Segurança em Bancos de Dados

Stupid Stuff I Wish Someone Had Told Me Four Years Ago

(Read the .tex file along with this or it won't make much sense)

1 Introdução

Tecnologias de banco de dados são o componente principal de muitos sistemas de computação. Elas permitem que a informação seja retida e compartilhada eletronicamente e a quantidade de informação armazenada nesses sistemas continua a crescer à uma taxa exponencial.[1]

Porém, danos e uso indevido de dados afetam não apenas um único usuário ou aplicação, mas pode ter consequências desastrosas para toda a organização. A proliferação rápida de aplicações baseadas na web e sistemas de informação aumentou ainda mais o risco de exposição de bancos de dados e, assim, a proteção de dados é hoje mais crucial do que nunca.[2]

Brechas de segurança são tipicamente categorizadas como observação não autorizada de dados, modificação incorreta de dados, e indisponibilidade de dados. Observação não autorizada de dados resulta em revelação de informação para usuários sem dinheiro de ganhar acesso a tal informação. Todas as organizações, variando de organizações comerciais para organizações sociais, numa variedade de domínios podem sofrer grandes perdas tanto do ponto de vista financeiro como humano como consequências da observação não autorizada de dados.[2]

Modificação incorreta de dados, tanto intencional como não intencional, resulta em um estado incorreto no banco de dados. Qualquer uso de dados incorretos pode resultar em grandes perdas para a organização. Quando a informação está indisponível, informação crucial para o funcionamento adequado da organização não está prontamente disponível quando preciso.[2]

Assim, uma solução completa para a segurança de dados deve atender os seguintes três requisitos:

1) Sigilo ou confidencialidade refere-se a proteção dos dados contra revelação não autorizada, 2) integridade refere-se a prevenção da modificação imprópria de dados não autorizada, e 3) disponibilidade refere-se a prevenção e recuperação do hardware e software de erros e de negações de acesso de dados maliciosas tornando o banco de dados indisponível.

Esses três requisitos surgem em praticamente todos os ambientes de aplicação, e para atender a estes requisitos são utilizadas diversas técnicas, que serão abordadas neste artigo.[2]

2 Um pouco de história

Os primeiros esforços em pesquisa na área de modelos de controle de acesso e confidencialidade para DBMSs (database management systems) focaram no desenvolvimento de duas diferentes classes de modelos, baseados na política de controle de acesso discricionário e na política de controle de acesso mandatório. Essa primeira pesquisa foi lançada na estrutura dos sistemas de banco de dados relacionais.[2]

O modelo de dados relacionado, sendo um modelo declarativo de alto-nível especificando a estrutura lógica dos dados, fez o desenvolvimento de linguagens declarativas simples para a especificação de políticas de controle de acesso possível. Esses primeiros modelos e os modelos discricionários em particular, introduziram alguns princípios importantes que separam os modelos de controle de acesso para sistemas de banco de dados de modelos de controle de acesso adotados por sistemas operacionais e sistemas de arquivos.[2]

O primeiro princípio era que modelos de controle de acesso para bancos de dados deveriam ser expressos em termos do modelo lógico de dados; assim autorizações para um banco de dados relacional deveria ser expresso em termos de relações, atributos de relações, e tuplas. O segundo princípio é que bancos de dados, em adição ao controle de acesso baseado em nomes, onde os objetos protegidos são especificados ao dá-los nomes, controle de acesso baseado em conteúdo deve ser suportado. Controle de acesso baseado em conteúdo permite que o sistema determine se deve dar ou negar acesso para um item de dados baseado no conteúdo desse item de dados. O desenvolvimento de controle de acesso baseado em conteúdo, que são, em geral, baseados na especificação de condições conta conteúdos de dados, tornou-se fácil em bancos de dados relacionais pela disponibilidade de linguagens de consulta declarativas, como SQL.[2]

Na área de modelos de controle de acesso discricionário para sistemas de bancos de dados relacionais, uma importante contribuição foi o desenvolvimento do modelo de controle de acesso System R, que influenciou fortemente os modelos de controle de acesso dos DBMSs comerciais modernos. Algumas características chave desse modelo incluem a noção de administração de autorização descentralizada, concedimento e revogação dinâmica de autorizações, e o uso de visualizações para suportar autorizações baseadas em conteúdo.[2]

Além disso, o formato inicial de comandos bem conhecidos para conceder e revogar autorizações, que hoje são parte do padrão SQL, foram desenvolvidos como parte desse modelo. Propostas de pesquisa posteriores estenderam esse modelo com uma variedade de características, como autorização negativa, autorização baseada em papel ou tarefa, autorização temporária e autorização ciente do contexto.[2]

Modelos de acesso discricionário têm, entretanto, uma fraqueza no sentido de que eles não impõem nenhum controle sobre como a informação é propagada e usada uma vez que tenha sido acessada por indivíduos para fazê-lo. Essa fraqueza faz os controles de acesso discricionários vulneráveis a ataques maliciosos, como Trojan Horses embutidos nas aplicações de programas. Um canal secreto é qualquer componente ou característica de um sistema que é usurpada

para codificar ou representar informação para transmissão não autorizada, sem violar o a política de controle de acesso estabelecida.[2]

A área de controle de acesso mandatório e sistemas de bancos de dados multinível tentaram abordar esses problemas mas sem muito sucesso. Porém, muitas soluções nas pesquisas para os problemas de canais secretos foram incorporadas nos banco de dados modernos, que contam com recursos como hierarquias de herança, agregação, métodos e stored procedures, tornando-os mais seguros.

3 Controle de acesso

4 Database Audit

As técnicas de Database Auditing tem por finalidade rastrear atividades de usuário, assim como tentativas de acesso (bem ou mal sucedidas) ao banco de dados, para assim, permitir a identificação de possíveis comprometimentos a segurança do mesmo.

Ao realizar esta prática é preciso atentar-se a certas informações chave que podem, ou não, indicar um comprometimento do seu banco. Dentre estas podemos destacar alterações indevidas na configuração do banco de dados, que podem ser um indício de que segurança foi violada. Vale lembrar que nem sempre tal alteração se deve realmente a intrusão de um indivíduo mal intencionado em seu sistema, no entanto certas mudanças podem gerar vulnerabilidade em seu banco, de modo que é preciso obter informações qualquer mudança de configuração ocorrida.

É preciso prestar especial atenção a execução de certos comandos no banco, como adição, exclusão e alteração de privilegio dos usuários (Data Control Languagem), ou mudanças estruturais em tabelas ou alteração de tipos de atributos (Data Definition Language).

Quanto a queries DML (Data Manipulation Language), é preciso notar principalmente alterações nos dados, assim como queries ad hoc.

Todas as queries acima mencionadas são executadas de maneira legitima em determinadas situações, portanto nenhuma caracteriza um comprometimento de segurança por si mesma. Assim, é preciso verificar se elas foram executadas dentro de um contexto apropriado ou não. Por exemplo caso um funcionário de uma empresa faça uma pesquisa no banco de dados por informações consideradas sensíveis, seu comportamento sera considerado normal caso ação seja condizente com suas responsabilidades na empresa, e anormal caso contrário.

Para realizar a auditoria, pode se utilizar tanto funcionalidades disponibilizadas pelos próprios sistemas gerenciadores de banco de dados quanto soluções disponibilizados por terceiros.

Usualmente, os sistemas de gerenciamento de banco de dados vem com estas funcionalidades desativas por default, sendo necessário a que sejam ativadas pelo DBA. Geralmente as funcionalidades nativas dos gerenciadores de banco de dados se resumem a manter um log de toda a atividade ocorrida no banco, que devem ser examinado periodicamente. Existem duas principais desvantagens de

utilizar os recursos nativos. A primeira é a diminuição de desempenho que o banco de dados terá devido ao custo computacional de armazenar localmente um log de todas as atividades ocorridas no banco. A segunda é a presença de um intervalo de tempo significativo entre a atividade suspeita ser registrada no log e o responsável pelo banco tomar conhecimento desta atividade.

Soluções disponibilizadas por terceiros podem ser de 3 tipos diferentes. Podem ser baseadas em sniffar pacotes com queries SQL destinadas ao servidor de banco de dados. Podem ser baseadas apenas em software. . Ou podem ser soluções mistas, que utilizam tanto um como outro.

Ao aplicar a primeira solução, sniffar a rede em busca das queries SQL enviadas ao servidor, temos alguns problemas. Uma delas é o custo alto, pois é necessário a aquisição e manutenção de hardware dedicado a tarefa de sniffar a rede. Outro problema é que, devido a conexão estar criptografada, nem sempre é possível aplicar esta solução. Fora isso ainda é preciso considerar que, nem todas as queries seram capturadas, já que qualquer uma que seja executada direto no servidor estará fora do alcance do sniffer.

Para se livrar do último problema e tentar garantir todas as queries sejam armazenadas, é possível incluir também um software no servidor para armazenar as queries feitas diretamente nele. No entanto, assim como as funcionalidades nativas do gerenciador, este software causaria uma perda de desempenho no servidor de banco de dados.

A verdadeira vantagem de se ter um soluções terceirizadas ao invés de contar com as funcionalidades nativas do gerenciador de banco de dados, reside no fato de que tais soluções contam com características que as outras não possuem. Dentre estas, podemos destacar a criação de políticas, a detecção de violação de política em tempo real, mensagens de alerta e a separação de funções.

Basicamente, criação de políticas, detecção de violação de política em tempo real e as mensagens de alerta, significa que é possível criar um conjunto de regras que definam quais comportamentos são considerados normais dentro do sistema, e quais não são, que o sistema conseguira detectar em tempo real quando uma destas regras foi quebrada e que o responsável pelo banco de dados sera notificado na hora.

Separação de funções significa que, ao contrário do que acontece com o com as funções nativas do gerenciador de banco de dados, quando se lida com soluções de terceiros a responsabilidade pelo monitoramento e pela administração do banco de dados estão bem separados, de modo que é possível realizar o database auditing com a menor interferência possível do DBA.

5 Referências bibliográficas

Arthur Monteforte: Introdução, Um pouco de história.

Paulo Gabriel Massa: Database Audit.

[1] - “Database Security: What Students Need to Know”, Journal of Information Technology Education, 2010.

[2] - “Database Security: Concepts, Approaches, and Challenges”, IEEE Transactions on Dependable and Secure Computing, 2005.

[3] - ”DATABASE AUDITING TOOLS AND STRATEGIES”, Authored by: Ed Chopskie, Vice President SenSage.