Usando o Encadeamento de Transações para Implementar um Controle Fim-a-Fim de Fraudes em Aplicações Distribuídas

Rostand Edson Oliveira Costa — rostand@nti.ufpb.br
Francisco Vilar Brasileiro — fubica@dsc.ufpb.br

Universidade Federal da Paraíba
Centro de Ciências e Tecnologia
Coordenação de Pós-graduação em Informática
Av. Aprígio Veloso, s/n, Bodocongó
58.109-970, Campina Grande, Paraíba

1. Introdução

O objetivo deste trabalho é apresentar um mecanismo, chamado de encadeamento de transações, que atua na detecção de quando senhas e outras informações sigilosas vazaram e estão sendo usadas de forma fraudulenta. Esse mecanismo foi utilizado na implementação de uma solução para o problema de detectar fraudes na autenticação de usuários remotos em um ISP *(Internet Service Provider)*.

2. Implementando Segurança Fim-a-Fim através do Encadeamento de Transações

Atuando na camada de aplicação, o encadeamento de transações permite que seja criado um controle de segurança fim-a-fim entre os componentes de aplicações distribuídas, independentemente do uso ou não de outras técnicas de controle. Ele possui também características ligadas à prevenção de fraudes, pois permite uma autenticação mais eficiente das partes envolvidas em uma transação, do que aquela baseada apenas em senhas estáticas. Isto se dá porque o encadeamento de transações propicia uma associação da senha com alguns elementos dinâmicos e de validade temporária. Além disso, o encadeamento de transações também auxilia no controle de erros humanos e falhas de *software* ou *hardware* por possibilitar uma detecção posterior de quando a camada de dados persistente sofreu alterações indesejáveis de origem interna ou externa, com intuitos criminosos ou não.

É possível criar um contexto entre as partes cliente e servidor de aplicações distribuídas dividindo entre cada uma das partes o controle e detecção de situações de anormalidade. Com a manutenção por parte da aplicação cliente de uma espécie de resumo, com um mínimo de informações sobre as transações que trocou com a aplicação servidor desde um dado momento, e a checagem por parte do servidor se tal informação mantida pelo cliente confere com os dados mantidos na sua base persistente, é possível criar um encadeamento entre as transações efetuadas por ambos.

Encadeando uma transação com a sua antecessora, criamos então um contexto em que uma transação só será válida se referenciar como anterior a última transação registrada pelo servidor. Ou seja, mesmo uma transação legítima só será considerada como tal se chegar na seqüência em que foi gerada e sendo esperada como a próxima pelo servidor. Assim, a autenticação entre as partes passa a ser realizada de forma permanente, desde o momento em que a transação é composta pelo módulo cliente até a sua efetivação pelo módulo servidor, em uma espécie de desafio mútuo.

O encadeamento de transações aqui proposto é, antes de tudo, um algoritmo e não um produto. Como tal, ele pode ser facilmente adaptado para os recursos disponíveis ou já utilizados na aplicação. Entretanto, alguns elementos são fundamentais e devem, de alguma forma, estar presentes; são eles: um algoritmo de *secure hashing*; a inclusão de dois novos campos na camada persistente do servidor (hash da transação anterior e hash da transação atual); a manutenção de dados persistentes no cliente (hash da última transação); a criação de

validação adicional no servidor; e a criação de alguns processos "*batch*" para detecção de exceções.

Quatro tipos de validação são possíveis com o uso do encadeamento de transações. A **Validação Horizontal Imediata** que detecta clientes inválidos ou erros de transmissão e que verifica se o ***Hash Informado = Hashing(Dados Transação + HashAnterior Informado)***. A **Validação Vertical Imediata** que previne ou detecta fraudes ou exceções verificando se o ***Hash Anterior Armazenado = Hash Anterior Informado***. Existem ainda a **Validação Horizontal Posterior** e a **Validação Vertical Posterior** que são aplicadas em todas as transações de um determinado contexto verificando, respectivamente, se ***Hash Atual = Hashing(Dados Transação + Hash Anterior)*** e se ***Hash Anterior [i] = Hash Atual [i-1]***. Ambas servem para detectar exceções posteriores causadas por falhas de *hardware* e *software*, erros humanos ou fraudes diretas na base de dados.

3. Segurança Fim-a-Fim na Autenticação de Usuários Remotos

O uso do encadeamento de transações para a autenticação de usuários de acesso remoto pode minimizar o problema de fraudes no uso indevido de contas em um ISP, pois o candidato à nova conexão precisa provar que realizou também a anterior e assim por diante, em uma validação adicional. O principal fator para o incremento da segurança é que ao invés de contarmos apenas com uma autenticação baseada em identificação/senha, passaremos a contar também com uma espécie de *one-time password* lógico, solicitado como desafio pelo provedor. Este valor, que é o *hash* do encadeamento das conexões anteriores, mudará constantemente e será válido apenas para uma determinada conexão. Após a sua utilização, o valor representativo do contexto do usuário com o provedor perderá imediatamente a sua validade. Desta forma, se um *hacker* conseguisse obter a identificação e a senha de um usuário de um provedor, não conseguiria se conectar, a menos, que conseguisse obter também o valor atual do contexto. Note que para obter isto o intruso precisa invadir dois sistemas, o servidor do provedor e o computador do usuário.

Supondo que o *hacker* conseguisse também o valor atual do contexto e realizasse uma ou mais conexões se fazendo passar por um determinado usuário, ainda assim teríamos vantagens por usar o encadeamento de transações. Na primeira vez que o usuário legítimo tentasse realizar uma conexão, a mesma seria rejeitada, pois o contexto armazenado pelo usuário legítimo já estaria desatualizado. Esta exceção poderia desencadear diversas ações como alertas para o suporte ou, até mesmo, o bloqueio da conta do usuário para averiguação. Também seria possível identificar com precisão que conexões foram realizadas por um *hacker*, bastando apenas identificar o valor de contexto mantido pelo usuário legítimo, que indicaria a última conexão válida. Todas as posteriores a esta teriam sido feitas pelo *hacker*. Temos então proteção para o provedor e também para os usuários legítimos, com a minimização das perdas até mesmo em caso extremos de vazamento tanto das senhas quanto das informações sobre os contextos de conexão de cada usuário. É importante frisar que o uso do encadeamento de transações não substitui a autenticação já utilizada pelo provedor, mas soma-se a ela, complementado-a em busca de maior segurança.

4. Conclusão

Acreditamos que o mecanismo de encadeamento de transações para controle fim-a-fim aqui apresentado, por sua simplicidade, pode ser implementado com facilidade em muitos casos, quer seja como forma adicional de controle em situações que tenham a segurança como fator crítico, apoiando os outros mecanismos de proteção já implementados, quer seja como única forma de validação em aplicações com menores requisitos de segurança. Esse sentimento é substanciado pela nossa experiência na implementação da aplicação cliente/servidor para detecção de fraudes na autenticação de usuários remotos descrita nesse trabalho.