A segurança com criatividade:
Os mínimos detalhes são os maiores amigos
Parte I.

Por Adriano Carvalho (ch0wn).
adriano@netnix.com.br
adriano.carvalho@niteroi.rj.gov.br

Introdução:

Bom, neste pequeno artigo vou tentar explanar algumas questões muito simples, porém de grande utilidade. Acredito que o título tenha sido muito direto.

Em muitos lugares, diversos administradores têm nas mãos ferramentas das mais complexas e específicas, mas esqueçam da base do sistema, dos comandos simples que fazem a diferença, e principalmente da criatividade. Sim, quem trabalha com segurança, TEM QUE TER obrigatoriamente criatividade, do contrário, de que adiantaria o conhecimento se não souber onde aplicar ? Pois bem, estamos em um ponto de equilíbrio, onde conhecimento e "capacidade de imaginação" se cruzam.

Para se desenvolver um bom trabalho, é necessária uma preparação, um planejamento de tudo o que é voltado como objetivo. Como no caso a segurança é a questão, o foco desse artigo será de mostrar possibilidades simples, com ferramentas básicas, mas não esquecendo de tudo o que fortalece um servidor. O que segue abaixo será a visão de um servidor diretamente, podendo se prolongar para outras máquinas.

O artigo será dividido em 2 partes, uma voltada para a questão interna, e a segunda parte voltada para a parte da rede/conexões. Serão distribuídos em momentos diferentes, porém um é complemento do outro.

O servidor, internamente:

Antes de nos preocuparmos com um servidor por fora , devemos conhecê-lo internamente, fazer com que ele seja a nossa casa. Um servidor desconhecido não é um servidor gerenciável. Por esse motivo, o ideal é que se saiba tudo, e para isso, precisamos montá-lo. Quando criamos, sabemos o que tem ali, diferentemente de quando já pegamos em um processo de produção.

O objetivo de um atacante é penetrar na máquina, não importa como. Tudo se torna muito mais fácil quando se está dentro do sistema, logo, ele PRECISARÁ de um usuário, caso queira tomar o sistema. Ciente de que o que "eles" querem é ganhar o sistema, temos que preparar para tudo que todo e qualquer possível estrago seja reduzido.

– **O desnecessário**

Nunca tenha pacotes desnecessários no sistema. Eles são uma ameaça, e podem ser prejudiciais posteriormente. Qualquer pacote que desconheça e/ou seja desnecessário, desinstale.

– **Os usuários**

Certifique-se de quem precisa ter acesso ao sistema. Um acesso shell aos usuários nunca é uma boa idéia. Limite e controle cada usuário. Saiba quem REALMENTE precisa de um acesso shell, monitore e tire privilégios dos demais.

Somente o root deve ter acesso aos arquivos passwd e shadow . Usuários comuns nem mesmo podem ter acesso a leitura, pois já seria o início de uma tentativa de quebrar a senha pelo método convencional. Ao monitorar o arquivo passwd e shadow, aproveite para tirar a shell válida de usuários desnecessários, e retorne uma shell inválida, impossibilitando uma administração mesmo que simples do sistema. Utilize o John The Ripper para testar as senhas dos usuários.

– **A shell**

Para que um atacante consiga o controle do sistema, é necessário um usuário, de preferência com uma shell. A verdade é essa, o que buscam é uma shell.

Além de evitar que diversos usuários tenha uma shell válida no sistema, podemos fazer com que cada vez que algum dos usuários, PRINCIPALMENTE o root entre no sistema, que um aviso seja enviado! Para que alguém, que não o administrador, entre no sistema? Então entra a idéia do profile.

Todo usuário quando entra no sistema executa algum tipo de profile. E é nele que pode entrar nossa primeira precaução. Vamos pegar como exemplo o BASH, que é o shell padrão dos sistemas Linux, e supor que ao ser utilizado, ele executa o ~/.bash_profile e/ou ~/.bashrc . Isso nos ajudaria, e muito.

Se nossa preocupação for com o root, vamos trabalhar em cima do /root/.bash_profile então. Não seria uma boa receber um email ou um SMS no celular toda vez que isso ocorresse ? De imediato, estaríamos a par da situação.

```
root@whitehat:~# cat .bash_profile
root@whitehat:~#
root@whitehat:~# echo "echo LOGIN NO SISTEMA | mail -s login seu_email@endereco.com.br" >> .bash_profile
root@whitehat:~# cat .bash_profile
echo LOGIN NO SISTEMA | mail -s TENTATIVA seu_email@endereco.com.br
root@whitehat:~#
```

Pronto. Toda vez que alguém utilizar o bash como root, antes de tudo ele executará o /root/.bash_profile , e com isso, um email ou SMS (depende do gosto do freguês) será enviado avisando do suposto uso indevido.

– **Dificultando o trabalho...**

Após uma invasão, o atacante tem diversas opções. Muitos fazem por prazer apenas, outros para destruir, outros para desfigurar somente as páginas, outros como “pontes” para futuros ataques, e alguns com intuito até de estudo. O fator principal é que: um sistema penetrado, não é o mesmo. Mesmo que o sistema tenha sido invadido, o ideal é que possamos dificultar um pouco o trabalho do atacante. Que fique BEM claro, vamos dificultar...impedir é muito complexo.

Uma das maneiras de manter o sistema sempre “escravo” ou acessível é tentar esconder uma invasão. Assim, são utilizados os chamados rootkits, que tem diversos níveis de interação com o sistema.

De simples modificações à implementação de módulos do kernel, os rootkits são sempre uma ameaça. E vamos tratar do básico deles: da substituição dos principais softwares utilizados para a análise do sistema, forjando e se escondendo do administrador.

Passando adiante da utilização de comandos como chown e chmod que são absolutamente NECESSÁRIOS para uma boa administração, irei falar dos comandos chattr e lsattr .

Com o chattr, é possível mudar atributos de arquivos e/ou diretórios específicos, e tomar conhecimento deles através do lsattr. Com isso, tornaremos imutáveis alguns softwares que são alvo de modificações por rootkits.

```
root@whitehat:~# cd /bin/
root@whitehat:/bin# chattr +i ps bash date dmesg kill netstat su
root@whitehat:/bin# lsattr | grep i-
----i-------- ./sh
----i-------- ./bash
----i-------- ./date
```

```
----i-------- ./ps
----i-------- ./su
----i-------- ./kill
----i-------- ./dmesg
----i-------- ./netstat
root@whitehat:~#
root@whitehat:/sbin# chattr +i arp arpd ifconfig lsmod
root@whitehat:/sbin# lsattr | grep i-
------------ ./rescan-scsi-bus
----i-------- ./lsmod
----i-------- ./arpd
----i-------- ./arp
------------ ./mii-tool
----i-------- ./ifconfig
root@whitehat:/sbin#
```

Como foi visto, demos o atribudo “i” (chattr +i) a determinados arquivos, tornando-os imutáveis. Depois, fizemos uma verificação com o lsattr. No segundo caso, vimos que apareceram 2 arquivos que não tinha atributo nenhum. Agora, o teste para verificar a imutabilidade de algum arquivo.

Vamos pegar como exemplo o /sbin/ifconfig, e tentar primeiramente modificá-lo, depois deletá-lo:

```
root@whitehat:/sbin# echo "TESTE DE MODIFICACAO" >> ifconfig
-su: ifconfig: Permission denied
root@whitehat:/sbin#
root@whitehat:/sbin# rm -rf ifconfig
rm: cannot remove `ifconfig': Operation not permitted
root@whitehat:/sbin#
```

Comprovado que o atributo funciona. Mesmo o usuário root não foi capaz de modificar ou deletar o arquivo. Assim, poderíamos dificultar um pouco a tentativa de instalação de um rootkit, porém não a torna impossível.

– **Conheça seus logs**

E existe algo mais fundamental para uma boa administração ? Conheça a fundo seus logs, monitore diariamente, crie scripts para isso, personalize...faça do /var/log o seu quintal, e cuide dele observando TUDO o que ocorre. Há diversos softwares que analizam tudo o que ocorre.

Tentativas de acessos, erros nos serviços disponibilizados... A defesa de um sistema é baseada no CONHECIMENTO do ataque. Para isso, temos que analizar, descobrir o que pretendem e o que fazem para que possamos ter uma boa gerência da segurança.

Um ótimo sistema de análise de logs é o OSAudit, desenvolvido pelo nosso amigo Daniel Cid. O software chega a fazer estatísticas, e leitura de diversos logs.

O software pode ser encontrado em http://osaudit.sourceforge.net/

– **O núcleo do sistema**

Enfim a parte mais complexa, precisa e importante. O kernel, como todos conhecem, é o núcleo do sistema. Através dele são feitas as chamadas do sistema, a interação entre hardware o software geral. O kernel é a cabeça, e danificando o kernel, tudo se acaba.

Qual a finalidade de um atacante ao ter a possibilidade de modificar o núcleo? Por que ele faria isso? Quais as vantagens? Perguntas comuns que com certeza facilitarão o trabalho de defesa e análise.

Modificar o kernel significa torná-lo adequado, atendendo suas necessidades. Atender as necessidades de um atacante é o mesmo que escondê-lo no sistema, torná-lo invisível até, fazer com que seus rastros não sejam descobertos, e, PRINCIPALMENTE, tenha a possibilidade de retornar mais tarde. Com o núcleo atendendo às essas "necessidades", um invasor pode ter uma máquina em suas mãos, que dificilmente será descoberta. Portanto, algumas medidas podem ser necessárias para que possamos evitar que alguém consiga entrar no sistema através de bugs ou vulnerabilidades do núcleo. E uma delas seria aplicar patchs que contribuam para tornar o sistema mais seguro que o comum.

Algumas regras básicas para reduzir os riscos de se ter um sistema vulnerável:

1) Mantenha o kernel sempre atualizado, confiando na versão ou série que mais te agrada. Ex: 2.4.X ou 2.6.X

2) Recompile sempre o núcleo, para que as possibilidades de exploração fiquem menores. Um kernel recompilado significa um kernel com opções apenas necessárias para um funcionamento direto, em outras palavras, recompile para o que precisa apenas.

3) Escolha entre um kernel modular ou de imagem. Procure decidir se prefere a praticidade de um kernel modular, onde a qualquer momento temos a facilidade de adicionar um módulo, ou um kernel estático, em que tudo o que precisamos fica apenas em uma imagem, fazendo com que a imagem fique mais pesada, e carregue mais opções.

PS: Geralmente bons rootkits são modulares, ou seja, modificam funções do núcleo através de módulos, fazendo com que diretórios estratégicos não apareçam, e escondendo funções e ações no sistema, tornando tudo mais vulnerável. Esse tipo de software é de ALTO risco, e uma vez com o sistema comprometido, a probabilidade de se ter um rootkit instalado na máquina é alta.

4) Utilize patchs de confiança no núcleo. Geralmente correções e/ou atualizações gerais podem demorar um pouco para sair oficialmente, e muitas vezes são lançados patchs que cobrem isso, e fazem mais ainda: oferecem funções diferenciadas como ACL´s (controle de acesso), funções paranóicas de segurança (como não permitir que usuários vejam outros processos rodando além dos que foram iniciados por ele mesmo) e pacotes de redes. Com isso, é muito válido aplicar esse tipo de patch no kernel.

Aqui serão apresentados 2 projetos conhecidos pela estabilidade, confiabilidade e rapidez com que são lançados. Muitas vezes, um kernel antigo porém com um patch pode ser mais seguro do que um kernel novo, porém padrão ou pouco customizado.

http://www.openwall.com -> Projeto OpenWall, que oferece um ótimo patch para o kernel com várias curiosidades. É também responsável pelo famoso John the Ripper, software muito rápido para quebrar senhas, e ótimo para testes no sistema.

http://www.grsecurity.net -> Grsecurity é também muito difundido, pela estabilidade proposta, e pelas características do patch. Foi o primeiro patch que usei, e atualmente uso em 1 servidor o Grsecurity e o OpenWall (owl) em outros.

Ambos são ótimos, e funcionam muito bem. Com um patch, a segurança aumenta consideralmente em comparação ao padrão.

– **Conclusão da Parte I**

Nessa primeira parte, terminei de escrever sobre o sistema pelo lado interno, ou seja, não tratei dos serviços e da rede em geral. Na próxima parte, escreverei sobre um servidor pelo lado externo, buscando meios de impedir ou minimizar um possível impacto. Falarei ainda de ferramentas comuns (lsm, bfd, iptables, snort...) para uma possível análise sobre o andamento do conjunto.

A orientação de um sistema internamente é fundamental para o seu funcionamento correto. Um servidor bem planejado é um bom candidato para um caso de sucesso, e de evitar dores de cabeça. Testes frequentes são quase uma obrigação, principalmente os de vulnerabilidades e conhecimento interno. Procure checar por usuários conectados, e vasculhe os logs para encontrar tentativas ou até mesmo para verificar quem anda utilizando a máquina. Crie scripts para fazer um monitoramento automático, e que possa enviar por email, é uma idéia. :)

- **Referências**

http://www.netnix.com.br -> Soluções gerais.
http://www.openwall.com -> OpenWall Project, owl e John The Ripper
http://www.grsecurity.net -> Grsecurity Patch
http://osaudit.sourceforge.net -> OsAudit