

Uma visão prática sobre segurança em Aplicações com o Firebird

Pensando em segurança

Firebird instalado, e tudo resolvido. Certo? A menos que você esteja pensando em fazer um aplicativo para uso próprio sim. Se você vai distribuir a aplicação, é bom preocupar-se com o controle de acesso. Como 99% dos desenvolvedores querem fazer uma aplicação para o "mundo", pensar em segurança e controle de acesso é uma parte importante de qualquer projeto de desenvolvimento.

Apesar do Firebird oferecer a possibilidade da criação de usuários para os bancos de dados e da criação de perfis (ROLES), isto por si só não garante a segurança das informações. Há fatores adicionais que pesam muito e que precisam ser considerados, além da própria política de criação de usuários e perfis de acesso. Por exemplo, redes baseadas em TCP/IP podem ser alvo de ataque externo. Você pode ter um usuário mal-intencionado que queira "sabotar" ou mesmo ter acesso à áreas restritas dos dados.

Algumas perguntas precisam ser feitas para se definir os passos, mas a mais importante de todas é: Até onde devo ir com segurança de acesso? Para alguns isto chega a beira da paranóia, para outros um relaxamento total. Como o "caminho do meio" é sempre o melhor, não devemos pecar nem pela falta, nem pelo excesso.

E este é o objetivo deste artigo: expor uma visão prática da segurança que pode ser implementada numa aplicação com o Firebird, algumas alternativas, e métodos que podem ser usados.

Você já ouviu algo assim: "*Meus dados não estão mais aqui, como isto foi acontecer?*" ou, "*Puxa, fulano de tal saiu da empresa e levou/apagou uma cópia dos arquivos confidenciais...*" ou ainda, "*Este relatório não podia ter caído na mão do fulano...*" Se você já ouviu isto alguma vez ou não quer ouvir, recomendo que você leia este artigo.

Vamos combinar uma coisa antes de prosseguir: esta é uma visão particular do assunto, baseado na experiência de alguns anos convivendo com o dia-a-dia do usuário e do desenvolvimento. Não é um guia DEFINITIVO sobre o assunto, não é um artigo pra ensinar a programar, com código fonte publicado, e não se aprofunda em questões muito específicas (poderão ser alvo de outros artigos). Mas é uma visão que pode lhe ser útil (assim espero).

Aperte o cinto, e boa leitura.

Quais os Fatores de Risco para minha aplicação?

Esta pergunta é determinante para definir as regras gerais de controle de acesso. Estes fatores não dependem de você em quase nada, e são fatores limitadores ou a chave para o sucesso de sua política de segurança. Dependendo do ambiente, quebrar a cabeça com a segurança é quase inútil.

Podemos analisar isto sob vários aspectos, mas os principais itens que fazem diferença são:

1. Plataforma do servidor (Windows ou Linux)
2. Servidor dedicado ou não-dedicado?
3. Visibilidade do servidor (intranet ou internet)
4. Número de usuários
5. Uso de vários bancos de dados separados
6. Existência de aplicações de terceiros no mesmo servidor

Como se proteger (nível de “paranóia”)

Chamarei de "nível de paranóia" porque estes itens estão na quantidade de providências que você vai querer tomar no que diz respeito à segurança e dependem só de você. São eles:

1. Política de segurança de acesso ao Servidor/Rede
2. Arquitetura e quantidade de aplicações/módulos que acessarão o banco
3. Proteção anti-cópia e contra engenharia reversa de seu banco de dados
4. Requerimentos de segurança do usuário

A combinação entre todos estes elementos, determinará o quanto sua solução é segura, e o quanto ela está exposta. Vamos dar uma examinada em cada um dos aspectos mencionados e qual sua relação com segurança. Você poderá a partir do momento que conhecê-los melhor, avaliar situações de risco, e quem sabe poder contorná-las com as medidas de segurança apropriadas.

Considerando os fatores de risco para uma Aplicação e o Firebird

Fator de Risco 1: Sistema Operacional

Cada sistema operacional trata segurança de modo diferenciado. É preciso ter uma noção do que pode ser chamado de S.O. seguro.

Basicamente um sistema operacional seguro, precisa oferecer mecanismos de controle de acesso à dispositivos do equipamento (discos, diretórios, arquivos, sistemas, impressoras, etc) à pessoas autorizadas, através de procedimentos de identificação da pessoa que está na máquina, mediante senhas. Podem existir usuários com diferentes perfis, com acesso limitado à certos locais e aplicações. Isto pode ser definido localmente ou por meio do compartilhamento em rede.

Num nível um pouco mais alto, um sistema operacional seguro é aquele que oferece um isolamento de execução às aplicações, permitindo que rodem como serviços, isto é, numa camada abaixo da sessão iniciado pelo usuário. Assim, mesmo que o usuário encerre a sua sessão de trabalho, ele seguirá rodando normalmente.

E em nível mais alto ainda, pode ser considerado seguro um sistema operacional que trabalhe com múltiplas threads (linhas de processamento), de forma que se houver algum travamento em uma delas, as demais possam seguir trabalhando normalmente e até permitirem o encerramento de threads problemáticas (trancadas).

Sendo assim, dependendo do sistema operacional adotado, você terá mais ou menos segurança implementada. E o Firebird e sua aplicação ficarão com as limitações impostas por ele.

Abaixo uma pequena tabela de sistemas operacionais e os diferentes aspectos de segurança possíveis:

Característica	Windows 9x/Me	Windows NT/W2K/XP	Linux / Unix
Acesso somente de usuários autorizados (login)	Não	Sim	Sim
Restrição à Diretórios e Arquivos em modo local	Não	Somente Diretórios	Diretórios e Arquivos
Nível de Compartilhamento	Somente Diretórios	Somente Diretórios	Diretórios e Arquivos
Execução de Serviços	Não	Sim	Sim
Múltiplas Threads	Não	Não	Sim

Percebemos assim, que antes de pensar em técnicas de segurança, você deve saber em que sistema operacional você estará instalando o Firebird e a aplicação.

Se a sua escolha recair sobre plataformas mais vulneráveis, você terá que conviver com certos riscos imagináveis: cópia ilegal, exclusão de dados indevida, acesso não autorizado, e pouco poderá ser feito para evitar isto.

Portanto se você quer garantir proteção aos dados, adote um Sistema Operacional que lhe dê condições de fazer isto.

Pondere muito sobre este aspecto, pois há sérias limitações nos sistemas operacionais Windows 9x/Me, além do conhecido histórico de instabilidade, que diminui sensivelmente nas plataformas Windos NT/W2K.

No entanto, você pode ter que adotar uma solução menos segura em função do próximo fator que discutiremos.

Fator de Risco 2: Servidor dedicado x Servidor não-dedicado

A decisão de dedicar uma máquina ao servidor Firebird passa inicialmente pela restrição de orçamento e pelo desconhecimento das vantagens que isto pode trazer. Existem alguns artigos que discutem as vantagens de se dedicar o servidor, por isto não vou me ater nestes aspectos, apenas nas questões relativas à segurança.

Em um ambiente mono-usuário, fica mais difícil justificar a necessidade de dedicar um servidor, embora possa se ganhar performance, pois distribuimos o processamento e o acesso à disco. Fica apenas a questão do S.O. *versus* restrição de acesso, isto é, se você escolher um S.O. Windows 9x/Me, qualquer pessoa senta na frente da máquina e pega o que quiser. Nas plataformas NT/W2K/XP, já é possível ficar mais tranquilo, e com o Linux nem se fala. Dedicar o servidor pode diminuir bastante a chance de problemas, pois você não terá um usuário disputando recursos e trabalhando “em cima” do servidor e de seus dados. Isto pode representar a diferença entre os dados estarem ou não onde deveriam.

Fator de Risco 3: Visibilidade do Servidor

Dependendo da grau de visibilidade do servidor, ele poderá estar mais ou menos exposto à acesso ou ataques de usuários indesejados.

Você pode por exemplo, desejar que os dados sejam visíveis somente ao ambiente local, dentro da empresa. Isto significa que a máquina onde está o servidor não deverá ser visível por equipamentos “estranhos”. Digamos porém que você dispõe de um acesso direto à Internet e precisa compartilhar acesso ao banco de dados Firebird com suas filiais, ou disponibilizar acesso à clientes remotamente. Bem então, se faz mais necessário o uso de técnicas de segurança.

Fator de Risco 4: Número de usuários

Uma questão sem importância para alguns, e crítica para outros. O número de usuários que terão acesso a um servidor pode representar uma dor-de-cabeça não só pela preocupação com a performance, mas com os aspectos de segurança.

A partir do momento em que o ambiente torna-se confiável (acesso apenas por máquinas aprovadas) há uma maior dificuldade de identificação de quem é o “sabotador”. Se um usuário é autorizado a acessar o banco de dados, presume-se que ele tem este direito, por isto, as medidas de isolamento do servidor nem sempre bastam. Numa grande corporação podem haver problemas com interesses divergentes, e é preciso monitorar e restringir o acesso mesmo internamente.

Fator de Risco 5: Uso de vários bancos de dados separados

Pode haver a necessidade de um banco de dados independente por atividade, setor ou departamento. Apesar algo pouco usual, uma vez que o uso de bancos de dados distintos gera processos adicionais de backup e manutenção, algumas vezes é este o cenário. Se não houver uma política de restrição de acesso por usuários, grupos de usuários ou perfis, os dados estarão disponíveis a todos que acessem o Servidor.

Fator de Risco 6: Existência de aplicações de terceiros no mesmo servidor

Por estar tornando-se cada vez mais comum a existência de aplicações “powered by” Firebird, é bem provável que em dado momento sua aplicação tenha que conviver com aplicações desenvolvidas por terceiros.

Isto pode tornar-se um problema se as políticas de segurança forem distintas ou conflitantes, especialmente na questão de nomes de usuários, que terão acesso aos bancos de dados. Por exemplo, se seu colega não tiver o cuidado de utilizar um usuário específico para as aplicações dele e você também não, isto é, se os dois utilizarem o SYSDBA (super-usuário do Firebird) com senhas de acesso distintas, teremos uma das aplicações inoperante.

Implementando Segurança – Administrando o “nível de paranóia”

Paranóia 1: Política de segurança de acesso ao Servidor/Rede

Este é um item fundamental a definir. Sem uma definição correta, as demais medidas poderão ser inúteis.

Se existir a figura de um administrador de rede, será necessário uma boa palestra com ele, sobre como é a atual política de segurança, quais as necessidades e expectativas de sua aplicação no que diz respeito à proteção. Não havendo esta pessoa, (normalmente não há) é preciso planejar bem os passos, conhecer quais as intenções da empresa, e expor a importância da segurança, e o qual o impacto de não adotar estas medidas. Também é preciso saber o que está sendo utilizado atualmente e sondar como serão implantadas novas aplicações no futuro, principalmente se quem irá fazer isto compreende e conhece estas normas.

Os cuidados básicos que se devem ter com o servidor, são:

Impedir acesso ao equipamento e às áreas do sistema operacional

Qualquer usuário que tiver acesso físico ao equipamento poderá indevidamente tentar acessar a máquina. Portanto é indispensável que o computador ao ser iniciado, identifique e autentique o usuário.

Isto pode ser feito, mesmo antes da carga do S.O. através da colocação de uma senha na BIOS da máquina, que só inicia após a informação da senha. Algumas máquinas permitem que sejam informadas duas senhas: uma de Supervisor e outra de Usuário. A diferença entre elas é que o usuário pode trabalhar com a máquina, mas não pode modificar nenhuma especificação da BIOS (mudar senhas, configurações, etc). Esta é uma saída para quem estiver com S.O. Windows 9x/Me, muito embora também seja possível “burlar” este mecanismo, através do descarregamento da memória da BIOS, ou transferindo-se o disco para outra máquina.

Se o sistema operacional escolhido não oferece uma autenticação segura de usuários, existem ferramentas que implementam uma segurança um pouco melhor, como por exemplo o POLEDIT (editor de Diretivas – vide anexo) do NT, ou o [TweakAll](#), que permitem restringir o acesso ao Windows 9x/Me, mas não impedem que seja dado um boot em modo de segurança e com isto ter acesso a todos os arquivos locais, além de não serem nativas da plataforma. Mas podem ajudar, se você não tiver escolha, diminuindo sensivelmente alguns riscos.

Outra coisa a se cuidar, é que o usuário comum seja impedido de alterar configurações de segurança do S.O., desinstalar programas, ou apagar componentes de funcionamento (drivers, programas de manutenção e de proteção anti-vírus). Como já dito antes dependendo do S.O. isto pode ou não ser feito, mas as ferramentas mencionadas oferecem a possibilidade de senão impedir totalmente, dificultar esta ação.

Impedir acesso às áreas do Firebird

Na área onde encontra-se instalado o Firebird estão além do próprio servidor, o arquivo de configuração (ibconfig nas plataformas Windows e isconfig no Linux) e a base de dados de usuários válidos para acesso aos bancos de dados disponíveis (isc4.gdb). Portanto, deve-se evitar que um usuário comum tenha acesso à estes arquivos, pois se ele substituir a base de dados de usuários por outra que ele conheça a senha, facilmente ele acessa qualquer um dos bancos de dados disponíveis na máquina.

Impedir acesso direto ao banco de dados

Para enxergar um banco de dados Firebird não é necessário que o usuário tenha qualquer permissão de gravação ou leitura nos arquivos físicos que o componham. Mesmo que sua aplicação precise fazer mudanças estruturais no banco de dados, isto

pode ser feito sem acesso direto ao arquivo. Quem deve ter estas permissões é o próprio servidor, que acessa os bancos de dados localmente, isto é, um Servidor Firebird não acessa bancos de dados remotos.

Por isto é uma importante providência restringir o acesso ao local/locais (no caso do uso de multi-arquivos) onde está o banco de dados. Isto impedirá cópia indevida, exclusão indevida, ou danos à base.

☞ *Rodar o Firebird como serviço, não como aplicação*

Como já explicado no Fator de Risco 1, se o Firebird rodar como serviço ele está protegido se o usuário trancar a máquina, ou efetuar o logoff. Manter o servidor disponível todo o tempo também faz parte da política de segurança.

☞ *Se o servidor for utilizado para acessar a Internet*

Esta é uma máquina que acessa redes externas (internet ou outras redes privadas)? Se realmente não for possível evitar isto, é preciso utilizar um Firewall (muro de proteção), por meio hardware ou software que impeça o acesso não autorizado, através de bloqueios. E mesmo que a máquina Firebird não esteja conectada diretamente à uma rede externa, mas se alguma máquina da rede tiver acesso externo sempre é interessante adotar um Firewall. Algumas dicas interessantes sobre o assunto podem ser encontradas no artigo "[Protegendo seu Servidor Firebird](#)" de Mark O'Donohue.

☞ *Se o servidor tem de ficar disponível à máquinas remotas pela Internet*

O resposta é implantar um Firewall, e ainda mais: um túnel seguro entre os pontos de acesso e o Firebird.

Existem algumas ferramentas como o [Zebedee](#), que implementam esta funcionalidade. Ele oferece um modo seguro de conexão entre o Servidor e os clientes, através da criptografia de dados, conexão por chaves de segurança e ainda ajuda na performance, pois compacta os dados que estão trafegando. O único "senão" é que por não ser um dispositivo "nativo" do Firebird é necessário distribuir junto da sua aplicação um módulo cliente que se conecta ao módulo no servidor. Desta forma o Firebird fica protegido "atrás" do Zebedee, podendo ficar em outra máquina, o que é inclusive uma maneira ainda mais eficiente de garantir segurança. Há um tutorial disponível, que lhe ajudará bastante com o Zebedee, chamado "[Acesso Firebird via Internet – A utilização do Zebedee como túnel de compressão seguro](#)".

Estas recomendações são válidas para em ambiente mono e multi-usuário. Em ambientes multi-usuário, estes riscos são potencializados, por isto é necessário fazer um plano de instalação que restrinja o acesso não autorizado.

Consulte três artigos que escrevi que fornecem mais detalhes sobre este tópico:

☞ [Escolhendo e configurando uma máquina para Firebird](#)

☞ [Escolhendo e configurando o sistema operacional do Servidor Firebird](#)

☞ [Montando um ambiente confiável para uso do Firebird](#)

Paranóia 2: Arquitetura e quantidade de aplicações que acessarão o banco

Dependendo da maneira como as aplicações são desenvolvidas (tecnologia e metodologia), e a quantidade delas, assegurar a segurança pode ser uma tarefa hercúlea. O Firebird é um SGDB que provê maneiras fáceis de se construir aplicações verdadeiramente Cliente-Servidor.

No entanto, se você não utilizar estes recursos no seu projeto de forma plena, você poderá deixar o servidor vulnerável. Se você estiver desenvolvendo aplicações de 2 camadas (cliente/servidor) o indicado é que você use e abuse dos recursos do servidor, transferindo à ele tudo o que for possível, em vez de estar na aplicação: processamento, controle de integridade referencial, ações automáticas, baseadas em eventos com os dados (triggers), geração de relatórios de atividades (logs) e até controle de acesso. Isto simplifica a codificação da aplicação e cria as regras para que quando aplicações “estranhas” acessarem o seu banco de dados, ele saiba “proteger-se” e delimite a ação sobre os dados.

Vários controles podem ser implementados no seu banco de dados, no servidor e nas aplicações para garantir uma computação segura. Vamos discutir algumas destas técnicas, e os motivos pelos quais estas podem ser úteis:

☞ Registro dos acessos ao banco de dados (log de conexões)

É possível implementar isto de várias formas, no entanto, prefira sempre os métodos que possam ser usados diretamente pelo servidor (server-side) pois, não há garantias que só uma determinada aplicação vai acessar o servidor. Por exemplo, um usuário mais avançado pode através de uma ferramenta de acesso/manutenção (IBExpert, por exemplo) abrir o banco de dados e fazer um belo estrago. Isto é bastante possível, pois estão cada vez mais populares as ferramentas gráficas para acesso às bases Firebird, o que tornam muito amigável o trabalho direto com as bases, dispensando conhecimentos técnicos profundos.

Uma maneira de adicionar um controle destes, é utilizar um aplicativo de monitoramento de conexões e que possa gerar um registro (log) de conexões efetuadas. Através de um exame periódico destes registros é possível perceber quem está acessando o servidor, quando, e a quantidade de acessos. Há um aplicativo interessante que gera um arquivo de log, que pode ser incorporado ao banco de dados em forma de tabela externa, permitindo assim visibilidade por dentro de uma aplicação recuperar estas informações remotamente. Esta ferramenta chama-se [IBConSvc](#) e pode rodar como serviço no Windows NT/W2K.

Um outro modo de gerar esta informação é efetuar algumas modificações no banco de dados de usuários do Firebird o isc4.gdb, implementando a geração de um log. Há um artigo de Ivan Prenosil sobre este assunto, chamado “[Segurança - ISC4.GDB Otimizado](#)”. Consulte-o para maiores detalhes. A única desvantagem deste modo é a possibilidade de conflito com aplicações de terceiros, caso este arquivo seja sobreposto indevidamente.

☞ Restringindo acesso à bancos distintos

Quando há vários bancos de dados dentro da empresa, pode-se utilizar um servidor para cada departamento ou ainda, trabalhar com as restrições de acesso implementadas em cada banco de dados, concedendo direito à acesso para usuários específicos do Firebird.

Para isto pode-se criar um usuário padrão para cada departamento e as aplicações de cada departamento irão utilizá-lo. Pode ser mais conveniente criar-se um perfil (ROLE) padrão e associar determinados usuários a estes perfis. Lembre-se que no caso de trabalhar-se com perfis, todas os componentes que não estiverem limitados terão visibilidade por qualquer usuário.

Paranóia 3: Proteção anti-cópia e engenharia reversa de seu banco de dados

Sob este tópico há muitas considerações a serem feitas, mas todas envolvem técnicas de programação ou a utilização de produtos de terceiros. Portanto se o seu interesse é aprofundar-se no assunto é bom pesquisar bastante e considerar qual o impacto de cada solução.

Se falarmos em proteção anti-cópia temos que separar em duas coisas:

☞ *Proteger as aplicações contra cópia ilegal*

Normalmente a aplicação estará instalada no computador cliente, o que torna o controle sobre cópias algo mais complexo. Algumas técnicas que podem ser adotadas:

- ☞ Utilizar o Registro do Windows para guardar informações específicas do equipamento (número da CPU, endereço de IP, personalização do usuário), no momento da instalação. Se estes valores não forem encontrados, ou não coincidirem, impede-se a execução da aplicação.
- ☞ Guardar informações em arquivos locais para comparação na inicialização da aplicação também pode ajudar, pois mesmo que o registro seja copiado, pode-se utilizar técnicas de endereçamento do arquivo, que ao ser copiado muda de endereço.
- ☞ Guardar informações sobre as máquinas clientes no banco de dados, em vez de localmente, pode ser mais interessante, pois uma máquina não cadastrada no banco de dados, não executa a aplicação.
- ☞ Uso de chaveadores de cópia, que impõe o uso de um hardware conectado à uma porta de comunicação.

Todos estes métodos podem trazer alguma desvantagem. Por exemplo, digamos que você tem um ambiente com 50 estações de trabalho. A necessidade de proteger suas aplicações pode representar uma dificuldade maior no processo de instalação, e mesmo no uso dos sistemas, pois quando um equipamento precisar ser substituído, precisa haver um modo de fazer isto rápida e facilmente, e estas proteções podem representar uma dificuldade a mais.

☞ *Proteger o banco de dados contra cópia ilegal*

O melhor modo de fazer isto é negar o acesso ao arquivo de banco de dados. Assim ninguém conseguirá extraí-lo indevidamente. Outro cuidado é limitar os usuários autorizados a efetuar o backup, pois um backup pode ser facilmente restaurado em outro local.

Uma técnica adicional pode ser a criptografia dos dados armazenados, porém isto deve ser utilizado com muita cautela, pois além de impor um trabalho de criptografar/descriptografar os dados gerando mais tempo, isto afetará os índices e impedirá que os dados sejam visualizados corretamente por ferramentas de manutenção. Portanto o conselho é reservar esta técnica apenas para dados muito restritos.

Digamos porém que seu banco de dados foi copiado. Certamente será um pouco incômodo se ele for examinado por concorrentes, não é? É possível evitar isto? Sim. A técnica é ocultar o metadata do banco de dados, de forma a impedir a extração do mesmo. Isto não afetará a funcionalidade do banco de dados, no entanto, é preciso saber que uma vez aplicado este procedimento, os metadados não poderão mais ser recuperados daquele banco de dados.

Para fazer isto basta uma pequena alteração em uma tabela de sistema do banco de dados, podendo-se escolher o que será ocultado: Stored Procedures, Triggers e/ou Views. Dê uma examinada no Tutorial "[Protegendo sua Base de Dados Interbase/Firebird](#)" para obter mais informações.

Paranóia 4: Requerimentos de segurança do usuário

Aqui praticamente não há limite nas técnicas que se possam utilizar. Dependendo da vontade do usuário, isto pode realmente impor o desenvolvimento de um sistema de segurança paralelo à aplicação.

Entre as coisas que usualmente são solicitadas, poderíamos citar:

- ∞Uso de logs de alteração e exclusão de registros
- ∞Controle de acesso por níveis na aplicação, podendo delimitar-se qualquer operação disponível na aplicação.
- ∞Expiração periódica de senhas de acesso.
- ∞Mesmo o usuário tendo acesso à certas operações, só serem confirmadas com a autorização de um superior.
- ∞Logoff automático após certo tempo de inatividade.
- ∞Controle de acesso por dias/horários específicos.
- ∞Backup automático em intervalos periódicos ou após determinado número de operações.
- ∞Geração de logs de erros de execução e/ou operação.
- ∞Replicação de dados síncrona ou assíncrona.

Todas estas técnicas também tem impacto sobre a performance, operacionalidade e podem ou não surtir efeitos positivos, dependendo da estrutura da organização. O planejamento destes esquemas de segurança deve ser feitos criteriosamente.

Algumas das técnicas mencionadas, podem ser executadas no lado Servidor, como Backup Automático, Replicação, Geração de Logs de alteração/exclusão, controle de acesso em dias/horas específicos e expiração periódica de senhas. Isto tudo pode ser feito com o uso de ferramentas, UDFs ou projetado no seu banco de dados em conjunto com o ISC4.gdb.

Lembre-se de que tudo o que for implementado pelo lado da aplicação não surtirá efeito se for feito acesso direto ao banco de dados com ferramentas de terceiros. É importante portanto, saber separar o que é segurança em nível de aplicação e segurança em nível de banco de dados. Processos menos críticos podem ficar na aplicação, já questões mais delicadas devem ser controladas pelo Firebird. Para isto será preciso conhecer bem o projeto, e quais as especificações de segurança requeridas.

Visão prática: Escolhendo as técnicas e criando um modelo seguro

Veremos a seguir um modelo de segurança funcional, que pode evidentemente ser aperfeiçoado, de acordo com as necessidades do seu cliente/aplicação.

Vamos criar um “cenário” de trabalho para poder discutir os aspectos de segurança. Imaginemos o seguinte cenário:

Um cliente deseja instalar as aplicações de Vendas, Faturamento, Controle de Estoques, Contas à Pagar/Receber. A estrutura da empresa está dividida em 4 grandes setores: Administração, Almojarifado, Balcão de Vendas e Tele-Vendas. A empresa dispõe da seguinte infra-estrutura de equipamentos: 1 servidor e 7 estações de trabalho: 2 na administração, 1 no almojarifado, 2 no balcão e 2 nas tele-vendas. Não importa no momento determinar se todos estes equipamentos estão adequados, se há alguma insuficiência. Estaremos apenas olhando sob o aspecto de segurança.

Supondo que você utilize o Borland Delphi para desenvolver as aplicações, e que no servidor lhe seja concedido a possibilidade de instalar o Linux, já temos por onde começar a traçar nossos planos.

Bem o melhor é começar, distribuindo as aplicações de acordo com seu uso/necessidade nos setores. Isto não é muito difícil: Vendas e faturamento ficam aos cuidados do balcão e televendas. Controle de estoques aos cuidados do almojarifado e contas à pagar e receber com a administração. Se possível o melhor é distribuir mesmo em módulos separados, que serão disponibilizados diretamente nos equipamentos clientes.

O segundo passo é definir as políticas de acesso às aplicações, criando-se regras sobre quem “acessa o quê”: um usuário precisa ser identificado individualmente, ou pode-se utilizar um usuário por departamento? O normal é que se use identidades individuais. Criam-se ROLES ou perfis de usuários e atribuem-se estes perfis ao usuário Firebird.

Onde vai estar cadastrado este usuário? No banco de dados em uma tabela ou na tabela de usuários do Firebird (ISC4.gdb)? A resposta depende de alguns fatores:

⌘ Risco de ataque ao banco de dados por meio de ferramentas de terceiros

Se este risco for muito grande, o melhor é ficar com os usuários do Firebird, embora o controle possa ser eliminado se o “hacker” copiar sua base, ou sobrepor o arquivo de usuários (isc4.gdb).

⌘ Utilização de sistemas de terceiros no mesmo servidor com Firebird

Como já mencionado anteriormente, podem haver conflitos entre nomes de usuários e suas senhas, o que inviabilizará o uso de um dos sistemas. Neste caso o melhor mesmo é manter o cadastro de usuários no seu banco de dados, o que representa também uma possibilidade de num único backup ter todo o resgistro.

Sempre é bom ponderar bem sobre estes fatores, pois eles afetarão diretamente sobre o resultado final, isto é, são determinantes na construção do seu modelo de segurança. Como sempre fui favorável ao “caminho do meio”, penso que um modelo de segurança adequado não pode pesar apenas para um lado, sob pena de ficarmos vulneráveis de alguma forma. O jeito é então termos um misto entre as duas soluções.

Você pode por exemplo, criar duas ROLES e alguns usuários básicos no banco de dados, além do próprio SYSDBA. Uma ROLE seria para o usuário do banco de dados, aquele que vai trabalhar diariamente com ele. A outra seria para o administrador do sistema, que pode cadastrar novos usuários, mas não pode trabalhar com o banco de dados. Assim um usuário logado com uma ROLE “comum” não poderá cadastrar novos usuários ou ainda modificar dados em tabelas específicas, só utilizadas pelo administrador. Da mesma forma o administrador terá uma limitação não lhe sendo permitido trabalhar com as demais tabelas.

Por padrão não dê direito à nenhuma tabela à não ser ao SYSDBA. Assim você força a adoção de ROLES para o login. E restrinja as ROLES de acordo com os usuários, de forma que um usuário padrão não pode logar-se com a ROLE de administrador.

Paralelo à isto, crie uma tabela de usuários da aplicação no seu banco de dados

(que poderá ou não coincidir com os usuários Firebird), e associe à ela uma tabela de níveis de acesso, que serve para restringir o acesso às rotinas das aplicações (afinal, é melhor deixar o controle de acesso da aplicação dentro do banco de dados, pois esta informação não teria valor algum se não usarmos aquela aplicação específica).

Com isto você pode ter um módulo de gerenciamento de usuários à parte, que fará o login no banco de dados como administrador, e fará a manutenção dos usuários Firebird e das aplicações. A sincronia com os usuários Firebird será determinante para este módulo “reconhecer” a sua metodologia de segurança.

Nas aplicações sempre será feito o login com o usuário previamente cadastrado no banco de dados, com o usuário padrão e a ROLE indicada. Não confunda o usuário/senha da aplicação com o usuário/senha do Firebird. Poderão haver, se desejado, “n” usuários utilizando a mesma conta no Firebird, desde que se tenha um log de alterações e exclusões pelos usuários da aplicação, não há necessidade de um usuário equivalente do Firebird.

Sendo assim, se uma aplicação de terceiros estiver instalada, se ela utilizar o SYSDBA ou outro usuário, não haverá maiores problemas, pois você nunca irá utilizá-lo em suas aplicações, devendo se possível alterar a senha dele imediatamente. Se você ainda der uma “otimizada” no ISC4.gdb, aí sim teremos algo ainda mais seguro (vide artigo citado anteriormente).

Nem é preciso dizer que nada disto surtirá muito efeito, se o servidor estiver com os arquivos visíveis e disponíveis. Se não houver um Firewall, suas chances de problemas aumentam. Se as máquinas clientes acessarem internet, verifique a presença de um bom anti-vírus como o [AVG](#), que se atualize automaticamente. Se futuramente for disponibilizado um módulo de venda pela Internet, considere a possibilidade de utilizar o Zebedee.

Também será preciso ter uma rotina de backups, preferencialmente automática e com versões diferentes de modo que um banco corrompido não seja backupeado indefinidamente. Para isto considere dentre as ferramentas disponíveis qual a melhor. Recomendo o [GbakSheduler](#).

Deve haver preocupação também quanto à forma que os dados são recuperados e inseridos no banco de dados. Isto está relacionado com o tempo de vida e o nível de isolamento das aplicações. Não podemos chamar de seguro uma aplicação com longo tempo de vida das transações, ou que permita leitura indiscriminada de dados ainda não confirmados (dirty-read). Também não é por isto que devemos simular um travamento pessimista dos dados. “O caminho do meio”, certo? Transações precisam ocorrer no tempo mínimo, e com o nível de isolamento correto para cada caso. Mais detalhes no artigo [“Trabalhando com transações em IBO”](#), um enfoque do uso com o IBO, de Jason Wharton.

Procure adotar Triggers e Stored Procedures para reduzir o tráfego de rede, transferir código ao para processamento pelo Firebird, e com isto garantir segurança e performance. Sempre que possível vá armazenando os dados à medida que são inseridos, para evitar picos de tráfego e um “over head” do servidor. Isto deve ser implementado com técnicas adequadas de modelagem de dados, associadas aos recursos do Firebird.

Por último, é sempre bom treinar bem as pessoas envolvidas no manejo do sistema e nos responsáveis pela manutenção das questões de segurança. Um fato incontestável é que se você instalar sua aplicação num ambiente onde há desconhecimento, pouca cooperação ou até mesmo sabotadores, será difícil garantir o bom andamento das coisas. No entanto se você se precaver com medidas de segurança, poderá minimizar e eliminar alguns riscos. Por exemplo, o banco pode ficar comprometido por um usuário que fechou a aplicação abruptamente, não revelando isto a ninguém. Isto só poderá ser detectado com o uso de um log de conexões/desconexões. Algum dado que “desapareceu” novamente recorra ao log, de preferência alimentado por Triggers disparadas nas ações.

Conclusões

O Firebird é um excelente SGBD: Rápido, robusto e versátil. Mas ele pode ser ainda mais: pode ser seguro.

Não há sistema invulnerável, mas existe uma diferença clara entre algo bem pensado e algo suscetível à ataques.

Você terá de “malhar” um pouco em cima do Firebird e das suas aplicações, mas lembre-se que nenhum SGBD é totalmente seguro por si só. Precisa de um sistema operacional seguro, uma rede segura, e uma aplicação com um conjunto de regras que propicie segurança.

Por isto em dados momentos você terá uma escolha a fazer: colocar suas aplicações em ambientes pouco seguros ou “bater o pé” com o seu cliente. Este talvez seja o maior desafio: o convencimento dos usuários leigos nestas questões. Há quem defenda a tese de que se o usuário for advertido e ainda assim desconsiderar o aviso, irá “pagar caro” por isto, quando precisar que alguém lhe socorra. Algo como “pague agora ou pague depois”.

Se você já estiver adiantado nos seus projetos, considere o impacto de algumas técnicas aqui propostas, quanto ao seu custo x benefício. Se estiver iniciando, dedique tempo para considerar estes fatores, para que o resultado seja o esperado nos aspectos de segurança.

Espero que os caminhos aqui apontados possam servir para você fazer planejamento e adotar a sua estratégia de segurança. Se persistirem dúvidas, procure ajuda, estamos aqui para isto.

Bom trabalho!

Usando o Poledit (Policy Editor) para reforçar a segurança em computadores com Windows 95/98

Por padrão, o ambiente Windows 95/98 não é seguro. Os usuários podem acessá-lo sem o uso de senhas, podem "bagunçar" todas as configurações do sistema, e podem ter acesso a todos os programas e dados com facilidade.

A Microsoft oferece uma ferramenta para ajudar na limitação de acesso aos usuários - O editor de Diretivas do sistema, ou Poledit, como é mais conhecido. O Poledit não faz parte de nenhuma instalação do Windows 95/98, mas pode ser encontrado no CD da distribuição. No CD do Windows 95 procure na pasta D:\Admin\Apptools\Poledit (onde D: é a sua unidade de CD). No CD do Windows 98, procure na pasta D:\Tools\Reskit\Netadmin\Poledit.

Surpreendentemente a Microsoft tem a versão para Win95 do Poledit disponível para download no seu cavernoso site em:

<http://download.microsoft.com/download/win85upg/poledit/1/W95/EN-US/policy.exe>

Você pode utilizar a versão para Win95 no Win98, ou baixar a versão para Win98 que encontra-se em:

<http://www.microsoft.com/office/project/prk/utilitiesSetupPol.exe>

Após baixá-lo quando da execução do executável, escolha o local onde irá descompactá-lo. Para utilização no *Windows Me* baixe em:

<http://www.annoyances.org/downloads/ftp/poledit.zip> .

Antes de iniciar o uso do Poledit ou qualquer outra ferramenta de segurança, pense cuidadosamente naquilo que você quer proteger. Segurança sempre envolve restrição de acesso - manter os usuários longe de certas coisas, torna o acesso mais difícil para você também. E é fácil chavar sistemas tão fortemente que os usuários não poderão efetuar tarefas que talvez devessem poder fazer. É interessante considerar o quanto os usuários podem causar de danos se as configurações não estiverem travadas, e comparar com a inconveniência causada à você ou a outro usuário que tenha o direito de executar estas mudanças - uma espécie de análise custo-benefício. Não há uma resposta simples ou única sobre o quanto a segurança provê uma relação entre conveniência e proteção - há diferentes respostas para diferentes configurações.

Alterações feitas com o Poledit são globais - elas afetam qualquer um utilizando a máquina com aquele login - se você definir opções de segurança enquanto logado com um perfil "Estudante" por exemplo, a máquina irá restringir suas ações tanto quanto um "estudante" real. Particularmente eu não defino múltiplos perfis de usuários - eu restrinjo um determinado número de opções, que irei indicar logo adiante - e desde que haja um único perfil, qualquer um - estudantes, professores, e eu, somos igualmente restritos. Então, eu não desligo opções que eu preciso acessar regularmente - e eu entendo que se eu preciso acessar uma das definições restritas (por exemplo, mudar o papel de parede), então eu preciso rodar novamente o Poledit primeiro para garantir o acesso - e depois de feito, preciso lembrar de tornar o acesso restrito novamente.

Note que as versão Win95 e Win98 do Poledit não são idênticas - a versão 98 inclui muito mais modelos com funções específicas, a maioria específicas para o Internet Explorer e outras ferramentas de Internet. Em função disto a versão Win95 é mais fácil de usar - então a examinaremos primeiro. Até onde pude testar, a versão 95 funciona bem com o Windows 98, de modo que você pode escolher qual versão usar.

Você não poderá instalar o Poledit usando o instalador do Windows ou a opção de Adicionar/Remover Programas do Painel de Controle. Certamente isto se deve ao fato de que a Microsoft não deseja que usuários comuns façam este tipo de "bagunça". Você poderia copiar o conteúdo do Poledit para uma pasta no HD das máquinas, mas não faça isto. Se você copiar para um drive de rede compartilhado, não torne público ou óbvio o caminho. O melhor a fazer é manter uma cópia em um disquete e executá-lo a partir daí,

não deixando nenhuma cópia para acesso de um usuário.

Atenção! O uso do Poledit pode causar problemas - leia este documento cuidadosamente, e preste atenção no que você estiver fazendo. Experiências podem modificar uma configuração crítica e de difícil conserto.

Rode o Poledit, dando um clique duplo no arquivo Poledit.exe. A primeira vez que você executar a versão Win95 em cada computador, ele irá lhe pedir para abrir um arquivo *.ADM - e irá lhe mostrar os que estiverem disponíveis em sua pasta.

Escolha a única opção: Admin.adm. Então no menu Arquivo (File), escolha Abrir Registro (Open Registry). Dê um clique duplo sobre o ícone Usuário Local (Local User) (Há um grupo de opções para o Computador Local -mas elas na maioria afetam o logon em servidores NT - e algumas delas são itens que eu penso que você precisará mudar, então não vamos considerá-las nesta discussão).

Quando você abrir o ícone de Usuário Local, você terá uma janela com a lista de áreas que podem ser controladas, com um [+] ao lado de cada área. Clicando no [+] abre-se a área, mostrando suas opções relacionadas. Em cada conjunto, você encontrará um número de opções que você pode desativar...quando estiver terminado, clique em OK. Tão logo você salve suas modificações, utilizando o menu Arquivos/Salvar (File/Save), as mudanças terão efeito.

Nota importante: Se você estiver usando logons múltiplos de usuários, como "Professor/Estudante/Etc.", as mudanças feitas para o Usuário Local afetarão aquele usuário/logon... Se você fizer estas alterações enquanto logado na conta Professor, por exemplo, você precisará fazer novamente as mudanças para a conta Estudante. Vamos examinar a seguir as várias definições de usuários que podem ser controladas.

Painel de Controle - Vídeo (Control Panel - Display)

Quando você clica no [+] ao lado no Painel de Controle, uma lista de itens se abre com um [+] ao lado de cada uma... Se você clicar no primeiro [+], ao lado de Mostrar, você não terá opções visíveis até você clique no marcador para Restringir a visualização do painel de controle. Então você verá as seguintes opções: Painel de Controle, Área de Trabalho, Rede, Shell Sistema

Clicando e um ou mais dos marcadores da seção abaixo, você pode:

- ☒ *Desabilitar o Painel de Controle* - Desliga totalmente o acesso ao painel de controle, tanto a janela principal, como através do acesso pelo clique do botão direito do mouse na área de trabalho em "propriedades" no menu rápido. Se os usuários tentarem acessar este item verão uma mensagem: "O Administrador do Sistema desabilitou o acesso ao Painel de Controle".
- ☒ *Esconder página do Papel de Parede* – Permite o acesso ao Painel de controle, mas esconde esta página, impedindo que os usuários mudem a aparência da área de trabalho. Note que isto não impede o uso do recurso "Salvar como papel de parede" do browser.
- ☒ *Esconder página da Proteção de Tela* – Permite acesso ao painel de controle, mas a página do protetor de vídeo será ocultada.
- ☒ *Esconder página Aparência* – Impede o usuário de trocar as cores usadas nos elementos do Windows.

Painel de Controle – Senhas (Control Panel – Passwords)

Igualmente, este item permite restringir o acesso às senhas no painel de controle, removendo acesso às páginas mudar senhas, administração remota e Perfis de usuários.

Painel de Controle – Impressoras (Control Panel – Printers)

Remove o acesso ao painel de controle de impressoras. A primeira opção desliga o acesso às páginas "geral" e "detalhes" - que é usada para imprimir páginas de teste, trocar a

porta da impressoras (por exemplo mudar na rede). Particularmente eu não restrinjo este acesso, pois ele é muito útil na hora de resolver problemas de impressão.

Os dois itens seguintes restringem o poder de adicionar e remover impressoras. Normalmente eu restrinjo os usuários de remover impressoras, mas deixo eles adicionarem, o que torna fácil para mim criar uma nova impressora se preciso.

Painel de Controle – Sistema (Control Panel – System)

Eu restrinjo todos os itens desta lista, embora alguns argumentem que o acesso ao Gerenciador de Dispositivos (Device Manager) é vital para diagnosticar problemas de hardware. Entretanto, não há razões para os usuários terem acesso a num destes itens extremamente técnicos.

Nota: Não há como remover acesso total ao painel de controle com o Poledit. Mesmo com toda a proteção oferecida, os usuários poderão ainda acessar uma série de configurações. Nenhum dos itens a seguir deve ser desabilitado pois podem comprometer o funcionamento do Sistema Operacional!

Área de Trabalho (Desktop)

Este item mostra a definição do papel de parede e o esquema de cores (se houver). Pode ser usado para “resetar” o papel de parede, após alguém por exemplo, ter utilizado o browser para trocar o papel de parede. Esta mudança só é feita depois de reiniciar a máquina.

Rede – Compartilhamento (Network – Sharing)

Estes itens são diferentes do Painel de Controle – Rede, e permitem que você remova o acesso aos controles de compartilhamentos de arquivos e impressoras. Configure as máquinas de maneira correta e “chaveie” o compartilhamento.

Shell – Pastas Personalizadas

Eu não utilizo nenhuma destas restrições – que permitem que você modifique os locais padrão para os programas instalados (de C:\Arquivos de Programas), e outras definições. Você pode mudar isto para outros locais, talvez para um drive de rede compartilhado, por exemplo, mas eu não faço isto.

Shell – Restrições

Estes itens dão a você o poder de desligar um bom número de recursos da interface, e iremos examiná-los em mais detalhes:

- ✗ *Remover comando Executar* – Remove este item do menu Iniciar. Eu sempre o deixo, pois é mais fácil para acessar programas com linha de comando. Você pode querer remover esta opção.
- ✗ *Remover pasta Configurações do menu Iniciar* – Remove o acesso ao Meu Computador e opções do Explorer. Marque isto!
- ✗ *Remover Propriedades da Barra de Tarefas* – Remove o acesso às configurações da barra de tarefas e menu Iniciar. Eu recomendo que NÃO se use isto, pois a maioria dos computadores tem um menu Iniciar bem bagunçado, e é importante o usuário aprender a arrumar isto, mantendo-o limpo e enxuto.
- ✗ *Ocultar Unidades no Meu Computador* – Esconde todas as unidades locais e de rede. Isto torna impossível que o usuário clique em documentos salvos, entretanto eles podem encontrá-los através de Arquivos/Abrir nas aplicações. Eu recomendo que você não marque esta opção, talvez você possa usar o applet freeware TweakUI para o Painel de Controle para limitar o acesso às unidades.
- ✗ *Ocultar Ambiente de Rede* – Se você não usa uma rede, você pode marcar isto, para

remover este ícone da área de trabalho

- ✘ *Não mostrar a rede inteira no Ambiente de Rede* – Se você tem grupos de trabalhos que devam ser restritos, é interessante utilizar este recurso. Com isto, eles não conseguirão acesso à outros grupos de trabalhos.
- ✘ *Não mostrar o conteúdo dos grupos no Ambiente de Rede* – Isto irá limitar os usuários à unidades mapeadas e impressoras pré-definidas. Eu normalmente deixo o acesso à recursos compartilhados que não estejam mapeados, mas se você tem problemas com curiosos, você pode restringir o acesso desta forma.
- ✘ *Ocultar todos os ícones na área de trabalho* – Eu suponho que alguns gostem de esconder o Meu Computador, o Ambiente de rede, a Lixeira e qualquer outro ícone na área de trabalho, deixando apenas o menu iniciar e a barra de tarefas. Escolha o que você achar melhor. Tem gente que destesta uma tela lotada de ícones.
- ✘ *Desabilitar o comando Desligar* – Não posso imaginar o uso disto. O Windows PRECISA ser desligado e reiniciado de tempos em tempos (eu recomendo pelo menos uma vez por semana) para recuperar os recursos do sistema.
- ✘ *Não salvar mudanças ao Sair* – Eu gosto desta...Pegue a área de trabalho do jeito que você gosta, e aplique esta restrição. Desta forma se os usuários mudarem de lugar os ícones, quando o sistema for reiniciado, eles estarão de volta do jeito que você deixou. Isto não é útil se os usuários renomearem ou apagarem os ícones.

Sistema – Restrições (System – Restrictions)

Uma coleção de restrições de baixo nível:

- ✘ *Desabilitar ferramentas de edição do Registro* – Impede o uso do Regedit para fazer mudanças no registro...alguns tiveram problemas depois de usar isto, pois acabaram por “trancarem-se” no seu próprio sistema. O Regedit é poderoso e potencialmente perigoso, é bom pensar bem antes de habilitar esta restrição. Não me sinto confortável para recomendar o uso desta restrição.
- ✘ *Rodar somente aplicativos permitidos* – Se você quer realmente controlar o que os usuários tem acesso, isto é para você! Você adiciona as aplicações (uma de cada vez) permissíveis, e todas as outras não funcionarão.
- ✘ *Desabilitar o Prompt do MS-DOS* – Com isto ninguém vai conseguir explorar ou deletar arquivos pelo prompt do DOS! Se você permitir acesso, não reclame!
- ✘ *Desabilitar aplicações em modo somente MS-DOS* – Impede que aquelas aplicações que exijam boot somente DOS de rodarem.

Configurando um grupo de máquinas

A repetição das configurações em várias máquinas pode ser bastante tediosa. Em vez de fazer isto, defina as configurações em um computador e salve-as (aplicará ao Registro). Leve o arquivo ao outro computador e abra o arquivo .POL gerado. Faça as alterações específicas para a máquina e salve novamente.

Salvando sua pele

Sem querer abri um arquivo .POL do CD do Windows chamado Maximum.pol (eu queria abrir o Standard.pol) para dar uma olhada, e salvei. Com isto acabei proibindo o acesso ao registro do Windows, de modo que nem o Poledit pode ser usado para corrigir a mancada! O que fiz? Formatar? No Way! Pesquisando daqui e dali, descobri um jeito para resolver isto (graças a um bom livro, chamado “The Windows 98 Registry: A Survival Guide for Users” de John Woram). Crie e salve um arquivo texto como [c:\recover.reg](#) com o texto abaixo:

REGEDIT4

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer]

```
"RestrictRun"=dword:00000000  
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System]  
"DisableRegistryTools"=dword:00000000
```

Reinicie a máquina em modo MS-DOS (com F8, quando aparecer a mensagem "Iniciando o Windows" e no prompt do C: digite: *regedit recover.reg*)

Isto irá carregar o conteúdo deste arquivo no registro do sistema, desligando as duas linhas que restringiam o acesso ao Registro. Dáí reinicie o Windows, rode o Poledit e remova qualquer coisa indesejada. Thank you Mr. John Woram!

Diferenças entre o Poledit 95/98

O Poledit 98 além de diferença na ordem que as coisas são apresentadas, possui opções adicionais como controle sobre os menus Favoritos e Documentos, além de disponibilizar uma série de modelos de segurança pré-configurados.


Para saber mais

☞ Microsoft Knowledge Base Article- Q147381:

[How to Use System Policies On a Standalone Computer](#)

☞ [Create Secure User Profiles with Windows 9x Policy Editor](#)

☞ [System Policies vs Group Policies](#)

<p style="text-align: center;">Artigo Original</p> <p style="text-align: center;">Paulo Vaz (Colaborador da CFLP)</p> <p style="text-align: center;"><u>paulo@multi-informatica.com.br</u></p>	<p style="text-align: center;"></p> <p style="text-align: center;">Comunidade Firebird de Língua Portuguesa</p> <p style="text-align: center;">Visite a Comunidade em:</p> <p style="text-align: center;"><u>http://www.comunidade-firebird.org</u></p>
<p style="text-align: center;">A Comunidade Firebird de Língua Portuguesa foi autorizada pelo Autor do Original para divulgar este trabalho</p>	