Segurança em Software de Código Aberto

- Quem acha que software de código fechado é mais seguro? - perguntou o palestrante.

Uns poucos gatos pingados levantaram a mão.

- E quem acha que software de código aberto é mais seguro?

Outros poucos ouvintes acreditavam que essa última era a afirmação verdadeira.

Mas a esmagadora maioria ficou passiva, sem saber o que responder.

Senhoras e senhores, estamos diante de um dos maiores dilemas da TI pós-Internet. É assunto para horas de discussão na mesa do bar, que tem grande chance de dar em nada, se for tratado de forma religiosa. Mas vamos tentar ser frios, relembrar algumas histórias e não nos perder em crenças infundadas.

Tudo gira em torno de segurança por ser (código) público versus segurança por ser fechado. A comunidade de programadores de código aberto tem a cultura do mérito e respeito, então é natural que seus membros tomem mais cuidado ao programar. Além disso, é comum o trabalho de um ser revisado e auditado pelo outro. Outra vantagem é a velocidade com que correções são escritas. Foi o caso de um problema descoberto na implementação TCP/IP de qualquer sistemas operacional, em 1996. A correção para Linux foi publicada em 20 minutos, enquanto que para outros sistemas demorou 2 dias úteis. Há dezenas de casos semelhantes.

Mas aqui vale um ponto de atenção: o sistema operacional Linux é um caso muito especial de software de código aberto, simplesmente porque ele é muito usado e tem um ecossistema enorme de programadores e empresas interessadas em sua estabilidade e progresso. Em outras palavras, ele tem uma infinidade de observadores, e isso não é verdade para qualquer software livre. Ou seja, só quando um software aberto usufrui de muitos usuários e desenvolvedores é que terá pessoas cavando e corrigindo problemas rapidamente em seus fontes.

O modelo de código fonte aberto de desenvolvimento de software não é uma garantia de segurança. Contudo, softwares livres populares como o Linux, Apache, Samba e muitos outros tem tido seus códigos examinados por vários especialistas de segurança.

Por outro lado, software de código fechado tem a garantia de que ninguém pode vasculhar falhas em suas entranhas. Ironicamente também garante que se o fabricante decidir implanatar um backdoor, ninguém poderá encontra-lo. E foi exatamente o que aconteceu quando a Borland liberou o código do Interbase: eles esqueceram de remover um trecho do código que abria um backdoor de administração. Foi descoberto e removido assim que outros começaram a olhar seu código fonte, e motivo de vergonha para a Borland.

Outro aspecto é que certas coisas são muito difíceis de desenvolver corretamente quando somente poucas pessoas tem acesso ao código fonte. É o caso de boa criptografia e de protocolos de comunicação seguros. Só uma densa auditoria multicultural e independente pode analisar a fundo cada detalhe do código.

No âmbito de ferramentas de segurança, o mundo livre dispõe de uma lista sem fim de coisas como OpenSSL, OpenSSH, PAM, PKI, OpenLDAP, Tripwire, Kerberos, SELinux, etc, todos possuidores de um forte ecossistema de usuários e desenvolvedores.

O pensamento saudável para essa questão é que software aberto e fechado tem vantagens e desvantagens que muitas vezes se completam. Nenhum modelo é garantia de segurança, mas ter o código aberto dá pelo menos a chance de um certo software poder ser auditado. Além de que uma falha detectada pode ser corrigida por qualquer pessoa a qualquer hora, e não ser tratada como “característica do software” que o fabricante não acha que deve corrigir.

A fórmula do sucesso, para balancear custos e benefícios, tende a usar software livre nos elementos mais infraestruturais do data center, enquanto que software fechado vai melhor nas camadas relacionadas a lógica de negócio, sempre abusando do uso de padrões abertos para garantir a interoperabilidade. Um exemplo prático desse bom balanceamento é rodar seu ERP corporativo (de código fonte fechado) sobre um sistema operacional de código fonte aberto, mas que tenha suporte comercial no mercado, como Linux.

(652 palavras)

Mini curriculum:

Avi Alkalay foi, por alguns anos, responsável pela segurança corporativa da IBM Brasil, e já trabalhou praticamente com todas as tecnologias da web. Hoje é arquiteto de soluções e consultor de Linux, Padrões Abertos e Software Livre na IBM.