

Tornando-se um “Artista de Segurança Linux”

Autor: Jon “Maddog” Hall

Tradução: Sandro Venezuela

“Quando o custo para obtenção da informação excede o valor obtido pela sua posse, a solução é eficaz.” - Guia Prático do Red Hat Linux por Mark G. Sobell, Terceira Edition (Prentice Hall), página 989.

Depois de quarenta anos no ramo de computação, a ideia que tem sido difundida por profissionais de segurança é de que não existe tal coisa de um sistema de computador seguro, mas apenas níveis de insegurança. Portanto o custo de manter a informação e o sistema de segurança tem de ser equilibrado com o custo de perder a informação ou sistema, ou danificá-lo. Infelizmente, a velocidade e disponibilidade da Internet combinada com o baixo custo de computadores e serviços de rede muito potentes, proporcionaram um custo de “invasão” cada vez mais baixo e um custo de “segurança” cada vez mais alto.

A coisa mais importante para um sistema seguro é ter uma boa política de segurança. Sem isso, você está perdido e vagueia de forma ineficaz. Portanto, você tem que fazer uma reflexão a respeito de quem será capaz de fazer o que, se essas limitações são obrigatórias e sem restrições, ou como vai implementar e aplicar essas políticas. Um bom exemplo de como não ter uma boa política é a empresa que força todos os seus funcionários a terem senhas longas e complicadas, que mudam uma vez por semana, mas tolera estas pessoas escrevendo suas senhas em etiquetas adesivas e coladas em seus monitores LCD, “porque estas pessoas podem não se lembrar das senhas”.

A próxima coisa mais importante é um bom conjunto de ferramentas de segurança e pessoas treinadas para implantá-las e acompanhar os resultados gerados.

Muitos sistemas desktops se escondem atrás de um “firewall” em ambientes corporativos ou mesmo domésticos. O firewall é um sistema especializado em aceitar dados da Internet e encaminhar estes dados para os desktops ou servidores. A esperança é que o firewall isole as pessoas más das pessoas atrás do firewall e, portanto, os sistemas podem ser mais “relaxados” em sua segurança. Infelizmente nos dias atuais de computação móvel, laptops podem se mover de dentro do perímetro protegido pelo firewall para o desprotegido “ambiente selvagem” da Starbucks, por exemplo, onde as pessoas bebendo café e “surfando na net” têm os seus notebooks infectados com vírus e cavalos de tróia e que trazem de volta estes notebooks ao escritório. Atualmente, os ataques as vezes vêm de dentro da organização (em que o firewall não dá nenhuma proteção) e não do exterior.

Outros sistemas não podem se esconder trás de firewalls e são chamados de sistemas “Bastião”. Eles são os sistemas que executam o seu servidor Web, Correio Eletrônico (E-Mail) e outro serviços. Estes são os sistemas que têm de ser “absolutamente (tanto quanto possível) protegidos”.

Finalmente, o acompanhamento constante das listas de segurança, sites e rápida aplicação de patches é fundamental para a segurança de um sistema. Ter o código fonte do seu sistema disponível significa que você não tem que esperar para que seu fornecedor disponibilize a correção compilada e testada. Você pode tomar a decisão de aplicar a correção, dependendo da criticidade do ataque.

Dadas as filosofias e as questões acima, eu acredito que o Software Livre e de Código Aberto é a

melhor base para permitir que seus sistemas inseguros alcancem a segurança, e este blog trata disso.

Este blog não é uma explicação detalhada de segurança de rede, nem sobre como bloquear SPAM, nem em ser um guia de receitas de curso de segurança do sistema. A segurança é uma forma de arte, bem como uma ciência, e este blog não pode fazer de você um Michelangelo em 3000 palavras. Se eu puder mostrar aqui que o seu sistema está atualmente no nível “pintura a dedo” e que, com o Software Livre você pode fazer uma “aquarela”, “pinturas a óleo” e não só isto, então eu acho que fiz um bom trabalho.

Algum dia seu trabalho pode estar em um "Museu da Arte de Segurança".

História e Arquitetura do Unix

Em 1969, Ken Thompson e Dennis Ritchie começaram a desenvolver o sistema operacional Unix, “apenas por diversão”. Querendo ou não, o Unix foi concebido para ser um sistema compartilhado, em primeiro lugar, e tornou-se rapidamente um sistema que permitia o compartilhamento por várias pessoas, com vários processos para cada pessoa. Isto imediatamente definiu um design mais robusto e seguro do que um sistema de usuário único, já que o conceito de estabilidade e de segurança teve que ser construído dentro do próprio sistema.

Concedido, nos primeiros anos da Bell Laboratories, não se teve muita atenção para a senha ou a segurança em nível pessoal, mas ao longo dos anos as coisas como: tempo de vida da senha, fortificação da senha e senhas “escondidas” foram colocadas no sistema para melhorar a segurança.

Unix foi criticado por seu modelo inicial de “superusuário” versus “todo mundo” na execução de programas (especialmente programas administrativos) e no agrupamento das propriedades do “dono”, “grupo” e “outros” (ou seja, todos os outros do mundo) com as capacidades de “leitura”, “escrita” e/ou “execução” no arquivo. Enquanto isso era relativamente simples, mas também uma estrutura de permissão elegante, funcionou bem durante alguns anos, com o tempo as “Access Control Lists” ou “ACLs” foram habilitadas, permitindo às pessoas criarem classes de privilégios de execução e acesso a arquivos e diretórios em um nível mais refinado.

Quando o Unix deixou a Bell Labs e entrou para o meio acadêmico, nas universidades, ele passou pela clássica “prova de fogo”, com os estudantes tentando invadir o sistema e os desenvolvedores tentando mantê-los fora. O Unix se tornou o sistema operacional para o estudo sério de ciência da computação e, portanto, (em muitos casos) para estudos sérios de segurança do computador.

Arquitetura do Linux

Como eu mencionei antes, a arquitetura do Linux segue de perto a arquitetura dos sistemas Unix. Um kernel monolítico relativamente pequeno com bibliotecas e utilitários que acrescentam funcionalidade a ele.

Isso por si só agrega valor a segurança, pois permite que o usuário final desligue uma série de serviços (tanto de máquina quanto de rede) que não precisa, que funcionando no sistema cria mais possibilidades de ataque.

Por exemplo, a grande maioria dos sistemas desktops atuam como cliente para os serviços, não

como um servidor. A desativação destes serviços significa que outras pessoas através da rede não podem se conectar a eles. Nos primórdios do Linux diversas distribuições eram disponibilizadas com os serviços ativados no momento da instalação. Esta foi uma impressão errada de que ter os serviços funcionando seria mais fácil administrar, mas os especialistas em segurança rapidamente apontaram que ter os serviços em execução no momento da instalação (antes da aplicação dos patches necessários), também deixaria os sistemas, mesmo que por pouco tempo, abertos a ataque. Agora a maioria, senão todas, as distribuições deixam estes serviços desligados e você é instruído a habilitá-los na hora certa, espero que somente depois de ter aplicado os patches necessários.

Outro exemplo é o conceito de remover os compiladores e outras ferramentas de desenvolvimento de software do sistema, uma vez que estas ferramentas dão aos crackers maiores possibilidades para explorar o sistema. Remover estas ferramentas significa que o cracker terá que utilizar outros métodos para “quebrar” a segurança.

Vários pacotes FOSS (Free and Open Source Software) foram adicionado a esta funcionalidade básica ao longo dos anos, dando ao Linux uma segurança ainda maior.

O primeiro é “PAM” ou “Pluggable Authentication Modules”. Em qualquer sistema, “autenticação” significa que você se identificou de tal forma que o sistema lhe dará acesso aos serviços. Assim que você entra com seu nome de usuário e sua senha, você está sendo “autenticado” tipicamente pelo nome de usuário e senha no arquivo `/etc/passwd` e pelo programa `login`. Da mesma forma o `ftpd`, e outros programas de “serviço”, irá autenticá-lo da mesma maneira.

Se você estiver em uma rede, no entanto, poderá ser autenticado por inúmeros métodos, que podem ser LDAP, DCE, Kerberos ou mesmo os métodos mais recentes, e inúmeros programas podem ter que ser alterados para refletir o novo método de autenticação. O PAM foi fornecido para permitir que novos métodos de autenticação possam ser aplicados a todos os programas do sistema que precisa de autenticação sem ter que mudar e integrar cada novo método de autenticação.

Outro método de autenticação mencionado anteriormente foi “Access Control List”, ou “ACL”. Uma ACL concede “acesso” para um arquivo ou diretório com base em uma extensão das permissões tradicionais do Unix “dono/grupo/outros” e “rwx” mencionada acima. Desde que as ACLs estejam implementadas como parte da estrutura do sistema de arquivos, você tem que ter certeza que seu kernel tenha sido construído para suportá-las, que o sistema de arquivos que você está usando oferece suporte a elas, e que o sistema de arquivos foi montado com as ACLs ligadas. Entretanto, uma vez realizado tudo isto, você pode atribuir permissões a vários usuários em uma base de usuários individuais, vários grupos em uma base de grupo-por-grupo, e assim por diante.

Isso permitirá você facilmente criar um grupo de operadores que podem iniciar ou parar um banco de dados ou fazer backups, mas não conseguir desligar todo o sistema, por exemplo.

Finalmente, você tem que estar ciente de que nem todos os utilitários do Linux suportam ACLs. Se você está copiando arquivos de um diretório para outro com o comando `cp` você deve usar as opções “-p” (preserve) ou “-a” (archive) no comando. Alguns dos comandos robusto do Unix como, `cpio`, `tar` e outros, não suportam a cópia das ACLs, e portanto as ACLs seriam perdida.

Sistemas de Arquivos Criptografados

Criptografar seus dados deve ser parte de sua política de segurança em um mundo de dispositivos USB, unidades portáteis e laptops roubados, e o Linux permite que você criptografe arquivos individuais, sistemas de arquivos, partições swap e mesmo sistemas de arquivos dentro de arquivos individuais.

Alguns desses métodos de criptografia também trabalham com sistemas de arquivos em nível de usuário, o que significa que você pode configurá-los enquanto o sistema está funcionando.

Loop-AES usa uma técnica de loop-back para permitir que o dispositivo de bloco faça a criptografia sem ter que mudar nada no kernel. Técnicas de loop-back também são úteis para sistemas de arquivos mantidos em um único arquivo, assim este método pode ser usado para criar um sistema de arquivos criptografado que está contido em um único arquivo em sua máquina.

DM-Crypt usa a funcionalidade de mapeamento de dispositivos (também útil para o RAID via software, snapshotting e outros recursos) do kernel para criptografar arquivos.

Cryptofs é um sistema de arquivos em espaço do usuário (FUSE) que permite montar um sistema de arquivos em um diretório, e em seguida todos os arquivos armazenados nesse diretório são criptografados, incluindo o nome do arquivo. Quando você desmontar o sistema de arquivos, os arquivos são criptografados e não serão de-criptografados até o sistema de arquivos ser montado novamente usando a mesma chave.

Existem outros métodos para criptografar arquivos e sistemas de arquivos, tais como *EncFS* e *TrueCrypt*.

Além disso, recentemente, um administrador de sistema Microsoft Windows inicializou um Live CD Linux em uma de suas máquinas e ficou surpreso ao verificar que o Linux pode ler e escrever no sistema de arquivos do Microsoft Windows, apesar de ter definido os diretórios como privado sob o sistema operacional da Microsoft. Expliquei-lhe que era um sistema operacional diferente e a menos que ele criptografasse todos os dados em seu disco, ele devia esperar que alguém usando um sistema operacional diferente em sua máquina fosse capaz de ver, alterar e excluir dados no seu sistema de arquivos do Microsoft Windows.

SELinux

A maioria dos métodos de autenticação de controle de acesso são arbitrários. O dono do objeto (seja um programa ou dado) pode alterar as permissões para outras pessoas e grupos.

Anos atrás, a Agência de Segurança Nacional (NSA) criou um projeto para aplicar “Mandatory Access Control” (MAC) dentro do kernel Linux. Este projeto ficou conhecido como “Security Enhanced Linux” ou “SELinux”. O MAC reforça as políticas de segurança que limitam o que um usuário ou programa pode fazer, e quais arquivos, portas, aparelhos e diretórios um programa ou usuário pode acessar.

SELinux tem três modos: “Desabilitado”, “Permissivo” e “Execução”. No modo “Desabilitado” nada é feito. Neste modo você tem as políticas configuradas e prontas, mas não ativas. O modo “Permissivo” registra as violações da política em arquivos de log para que você possa verificar ou

monitorar. No modo “Execução” qualquer violação da política de segurança será contida.

SELinux utiliza cerca de 5 a 10% do desempenho do sistema quando no modo de Execução ou Permissivo.

Da mesma forma, o SELinux pode ser executado em uma política de “Orientada” ou “Estrita”. A política “Orientada” significa que os controles MAC apenas se aplicam a determinados processos. A “Estrita” significa que os controles MAC se aplicam a todos os processos.

As pessoas devem ser advertidas de que o uso indiscriminado da política “Estrita” do SELinux pode tornar o sistema praticamente inutilizável para alguns usuários. Tem que haver um compromisso de manter o sistema seguro, mas permitindo que os usuários façam o seu trabalho.

Argumenta-se que o SELinux é um “exagero” em um sistema de um único usuário, mas com modernos exploits e o poder do “sistemas de um único usuário”, podemos encontrar mais e mais aplicações do SELinux em um desktop de um único usuário.

AppArmor

AppArmor é um outro sistema para “Mandatory Access Control”, mas que se baseia mais em uma base de programa-por-programa do que o SELinux e permite que você misture e aplique políticas do tipo “Execução” e “Permissivo” no sistema ao mesmo tempo.

Através do “perfil” de cada programa, o AppArmor pode limitar o que um programa pode fazer e quais arquivos ele pode acessar, gravar ou executar.

Algumas pessoas acham que o AppArmor é mais fácil de configurar e controlar do que o SELinux.

Tornando Arquivos "Imutáveis"

Se alguém invade seu sistema, ele pode alterar vários arquivos de controle, como o arquivo `passwd`. Você pode impedir isto tornando o arquivo “imutável”. Quando um arquivo é “imutável” ele não pode ser alterado, seja por escrita, exclusão, renomeado ou hard links, até mesmo pelo superusuário. Primeiro o arquivo tem que voltar a ter permissões “normais”, e então ele pode ser alterado. O comando usado para fazer um arquivo tanto imutável quanto voltá-lo ao normal é `chattr`, e tem a sintaxe desta forma: `chattr +i <nome_de_arquivo>`.

Usando o comando `chattr` com o parâmetro “a” em vez do “i” faz com que o arquivo só possa ter informações adicionadas. Isso é útil para arquivos de log, onde você deseja que o sistema adicione novas informações, não apagando as informações antigas.

Depois que o comando `chattr` for executado em um arquivo, mesmo o usuário root não pode alterar ou apagar o arquivo até que o arquivo seja alterado com as permissões anteriores, com “-i” ou “-a”.

Novamente, você tem que verificar se o sistema de arquivos que você está usando suporta esta funcionalidade. Os sistemas de arquivos Ext2 e Ext3 suportam.

Logs

Sistemas Unix e Linux têm arquivos de log. Esses arquivos registram diferentes tipos de eventos, desde inicialização e finalização de um processo até mensagens explícitas sobre o seu servidor de e-mail ou sua base de dados. A maioria dos sistemas Unix e Linux tem a possibilidade de encaminhar os vários níveis de informações desde “bom saber” para “crítica” para um repositório central. Os administradores de sistemas podem criar filtros e scripts para ajudá-los a monitorar esses arquivos de log, identificando atividades que indiquem pessoas acessando o sistema.

Estes arquivos, claro, devem ser protegidos utilizando o comando `chattr`, mencionado acima, com a opção “+a”.

Sistemas de Detecção de Intrusos

Existem vários Sistemas de Detecção de Intrusos (IDC) disponíveis para Linux. SNORT (<http://www.snort.org/>) é um deles. O SNORT utiliza um conjunto de regras para determinar as intrusões.

Backups

Apesar de todo seu trabalho, tempo, suor e lágrimas, eventualmente, seu sistema será comprometido. Agora você tem que descobrir quando ele foi comprometido, como foi comprometido e estar pronto para recuperar o que foi danificado sem permitir que outros possíveis vírus e trojans permaneçam no seu sistema.

Com muito trabalho, você pode ser capaz de usar ferramentas para varrer seu sistema à procura desses vírus e trojans. Ou você pode re-instalar a partir de um CD-ROM original, ou imagem ISO em boas condições, e mais todos os patches associados.

A forma final é ter um bom nível de backup de todo o sistema de produção que você tem e atualizar periodicamente o backup para ter certeza de ter todos os patches de segurança que ocorreram desde a última atualização. Se você puder determinar com precisão quando seu sistema estava comprometido, você poderá restaurar o sistema a partir de um desses backups. Caso contrário, você terá que instalar a partir do código-fonte.

Resumo

Estou certo de que muitos dos profissionais de segurança vão olhar para este texto e dizer: “Realmente fundamental”.

Outras pessoas podem olhar para algumas destas características e dizer: “Como posso manter-me atualizado de todas essas políticas e comandos em um sistema tão complexo como o Unix ou Linux?”. A resposta é que provavelmente você não conseguirá se manter atualizado de todas estas considerações sobre o sistema e é aí que as políticas de segurança entram em jogo. Faça com que cada sistema fique tão seguro quanto os serviços e informações armazenadas nele, permitindo que você ainda realize seu trabalho.

Além de estudar os recursos listados abaixo, você também deve olhar o site da sua distribuição específica. Porque há muitas maneiras de fazer a criptografia de arquivos, compilar um kernel, e garantir a segurança de um sistema, e sua distribuição pode ter desenvolvido uma arquitetura de segurança geral que complementa suas políticas e fazem a segurança do sistema de uma maneira muito mais fácil.

Recursos

Existem diversos livros bons sobre segurança de computadores e segurança em Linux, especificamente. Os dois abaixo são muito bons:

- “Hardening Linux”, by James Turnbull (Apress, 2005)
- “Linux Server Security”, by Michael D. Bauer (O'Reilly, 2005)

Além destes, existem os sites:

- <http://www.linux-sec.net/>
- <http://www.linuxsecurity.com/>
- <http://www.bastille-unix.org/>
- <http://tldp.org>