Backdoors x Firewalls de Aplicação Praticando em Kernel do Linux

Rodrigo Rubira Branco

rodrigo@kernelhacking.com
bsdaemon@bsdaemon.org

Agenda:

- O que são Backdoors e Quais seus Recursos
- O que são Firewalls e Firewalls de Camada de Aplicação
- Entendendo o NetBUS
- Recursos do NetBUS
- Instalando o NetBUS em Linux
- Instalando o Wine para rodar o NetBUS
- Utilizando o comando eject para permitir ejeção do CD
- Agradecimentos

Agenda:

- O que são Backdoors e Quais seus Recursos
- O que são Firewalls e Firewalls de Camada de Aplicação
- Firewalls Linux
- Firewalls de Camada de Aplicação (Surpresa)
- Backdoors que Utilizam Comandos de Aplicação
- Recursos para se Evitar Inserção de Backdoors
- Recursos para a Backdoor não ser Detectada (Brincadeira)
- Outros Recursos de Backdoors e o Futuro
- Agradecimentos

RISE RESEARCH GROUP

O que são Backdoors e quais os seus recursos

LKM x KMEM Injection

Módulos de kernel precisam ser compilados para o sistema alvo específico

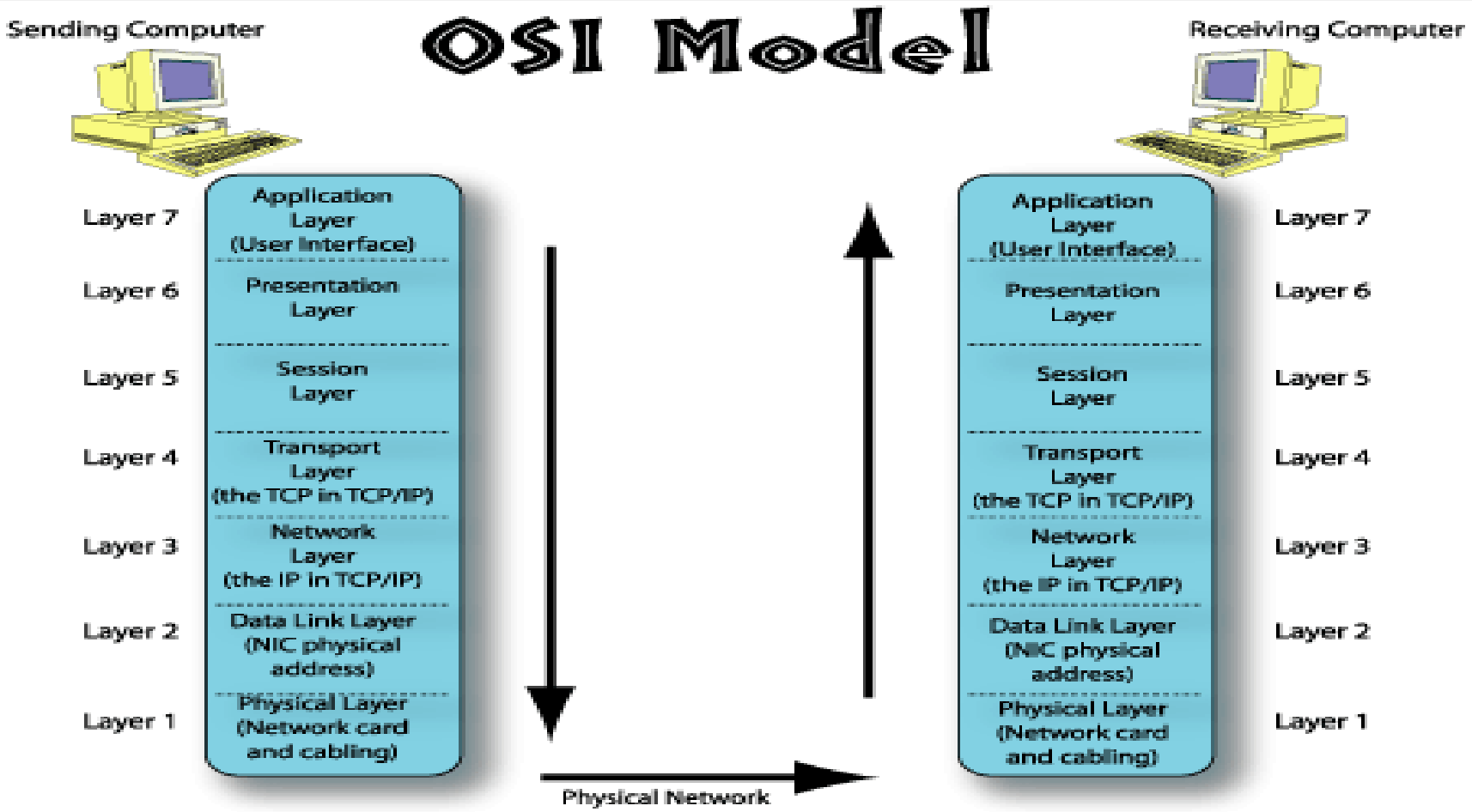
Através do /dev/kmem ou /dev/mem podemos injetar o código na memória do kernel modificando o mesmo sem o uso de módulos

Podemos via o uso de injeção de código, descobrir os endereços das funçoes necessárias, fazendo assim uma backdoor indepente de sistema e que possa ser utilizada pré-compilada

Diversos são os recursos oferecidos por uma backdoor (Redireção de processos, esconder processos, esconder modo promíscuo, shell remota, etc), não importando para nós tais recursos e sim a não detecção da backdoor.

RISE RESEARCH GROUP

O que são Firewalls e Firewalls de Camada de Aplicação



O que são Firewalls e Firewalls de Camada de Aplicação

Linux x App. Firewalls (atualmente o netfilter age apenas até a camada de transporte), não entendendo os protocolos das camadas superiores (projeto http://l7-filter.sourceforge.net/ busca adicionar esta funcionalidade ao mesmo)

Dizemos que ele atua até a camada de transporte principalmente pelo recurso de STATEFUL que este apresenta para protocolos complexos, como o FTP. Ele não conhece realmente o funcionamento da camada de aplicacão do FTP (não vê por exemplo a porta negociada para transmissão de dados), mas pode através do controle de seção e uso do RELATED,ESTABLISHED permitir apenas a conexão na porta de dados que pertença a uma conexão de controle pré-existente.

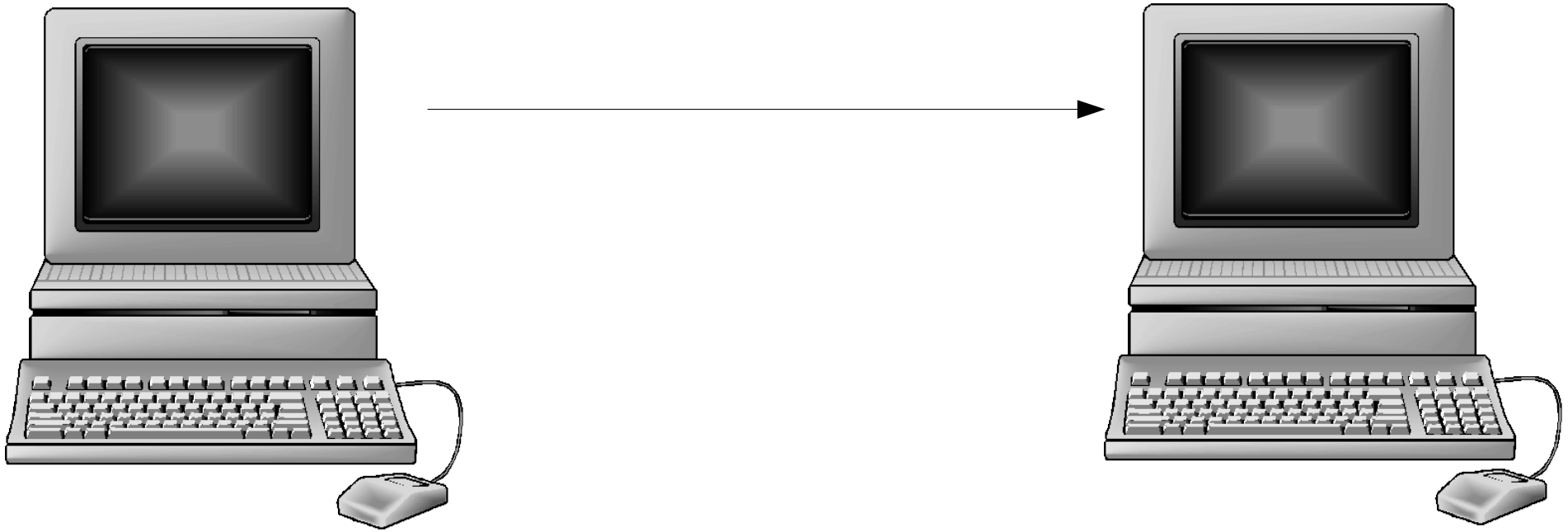
Fornecedores: MICROSOFT, Checkpoint, Cisco, etc

RISE
RESEARCH GROUP

Firewalls Linux (*) x Backdoors

Máquina alvo com Firewall ativado permitindo apenas conexões SSH (porta 22) e HTTP (porta 80)

Suckit é executado na máquina alvo e o acesso será feito a mesma (com um tcpdump aberto para se demonstrar que o destino é realmente a porta 80)



RISE
RESEARCH GROUP

Firewalls de Camada de Aplicação (Microsoft ISA Server)

O senhor Aylton da Microsoft ofereceu apoio não-oficial a esta apresentação por isso fica aqui o agradecimento e a viabilidade da demonstração prática da idéia. O appliance oferecido foi empresado pela SCUA Security e também agradeço a mesma.

ISA Server

Backdoors que utilizam comandos de aplicação (*)

O que aconteceria no exemplo anterior, se a backdoor recebesse e respondesse aos pedidos, através de comandos do protocolo da camada de aplicação?

No exemplo a seguir, nossa backdoor apenas recebe comandos e entende a camada de aplicação (parser do método GET) através da interceptação da syscall read(). Os retornos podem ser gerados no /var/www mesmo que nobody não possa escrever neste diretório, bastando acessar o servidor de páginas normalmente e verificá-los.

Nenhum Firewall conseguiria dissernir tal tipo de tráfego de aplicações normais do servidor (esta backdoor poderia ser utilizada via outros protocolos e não apenas HTTP).

RISE
RESEARCH GROUP

Backdoors que utilizam comandos de aplicacao (*)

Em um artigo ainda não publicado pelo Sr. Aylton (da Microsoft), o qual tive acesso, o mesmo menciona técnicas para elaboração do perfil da aplicação, que poderiam ser utilizadas para evitar-se SQL Injection e outras técnicas de mal-uso de aplicações web (consequentemente também o acesso a backdoors utilizando-se do protocolo HTTP), embora sua eficiência seja duvidosa, pois as requisições e respostas da backdoor podem ser legítimas mesmo (acesso a páginas do site de forma real).

RISE
RESEARCH GROUP

Recursos para se evitar inserção de backdoors

Limitar recursos do usuário root (lskm, grsec, rbac e outros) – evita-se carregamento de módulos, mas não serve contra subversão do kernel.

Proteger o kernel (stmichael) – tenta-se evitar subversões do kernel com recursos de voltar a um estado antes da mesma ocorrer ou até desligar o equipamento por completo – desenvolvido por Timothy Lawless e atualmente mantido por Rodrigo Rubira Branco (eu :)

Backdoors em modo usuário – através da infecção de processos e arquivos pode-se adicionar uma backdoor em modo usuário sem subversões diretas ao kernel (ou com recursos de aplicar patchs no kmem quando solicitados) – tais backdoors podem ser evitadas com verificações de hash (aide/tripwire) do sistema e recursos de ACL.

RISE
RESEARCH GROUP

Recursos para a Backdoor não ser Detectada – Detonando Chkrootkit e KSTAT

Chkrootkit – Criado por Nelson Murilo, palestrante desta e da anterior edição do H2HC. Ferramenta para checagem de rootkits que roda em modo usuário, e que como o próprio autor sempre admite, não é absoluta, apenas auxilia para detecção de rootkits mais simples.

Kstat – Criado por FuSyS, da SoftProject Digital Security for Y2K – Ferramenta que visa detectar rootkits através da leitura do /dev/kmem assim que instalado o sistema e gravação em arquivo dos resultados. Quando se executa, ele compara os endereços atuais com os armazenados originalmente, identificando mudanças.

RESEARCH GROUP

Recursos para a Backdoor não ser Detectada – Detonando Chkrootkit (*)

Diversas formas existem para se passar pelo chkrootkit, analisando algumas delas podemos citar:

- Alias do comando unalias seguido por alias dos comandos utilizados pelo chkrootkit (funciona apenas em alguns shells) – Rodrigo Rubira Branco

```
### workaround for some Bourne shell implementations
unalias login > /dev/null 2>&1
unalias ls > /dev/null 2>&1
```

- Detecção do chdir para /usr/lib/chkrootkit (Rodrigo Rubira Branco)

```
# the base chkrootkit is designed to be run from it's build directory,
# therefor it uses "./" as a prefix to all it's executables.  we need to
# change to /usr/lib/chkrootkit to keep this working
cd /usr/lib/chkrootkit
```

RESEARCH GROUP

Recursos para a Backdoor não ser Detectada – Detonando KSTAT (*)

Se interceptarmos a tentativa de leitura do kstat no /dev/kmem e retornarmos os hacks ao estado original, efetuarmos o read e logo após setarmos os hacks novamente, o kstat não conseguirá identificar os ataques.

RESEARCH GROUP

Outros Recursos de Backdoors e o Futuro (*)

Uma backdoor pode simplesmente esperar por um payload contendo um rootkit a ser inserido no kernel (via /dev/kmem ou /dev/mem por exemplo).

Tal backdoor pode rodar em modo kernel ou simplesmente ser um aplicativo modo usuário que realize tal operação.

O exemplo aqui apresentado ainda não está 100% funcional, mas permite verificarmos a inserção do código na memória do kernel e o trap realizado, embora o endereço em si do rootkit ainda não esteja sendo atingido e o mesmo não acabe sendo executado.

RESEARCH GROUP

Agradecimentos

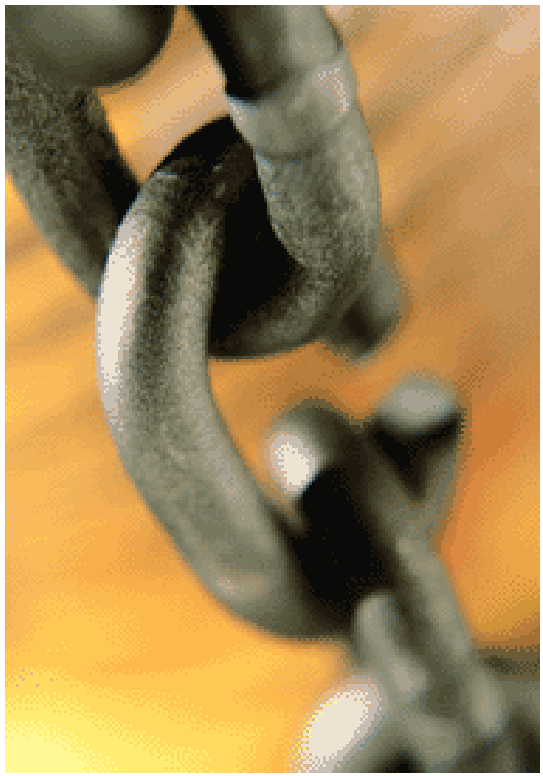
- **Aylton -> Microsoft**

- **Sergio -> Microdevices**

- **Angelo -> SCUA**

- **Organização do H2HC**

- **Todos os presentes pela paciência**



FIM! Será mesmo?

DÚVIDAS ?

Rodrigo Rubira Branco

rodrigo@kernelhacking.com
bsdaemon@bsdaemon.org