - convidamo-lo(a) a participar da lista de discussão ColdFusion Brasil - http://www.coldfusion.org.br. Cadastre-se (lembre-se de ler as regras) e poste sua dúvida/crítica/sugestão/correção para que todos possam se beneficiar das respostas de todos.

2. Hospedagem compartilhada

Algo que precisa estar claro desde já: uma hospedagem compartilhada (não importa qual tecnologia você estiver usando - ASP, PHP, JSP, etc.) 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". A razão para isso é que, em tal ambiente, não existe isolamento total entre usuários e aplicações. Qualquer aplicação pode causar interferência numa outra aplicação e no servidor como um todo. Um exemplo clássico são ataques do tipo DoS (não confundir com DDoS), propositais ou (muitas vezes) não. Alguns programadores "mãos de chumbo" conseguem criar, sem perceber, códigos mal feitos, devoradores de CPU e memória, colocando o processador (ou processadores) do servidor - e os outros clientes locados no mesmo servidor - em apuros.

Caso você não abra mão de ter um ambiente absolutamente seguro e com desempenho garantido, livre de riscos diversos, deverá partir para uma solução dedicada ou semi-dedicada. O ColdFusion, a partir da versão MX (6.0), oferece a possibilidade de se configurar, 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 desempenho já que, nesta modalidade, cada instância tem sua JVM própria. Para os que não conhecem Java, o termo JVM é uma abreviação de "*Java Virtual Machine*", onde o software cria uma abstração própria que, grosso modo, funciona como se fosse uma máquina exclusiva e separada - conheça mais aqui http://www.infowester.com/jvm.php. 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 "semi-dedicados" ou mesmo "dedicados", ainda não oferecidos no Brasil (veja que boa oportunidade de negócio!) e em raros casos no exterior. Por isso não iremos tratá-la neste documento.

3. As duas versões comerciais do ColdFusion

Muitos não sabem, mas existem duas versões comerciais do ColdFusion Server e uma terceira, gratuita, destinada ao desenvolvimento de aplicações CFML para posterior *deploy* em um servidor (justamente o caso do desenvolvedor que desenvolve localmente e faz o *upload* para um servidor de uma empresa de hospedagem). A primeira versão, mais barata, é a *Standard* e a outra, mais cara, é a *Enterprise*. As diferenças? Para quem desenvolve praticamente nenhuma, uma vez que o suporte a CFML é idêntico, com pequenas diferenças e recursos extras na versão *Enterprise*. No entanto, para quem hospeda, as diferenças são muito importantes e significativas. A versão *Enterprise* oferece, por exemplo, suporte a um número maior de bancos de dados. Adicionalmente, esta versão suporta a instalação de múltiplas instâncias (já mencionadas aqui), *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 4.0, e alguns outros recursos interessantes. A tabela comparativa das diferenças entre as versões pode ser encontrada aqui: http://www.macromedia.com/software/coldfusion/productinfo/product_editions/fp_frameset.html. Se você está oferecendo ColdFusion em um 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", a *Standard*. Lembre-se de

que o barato pode sair caro: uma hospedagem compartilhada na versão *Standard* é arriscada. Exija do seu provedor de hospedagem a utilização da versão *Enterprise*, a única capaz de oferecer o recurso de *sandbox security* (que veremos adiante) e também de múltiplas instâncias. A versão gratuita, chamada de *Developer Edition*, oferece exatamente os mesmos recursos da versão *Enterprise*, porém, como se trata de uma versão de desenvolvimento, está limitada a responder apenas três números IPs, incluindo o *loopback* da própria máquina ("localhost - 127.0.0.1").

Normalmente (repetindo: normalmente) desenvolvedores ColdFusion são bem intencionados e não perdem tempo com tentativas de invasão e ou outras atividades virtuais ilícitas (ou "desafiadoras"). Além disso, para explorar vulnerabilidades intrínsecas ao ColdFusion Server, usando scripts CFML - não confundir com vulnerabilidades da sua aplicação, tais como *SQL Injection* e afins - 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. Mas nunca confie na boa índole - ou capacidade de fazer besteira - de todos eles, tenha o "corpo fechado", incluindo seus servidores.

4. 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 *sandbox* no ColdFusion Server pode ser explicado, grosso modo, com uma analogia à caixa de areia dos gatos. Nesta caixa isolada, seus "usuários" poderão fazer o que bem desejarem, de forma higiênica e limpa, sem sujar ou estragar o que estiver ao redor. O que isto significa na prática? Significa que tags do CFML como CFFILE, CFDIRECTORY e outras que, potencialmente podem causar estragos, podem ser habilitadas sem problemas ou preocupações. O usuário de uma *sandbox* devidamente configurada, não terá permissão para "sujar" a *sandbox* do vizinho ou qualquer outro local no servidor. Ele não vai conseguir fazer *upload* de arquivos, deletar, copiar ou criar diretórios fora da sua pasta raiz, fora da sua *sandbox*. Porém existem algumas pequenas exceções que iremos tratar mais adiante.

Ficou claro que toda e qualquer hospedagem compartilhada deve fazer uso de *sandboxes?* Este recurso já está habilitado no seu servidor ColdFusion? Não? Então vamos fazê-lo (lembre-se de que você poderá seguir os passos usando a versão *Developer*):

a. Entre no *ColdFusion Administrator* (dependendo da sua instalação, ele poderá estar em http://seu-servidor:8500/CFIDE/Administrator ou http://seu-servidor/CFIDE/Administrator);
b. No menu lateral esquerdo, encontre a opção "*Sandbox security*" (a última) dentro da área chamada "*Security*" (expanda o menu se este estiver colapsado);
c. Se você não habilitou antes, a opção "*Enable ColdFusion Security*" deverá estar desabilitada ("ticada"). Selecione-a e na seqüência clique em "*Submit changes*".

Uma vez completado os passos acima, o ColdFusion emitirá uma mensagem de sucesso e

criará duas *sandboxes* padrões do sistema.

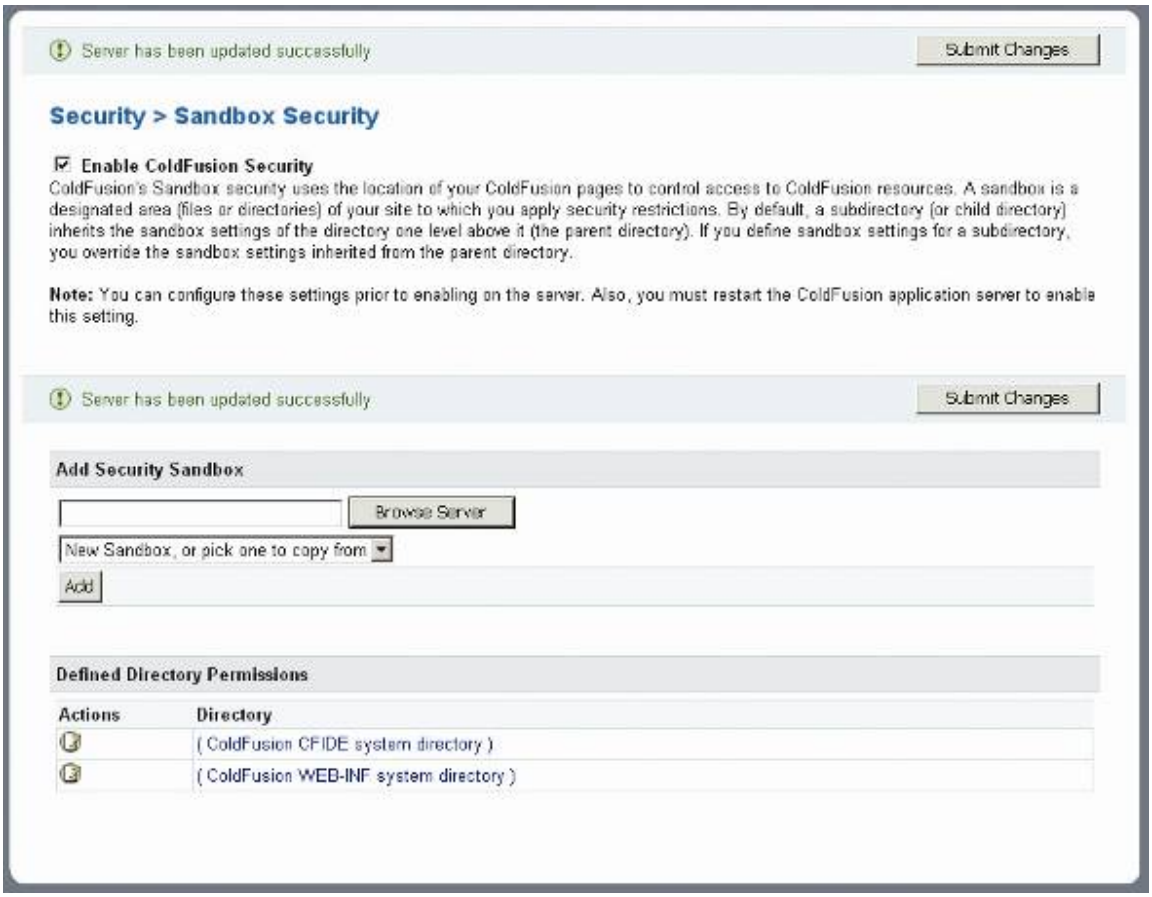

Figura 1 - O recurso sandbox security habilitado com as sandboxes padrões criadas

Estas duas *sandboxes* (“ColdFusion CFIDE” e “ColdFusion WEB-INF”) são necessárias para que o ColdFusion funcione perfeitamente. Por esta razão elas não podem (e não devem) ser removidas - note a ausência do ícone de deletar lado.

5. Criando novas *sandboxes*

O processo de criação de *sandboxes* é relativamente simples. Uma *sandbox* é criada/adicionada sempre sob uma pasta base ou raiz, na imensa maioria das vezes, sob 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 deve estar localizada 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* (para scripts .cfml) e *delete*) via CFML somente nos arquivos e pastas existentes sob sua pasta raiz (“D:\sites\alexhubner”), nada acima (subindo na estrutura de diretórios).

Diante de tal cenário, o processo de criação consiste nestes três passos simples:

a. Em "*Add Security Sandbox*", especifique o caminho completo para a pasta base do usuário. No nosso caso "D:\sites\alexhubner";
b. Mantenha a opção "*New sandbox or pick one to copy from*" selecionada e;
c. Clique no botão "*Add*".

Figura 2 - Criando uma nova sandbox

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

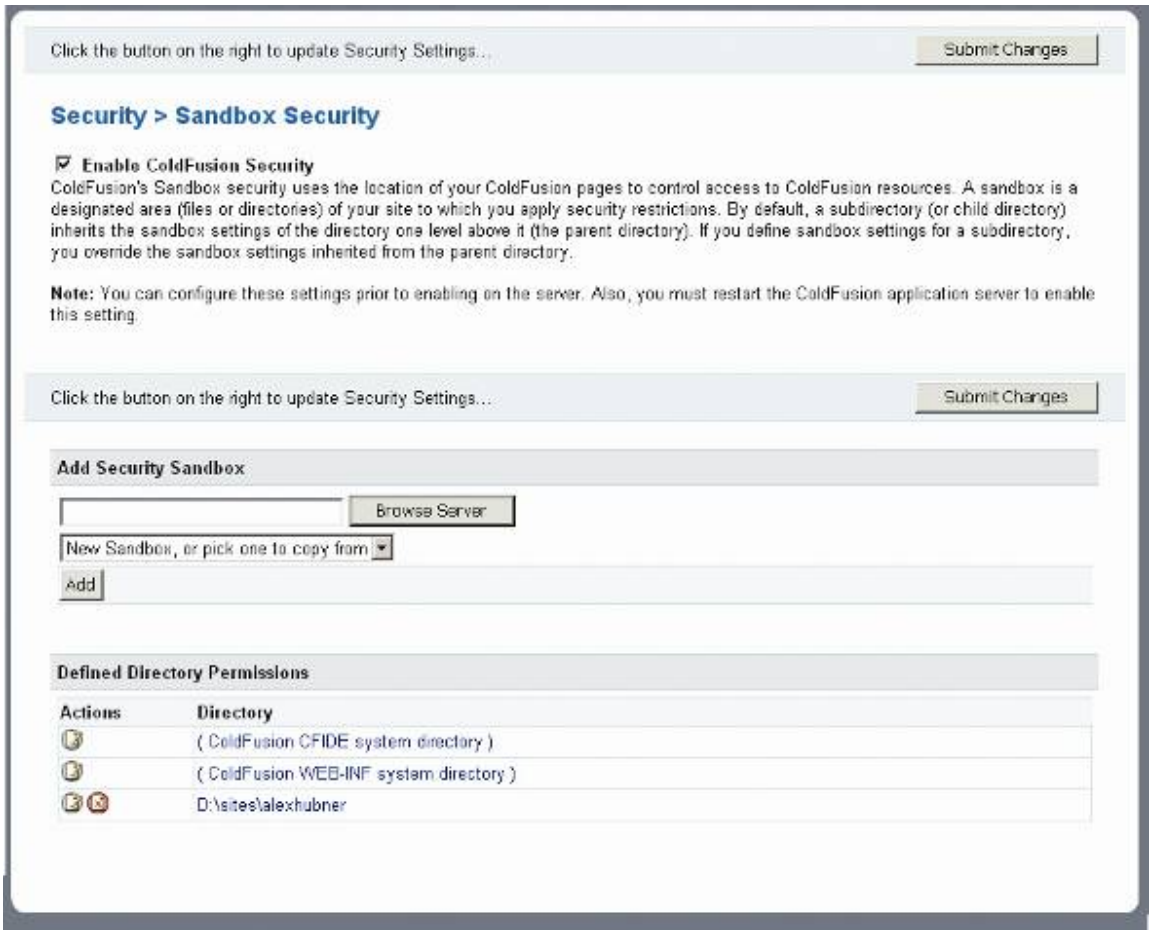

Figura 3 - Sandbox criada

O próximo passo será configurar em detalhes esta *sandbox*. É aqui que este tutorial começa a ser importante, na medida em que as configurações que propomos, visam a melhor segurança do servidor em ambientes compartilhados.

6. Configurando uma *sandbox* para ambiente compartilhado

Ao se clicar na *sandbox* em questão, veremos uma interface com cinco abas, a saber: "*Data Sources*", "*CF Tags*", "*CF Functions*", "*Files/Dirs*" e "*Server/Ports*".

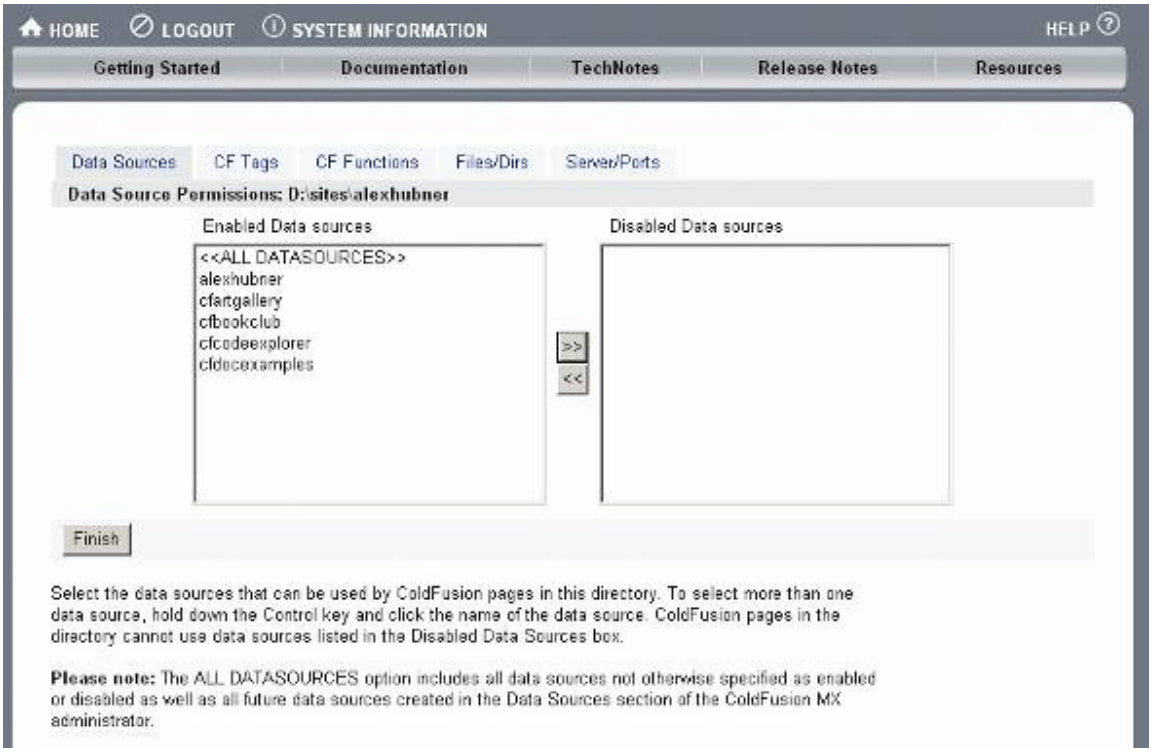

Figura 4 - Abas de uma sandbox

O esquema de funcionamento desta interface é bastante simples. Para todas as abas, na janela esquerda têm-se o que está habilitado e na janela direita, o que está desabilitado. Por padrão, cada *sandbox* recém criada terá permissão de acesso em todas as *Data Sources* existentes no servidor, bem como acesso a quaisquer tags e funções disponíveis no ColdFusion Server. Também poderá acessar qualquer recurso externo como, por exemplo, *Webservices*, servidores de e-mail, de autenticação LDAP, entre outros, através de consultas/acessos usando CFHTTP, CFFTP, CFPOP e afins. A exceção é a manipulação de arquivos que, por padrão, obviamente ficará limitada à pasta base do usuário (“D:\sites\alexhubner”). Um clique na aba “*File/Dirs*” evidencia isso.

Figura 5 - Configuração da pasta base do usuário (File/Dirs)

Perceba que existem duas entradas para a pasta do usuário. A primeira, com a pasta do usuário e a segunda, idêntica, porém com um hífen no final (“D:\sites\alexhubner\-“). Isto indica ao ColdFusion que a *sandbox* em questão tem autorização para ler, escrever, executar e deletar qualquer arquivo e pasta sob a pasta base (descendo na estrutura de diretórios). Eu recomendo adicionar uma terceira entrada, necessária para uma grande quantidade de aplicações CFML e eventualmente pelo próprio *runtime* do ColdFusion em operações de *upload* de arquivos pesados que, durante o processo de *upload*, precisam usar uma área de *swap* antes de serem destinadas à sua pasta final. A pasta temporária do ColdFusion no Windows - instalação padrão (“*Server Configuration*”) fica localizada em “C:\CFusionMX7\runtime\servers\coldfusion\SERVER-INF\temp\wwwroot-tmp\-“ (note que o hífen é necessário neste caso). Caso você tenha dúvidas sobre a localização da pasta temporária no seu servidor, execute o seguinte código CFML <cfoutput>#GetTempDirectory()#</cfoutput>, ele retornará a posição exata desta.

Para adicionar uma nova pasta com permissão de acesso na nossa *sandbox*, devemos proceder:

a. No campo “*File Path*”, entre com “C:\CFusionMX7\runtime\servers\coldfusion\SERVER-INF\temp\wwwroot-tmp\-“. Você também pode usar a opção “*Browse*” se desejar (mas não se esqueça de adicionar o hífen);
b. Clique em “*Add Files/Paths*”.

Note que você pode permitir acesso não somente a pastas, mas também a arquivos no servidor - suponha que o usuário necessite acessar um arquivo qualquer fora de sua pasta

base. O processo é idêntico, porém aponte, como *File Path*, o caminho completo do arquivo (incluindo seu nome e extensão). O uso de hífen não é necessário neste caso. Na verdade ele só deve ser usado para pastas.

Agora vamos dar início a um *tour* pelas abas e suas configurações recomendadas visando a maior segurança do seu servidor compartilhado.

7. Aba *"Data Sources"*

Na aba "*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*. Como já mencionamos, por padrão, uma *sandbox* recém criada tem permissão de acesso a quase tudo, inclusive a todas as *Data Sources* existentes. Por questões de segurança, obviamente deveremos permitir acesso somente as *Data Sources* pertencentes ao usuário cliente. Imagine que ele tenha direito apenas a uma *Data Source* (que aponte, por exemplo, para um arquivo *Access*, um banco no *MySQL* ou mesmo um banco no *SQL Server*) em seu plano de hospedagem e que esta *Data Source* se chama "alexhubner". A aba *Data Sources* ficaria desta maneira:

Figura 6 - Aba Data Sources permitindo acesso somente à Data Source do cliente

O procedimento mais recomendado neste caso é primeiro desabilitar todas as *Data Sources* (selecionando <<ALL DATASOURCES>> e clicando na setinha para direita, em direção ao campo "*Disabled Data sources*"). Depois, basta fazer o processo inverso, porém

somente para a *Data Source* em questão: “alexhubner”.

8. Abas “*CF Tags*” e “*CF Functions*”

Chegamos à parte mais importante deste tutorial. Você deve se recordar que, mais acima mencionamos o fato de existirem algumas exceções à segurança completa de uma *sandbox*. Justamente por conta destas exceções é que algumas tags e funções CFML precisam estar desabilitadas. Antes de prosseguir, é recomendado que você leia a documentação sobre a linguagem CFML (*ColdFusion Markup Language*) para entender a funcionalidade de cada uma das tags apresentadas aqui. Sugerimos conhecer o *CFML Reference* no site LiveDocs da Macromedia - http://livedocs.macromedia.com/coldfusion/7/htmldocs/part_cfm.htm

Na aba de “*CF Tags*”, recomendamos **desabilitar** as seguintes tags (e apresentamos os motivos):

- **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 desempenho caso o usuário tente acessar um executável problemático, que consuma muitos recursos do servidor;
- **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 com privilégio “System” ao registro do Windows, o que é arriscadíssimo, por isso recomendamos desabilitá-la;
- **CFSchedule:** o agendamento de tarefas é uma das grandes funcionalidades do ColdFusion Server, porém deve ser usada com parcimônia. Uma vez mal utilizada, pode trazer problemas de desempenho para o servidor. Em servidores compartilhados onde esta tag está habilitada é comum vermos *tasks* agendadas para rodar a cada minuto ou às vezes 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 desempenho e “abusos” de agendamentos com scripts pesados.

A aba “*CF Tags*” da nossa configuração ficaria assim:

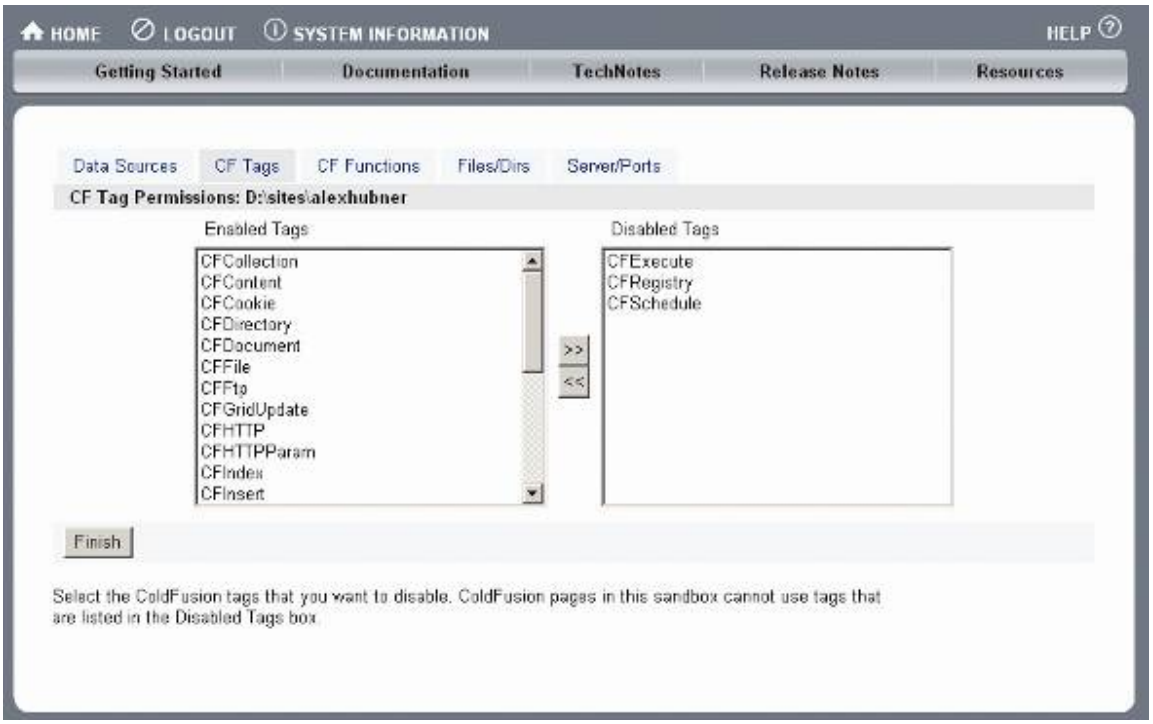

Figura 7 - Aba "CF Tags" permitindo acesso somente as tags seguras para usoem ambiente compartilhado

Na aba "*CF Functions*", recomendamos desabilitar as seguintes funções (e apresentamos os motivos):

- **CreateObject(Java):** leia a explicação no próximo item;
- **CreateObject(COM):** leia a explicação no próximo item;
- **GetProfileString e SetProfileString:** estas duas funções são raramente utilizadas em aplicações CFML. São usadas para, entre outras necessidades, configurar e ler arquivos de inicialização de aplicativos (arquivos com extensão .ini, entre outros), incluindo o próprio servidor ColdFusion. Recomendamos desabilitá-la, pois o seu uso é muito raro e pode oferecer um risco adicional desnecessário.

A aba "*CF Functions*" ficaria assim então:

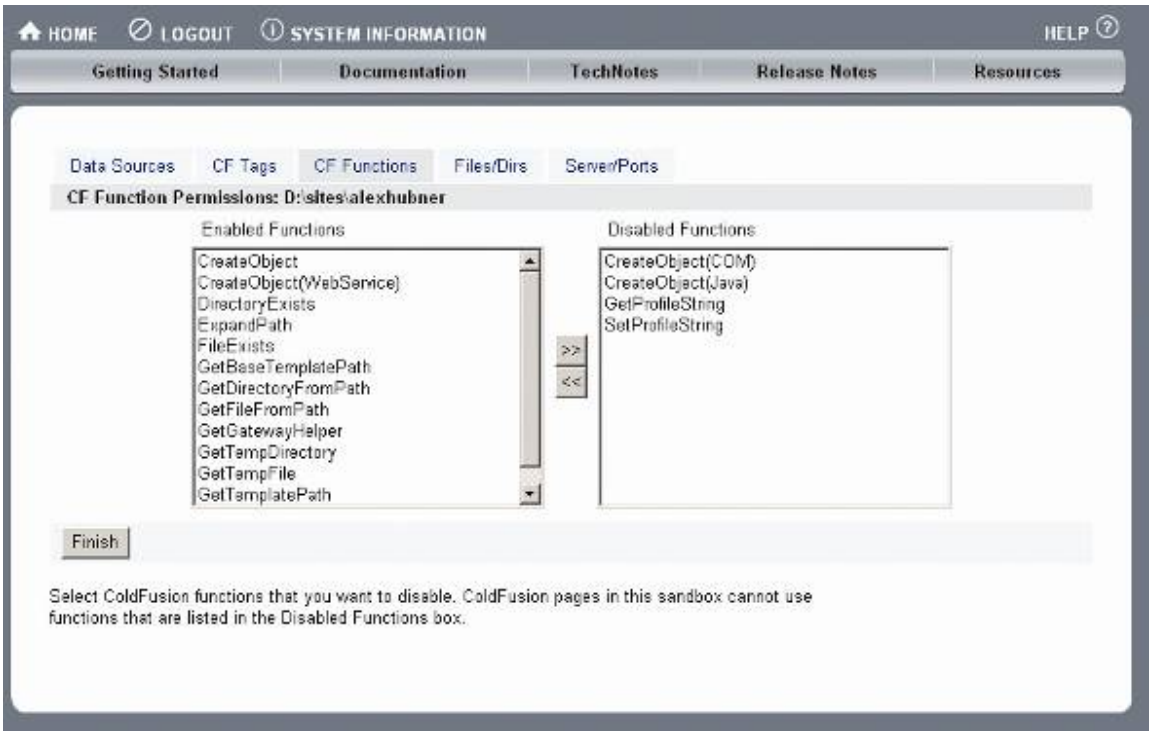

Figura 8 - Aba "CF Functions" permitindo acesso somente as funções seguras para uso em ambiente compartilhado

9. As funções CreateObject(Java), CreateObject(COM) e a tag CFOBJECT

No item anterior sugerimos que as funções CreateObject(Java) e CreateObject(COM) fossem desabilitadas. Com a mudança de plataforma para Java, o ColdFusion MX oferece aos seus usuários um leque enorme de possibilidades tais como integração total com a API Java, incluindo execução de classes, EJB, *Custom Tags*, JSP e muitos outros recursos nativos ou de terceiros, existentes na plataforma J2EE. Adicionalmente, o ColdFusion permite invocar componentes em COM para diversas finalidades e recursos não suportados nativamente pela CFML. Esta interação com Java e com componentes COM é feita usando-se as tags CFOBJECT e funções CreateObject(Java) e CreateObject(COM). 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 que não estarão isolados pela *sandbox*, tornando possível o acesso irrestrito a recursos, pastas e arquivos servidor. Também é possível acessar objetos e classes (e métodos) do próprio ColdFusion de uma forma muito íntima e insegura, permitindo a modificação de recursos tais como *Data Sources*, *Mappings* e afins. O problema não se limita à possibilidade de explorar e ler informações sensíveis, mas também de alterar, criar e deletar configurações diversas, uma vez que o uso desta tag não respeita a configuração de *sandbox security*. Isto não é um *bug* do ColdFusion, mas sim uma limitação do ambiente compartilhado, que utiliza/compartilha uma única JVM para todas as contas existentes na máquina. Você deve se lembrar de que comentamos, no início deste tutorial, sobre a possibilidade de se criar múltiplas instâncias de ColdFusion. Pois esta é única maneira de se oferecer suporte a estas funções que recomendamos desabilitar de forma irrestrita e com total segurança. Consulte o manual de administração e tutoriais do ColdFusion no site da Macromedia para maiores detalhes -

http://www.macromedia.com/devnet/mx/coldfusion/.

Você deve estar se perguntando para quê servem as outras duas funções CreateObject() e CreateObject(Webservices). Bem, a primeira é normalmente utilizada para se invocar *ColdFusion Components* (CFCs) e a segunda para se invocar *Webservices*, que não oferecem riscos ao servidor se forem habilitadas. Pelo contrário, deixá-las habilitadas (especialmente a CreateObject()) é fundamental para que o desenvolvedor tenha a possibilidade utilizar toda a potencialidade da CFML, através de componentes CF, incluindo programas prontos e frameworks tais como Fusebox e MachII. Até a versão MX 6.1, os administradores de servidores ColdFusion eram obrigados a desabilitar por completo a tag CFOBJECT e a função CreateObject() – que era única e compreendia todas as possibilidades de chamar um objeto no ColdFusion, fosse ele um simples CFC ou uma classe Java nociva. Isso limitava muito as possibilidades de desenvolvimento, o que foi solucionado na versão 7.0, através do desmembramento da CreateObject() em quatro funções distintas: CFCs, Java, COM e *Webservices*.

É muito importante notar que as restrições aplicadas às funções CreateObject() são imperativas à tag CFOBJECT (que também pode ser usada para se invocar componentes, classes e afins). Se, por exemplo, você desabilitar a função CreateObject(Java), o usuário não poderá invocar classes Java através da tag CFOBJECT, o mesmo vale para os outros três casos (COM, CFCs e *Webservices*). Por isso mantenha a tag CFOBJECT habilitada, pois estarão aderentes às restrições aplicadas nas funções CreateObject().

Até agora cobrimos quatro das cinco abas existentes na interface de configuração de *sandboxes*. Vimos como proteger *Data Sources*, *Tags*, *Funções*, *Arquivos e Pastas*. Ficou faltando última e quinta aba, chamada "*Server/Ports*".

Esta aba permite controlar as permissões de acesso a recursos externos tais como servidores FTP, HTTP, *Webservices*, SMTP, POP e outros. As configurações desta aba não são tão importantes em termos de segurança, uma vez que o consumo de recursos externos não é um "risco" muito diferente do risco que se corre ao se oferecer uma conta de FTP (o que com quase certeza todos terão) num servidor compartilhado. O cliente poderá colocar o que bem quiser dentro da sua conta. Entretanto, se por um acaso, tais como questões de consumo de banda ou qualquer outro relevante, resolva-se aplicar restrições, tome ciência de um *bug* de funcionalidade não documentado pela Macromedia e que afeta o uso da tag CFFTP em contas "sandboxeadas" - http://www.cfgigolo.com/archives/2003/11/bug_com_cfftp_e.html.

10. Conclusão

Assim como qualquer tecnologia oferecida numa hospedagem compartilhada, existem muitas considerações a se fazer e detalhes a se observar. Muitas delas envolvem desempenho e outras tantas a questão segurança. A imensa maioria **não** foi coberta por este tutorial, que focou apenas a criação de *sandboxes* no ColdFusion Server e a necessidade de se utilizá-las. Por isso leia muito atentamente a documentação do produto, participe de listas de discussão, leia *blogs* internacionais e nacionais. Sobre estes últimos, sugerimos a leitura do CFGIGOLÔ – http://www.cfgigolo.com, que tem *posts* extensivos sobre

segurança e desempenho de servidores ColdFusion, que lhe dará embasamento necessário para seguir os próximos passos de administração de servidores ColdFusion. Fique ligado em dicas e *best-pratices* encontrados em sites sobre o ColdFusion. Acima de tudo: saiba como usar o produto em sua plenitude, não apenas a linguagem de programação CFML, mas também no produto que interpreta a CFML: o fascinante ColdFusion Server. Nos vemos na próxima!