



Guia do Administrador

Beraldo Leal <beraldo@unp.br>

Kernel Intrusion Detection System - KIDS

Beraldo Leal <beraldo@unp.br>

Guia do Administrador - versão 0.1
16 de Novembro de 2007

Resumo

Este manual descreve o funcionamento bem como a arquitetura do KIDS. Ele aborda a instalação, configuração e administração do software.

Nota de Copyright

Copyright © 2007 Beraldo Leal

Este manual é software livre; pode ser redistribuído e/ou modificado sobe os termos da Licença Pública Geral GNU, publicada pela Free Software Foundation; Uma outra versão 2, ou (se quiser) qualquer versão posterior.

Este documento se distribui com a esperança de que seja útil, pois vem *sem nenhuma garantia*; nem se quer a implícita garantia de comerciabilidade ou conveniência para um fim em particular. Ver a Licença Pública Geral GNU para mais detalhes.

Você deveria receber uma cópia da Licença Pública Geral GNU com o KIDS, em COPYING ou no GNU website (<http://www.gnu.org/copyleft/gpl.html>). Se não, escreva para a Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Sumário

1	Introdução	1
1.1	Disponibilidade	1
1.2	Visão Geral	1
1.3	Como o KIDS funciona?	2
2	Instalando o KIDS	3
2.1	Requisitos mínimos	3
2.2	Obtendo os fontes	3
2.3	Compilando	4
2.4	Instalação	4
3	Gerenciando o KIDS	7
3.1	Configurando os scripts de inicialização	7
3.2	Inicializando o KIDS	7
3.3	Arquivos de configuração	8
3.4	Manipulando as regras, utilizando um cliente	10
3.5	Analisando os logs	11

Capítulo 1

Introdução

1.1 Disponibilidade

O site oficial do Kernel Intrusion Detection System - KIDS, é <http://sourceforge.net/projects/ids-kids/>. Lá você irá encontrar informações sobre o KIDS, bem como os arquivos da lista de discussão, bugtrack, atualizações do projeto e também sempre a versão mais atualizada deste documento.

Envie qualquer bug ou comentário sobre o projeto para <beraldo@unp.br>. Você também pode usar o Bug Tracking System do Source Forge, no site do projeto para reportar erros.

Atualmente o KIDS, só se encontra disponível para download por meio de seus fontes, sendo necessário a compilação do mesmo, para o seu funcionamento, como será descrito mais adiante.

1.2 Visão Geral

O Kernel Intrusion Detection System – KIDS, é um Network Intrusion Detection System - NIDS, ou seja, um sistema de detecção de intrusos, porém sua principal função, capturar e comparar pacotes com regras pré-existentes, é exercida no kernel em forma de módulo.

Atualmente as aplicações que desempenham o papel de um IDS, sempre fazem o tratamento dos pacotes no userspace, geralmente utilizando-se de uma biblioteca bastante conhecida, a libpcap. Entre as aplicações que utilizam esta biblioteca, encontra-se os conhecidos sniffers tcpdump e ethereal, e o IDS mais estável até o momento da escrita deste documento, o SNORT, onde o seu desenvolvimento iniciou-se por volta de 1998.

A idéia básica do KIDS, é exercer as funcionalidades básicas de um IDS, porém sem a utilização da libpcap, tratando os pacotes no kernelspace, pretendendo com isso obter um ganho na performance do sistema como um todo.

1.3 Como o KIDS funciona?

A arquitetura do KIDS foi projetada para que a principal tarefa exercida por um Network IDS, que é a comparação dos pacotes de rede com as regras existentes em uma base, seja feita da forma simples e o mais rápido possível, por isso, o desejo de se executar esta ação no kernelspace. As outras atividades, como manipulação das regras, gerenciamento de logs, sistema de alertas, entre outras, serão todas manipuladas no userspace.

O KIDS é constituído de: a) um módulo para o kernel, chamado de `kids_mod`, responsável pelo processamento dos pacotes que passam pelo framework netfilter; b) um daemon servidor, `kidsd`, responsável pelo gerenciamento das regras, estabelecimento de conexões (para gerenciamento das regras, estatísticas, etc...) e principalmente para fazer a comunicação com o módulo e c) uma aplicação cliente no userspace, o `kids manager (kidsm)`, responsável pela comunicação com o servidor, sendo esta a que será utilizada pelo administrador do sistema para gerir as regras.

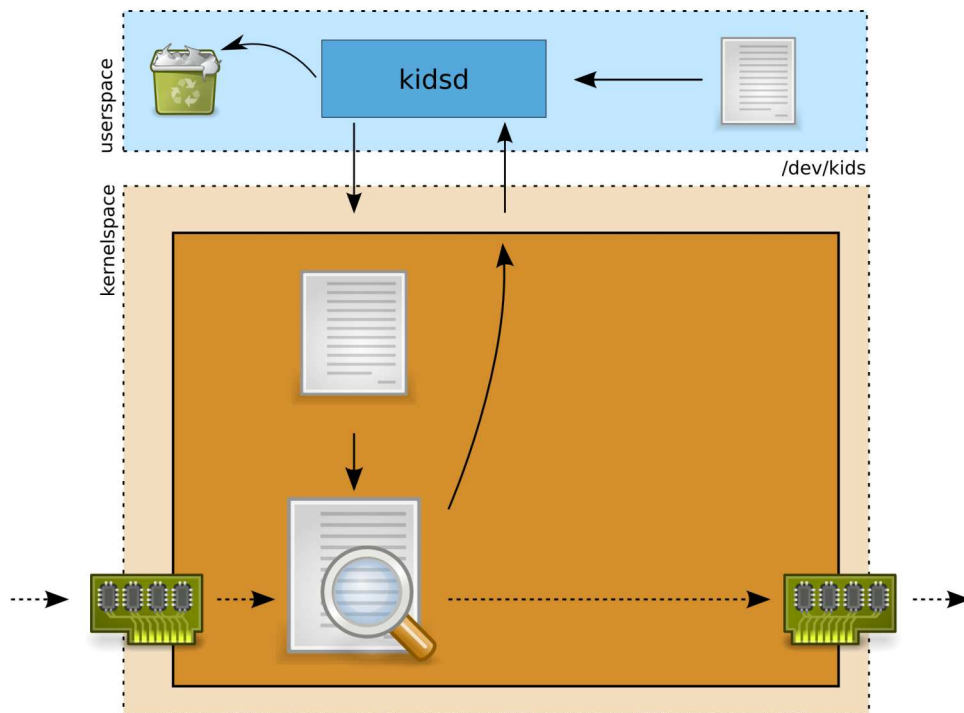


Figura 1.1: Funcionamento interno do KIDS

Capítulo 2

Instalando o KIDS

2.1 Requisitos mínimos

Como atualmente, o KIDS só encontra-se disponível para download via código fonte, sua compilação é necessária para o seu funcionamento, por tanto, os seguintes itens são necessários para a instalação dos componentes do KIDS:

gcc: Foi testado com o versão 4.1 do compilador gcc, porém outras versões, deve funcionar sem maiores problemas.

make: Utilizado para automatizar o processo de compilação.

libc6-dev: Biblioteca de desenvolvimento GNU C.

libopts25-dev: Biblioteca necessária pelo parser do KIDS.

linux-libc-dev: Arquivos de cabeçalhos necessários pelo módulo do kernel.

linux-headers: Arquivos de cabeçalhos em comum, necessários pelo módulo do kernel. Foi utilizado a versão 2.6.22, é fortemente aconselhado a utilização desta versão do kernel ou superior, bem como o kernel a ser executado na máquina, devido a nova estrutura skb.

Basicamente se você tem um ambiente de desenvolvimento instalado em sua máquina, onde você consegue compilar arquivos em `.c`, mais estas dependencias citadas a cima, não terá muito problemas para a compilação do KIDS.

2.2 Obtendo os fontes

Faça o download do código fonte do KIDS no site do projeto ¹. Após o download, descompacte o arquivo em qualquer diretório do seu sistema, preferencialmente em `/usr/src`:

¹<http://sourceforge.net/projects/ids-kids/>

```
# pwd
/usr/src
# tar zxf kids-0.1b.tgz
```

Após descompactar o arquivo, os diretórios `kids_mod`, `kidsmekidsd` serão criados dentro do diretório `kids-0.1b`. Estes diretórios contêm os fontes dos módulos do KIDS, separadamente.

2.3 Compilando

Com os arquivos descompactados, basta entrar em cada subdiretório e executar `make` para que cada componente do kids seja compilado. Caso você possua todos os requisitos, as saídas serão parecidas com a saída abaixo:

```
# pwd
/usr/src/kids-0.1b
# cd kidsd
# make
gcc -c common.c tcphandleconnection.c parser.c rules.c ioctl.c
`autoopts-config cflags ldflags`
gcc: -R/usr/lib: linker input file unused because linking not done
gcc: -lopts: linker input file unused because linking not done
gcc main.c common.o tcphandleconnection.o parser.o rules.o ioctl.o
-o kidsd -lpthread `autoopts-config cflags ldflags`
# cd ../kidsm
# make
gcc -o kidsm main.c splitline.c common.c sends.c
# cd ../kids_mod
# make
make -C /lib/modules/2.6.22-2-486/build
SUBDIRS=/usr/src/kids-0.1b/kids_mod modules
make[1]: Entering directory `/usr/src/linux-headers-2.6.22-2-486'
  CC [M] /usr/src/kids-0.1b/kids_mod/kids_mod.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC /usr/src/kids-0.1b/kids_mod/kids_mod.mod.o
  LD [M] /usr/src/kids-0.1b/kids_mod/kids_mod.ko
make[1]: Leaving directory `/usr/src/linux-headers-2.6.22-2-486'
```

2.4 Instalação

O processo de instalação é bem simples, ele irá copiar os binários (`kidsd` e `kidsm`) para `/usr/sbin`, irá criar um diretório para guardar os arquivos de configuração em `/etc/kids`,

colocando dentro deste diretório os dois arquivos de configuração: `kidsd.conf` e `kidsd.rules`. E por fim, irá copiar o módulo `kids_mod.ko`, para `/lib/modules/$(uname -r)/kernel/net/ipv4/`.

Para instalar, basta digitar, em cada diretório:

```
# make install
```


Capítulo 3

Gerenciando o KIDS

3.1 Configurando os scripts de inicialização

Para automatizar o processo de inicialização, tanto do daemon quanto do módulo do kernel `kids_mod`, existe um script de inicialização em `scripts/kids`. Copie ele para o diretório de inicialização dos scripts:

```
# pwd
/usr/src/kids-0.1b
# cp scripts/kids /etc/init.d/
```

Feito isto, basta executar o comando `update-rc.d` para que os links simbólicos sejam criados nos respectivos diretórios dos runlevels:

```
# update-rc.d kids defaults
Adding system startup for /etc/init.d/kids ...
/etc/rc0.d/K20kids -> ../init.d/kids
/etc/rc1.d/K20kids -> ../init.d/kids
/etc/rc6.d/K20kids -> ../init.d/kids
/etc/rc2.d/S20kids -> ../init.d/kids
/etc/rc3.d/S20kids -> ../init.d/kids
/etc/rc4.d/S20kids -> ../init.d/kids
/etc/rc5.d/S20kids -> ../init.d/kids
```

3.2 Inicializando o KIDS

Feito isso, você poderá a qualquer momento inicializar ou parar o kids, bastando que para isso execute o script com o argumento `start` ou `stop` como mostrado abaixo:

```
# /etc/init.d/kids start
```

Pronto, neste momento, o módulo `kids_mod` encontra-se no `kernel`space, capturando os pacotes que chegam, existe também um `daemon`, que foi inicializado, para que conexões vindas na porta 8107 (default), possam ser gerenciadas, e aplicações clientes possam manipular as regras que estão no `kernel`space.

Observe também que um dispositivo de caractere (`/dev/kids`) foi criado para que o `daemon` possa se comunicar com o módulo.

```
# lsmod | head -3
Module                Size  Used by
kids_mod              9160   0

# ls -la /dev/kids
crw-r--r-- 1 root root 100, 0 2007-11-26 21:37 /dev/kids

# tail -3 /var/log/syslog
Nov 26 21:37:27 kids kernel: kids_mod: Enabling kids.mod version 0.1
Nov 26 21:37:27 kids kernel: kids_mod: /proc/kids/ created sucessfully
Nov 26 21:37:27 kids kernel: kids_mod: rule 0 add sucessfully.
Nov 26 21:37:27 kids kidsd[8758]: Server listing at 127.0.0.1:8107

# lsof -P -n -i TCP:8107
COMMAND  PID      USER    FD   TYPE DEVICE SIZE NODE NAME
kidsd    10633    root     3u   IPv4  23467      TCP 127.0.0.1:8107 (LISTEN)
```

No exemplo acima, apenas uma regra foi adicionada ao módulo pelo `daemon`, e percebe também que o `daemon` encontra-se em estado de `LISTEN` na porta 8107, aguardando por conexões.

3.3 Arquivos de configuração

Basicamente, os arquivos de configuração do KIDS, são dois: `/etc/kids/kidsd.conf` e `/etc/kids/kidsd.rules`, no primeiro, você configura variáveis gerais para o funcionamento do KIDS, inclusive onde encontra-se o arquivo de configuração de regras que será carregado na inicialização do `daemon`, já o segundo é o próprio arquivo de regras do IDS.

A sintaxe dos dois arquivos é bastante simples, comentários são permitidos nos dois, bastando iniciar a linha com um `#`, e os valores devem levar em consideração o `case`, pois são sensíveis ao `case`.

Alguns exemplos de parâmetros básicos, possíveis em `/etc/kids/kidsd.conf`:

```
# The banner with will be send to the clients at connect.
ServerBanner          "Kidsd - beta version"
```

```
# ServerAddress configures the address to listen.
ServerAddress          127.0.0.1

# DefaultPort configures the port to listen.
DefaultPort           8107

# Timeout - (sec)
ConnectionTimeout     300

# Device filename to write and read from kids module in kernelspace.
DeviceName            "/dev/kids"

# If you need send messages to Syslog, put True, else False.
SysLog                "True"

# RulesFile defines the file of rules to load in kids.mod.
RulesFile              "/etc/kids/kidsd.rules"
```

Como você pode perceber, neste arquivo, configura-se: Em que porta o servidor irá tratar as requisições; Qual o dispositivo de caractere que ele irá se referenciar, para estabelecer uma comunicação com o módulo; Qual o tempo máximo que um cliente poderá permanecer conectado sem atividade; Se irá logar as mensagens, enviando-as para o daemon syslog; E onde encontra-se o arquivo de regras.

O arquivo de regras `/etc/kids/kidsd.rules` basicamente é constituído de uma regra por linha, e caracteriza-se como uma regra, as linhas iniciadas pela palavra reservada **rule** seguido da regra propriamente dita. A syntax da regra é

```
rule <protocol> <dst_port> <content>
```

Onde, **protocol** é um número inteiro que identifica que protocolo será analisado (6 para TCP e 17 para UDP), **dst_port** é a porta de destino para onde o pacote está sendo enviada, e **content** é o conteúdo, a string, que caracteriza um pacote malicioso. Abaixo um simples exemplo de um arquivo de regras:

```
# BOT IRC Traffic Detected By Nick Change
rule 16 6667 NICK

# IMAP GNU Mailutils imap4d hex attempt
rule 6 143 SEARCH TOPIC %

# Connect Direct Server - Session Terminated Invalid Credentials
rule 6 1364 SVTM056I
```

```
# Microsoft Messenger phishing attempt - corrupted registry
rule 17 1025 FAILURE TO ACT NOW MAY LEAD TO DATA LOSS AND CORRUPTION!

# Possible BlackWorm or Nymex infected host
rule 6 80 /cgi-bin/Count.cgi?df=765247
```

As simples regras acima, foram utilizadas com base nas regras *community* do Snort.

3.4 Manipulando as regras, utilizando um cliente

Apenas para propósitos administrativos, pensando em facilitar a vida do administrador, foi desenvolvido uma aplicação cliente, o Kids Manager *kidsm*, com ela você pode conectar na porta 8107 (default), do daemon, para manipular as regras. Para se conectar informe, o ip do servidor e a porta que deseja se conectar, isso, na linha de comando. Uma console será aberta, caso a conexão seja estabelecida. Observe:

```
# kidsm -s 127.0.0.1 -p 8107
Connecting to 127.0.0.1:8107 ... connected.
101 Kidsd - beta version

kidsd> help
kidsd - version 0.1 Help.
This help, print the basics syntax for send commands to kids daemon.

Possible commands:

help          Print this help.
show rules    Show rules actives in the server.
show tot      Show just only the total of rules in the server.
flush rules   Remove ALL rules in the server.
del rule N    Remove the rule with rule_id = N.
add rule RULE Add a rule to server, at runtime. (Check syntax of RULE)
quit         Disconnect to server and exit.

kidsm>
```

Através desta interface, você pode:

show rules: Solicita ao servidor uma lista de todas as regras que ele conhece em runtime. Isto irá mostrar: **rule_id**, **protocol**, **dst_port** e **content**.

show tot: Mostra apenas o total de regras no servidor.

flush rules: Remove todas as regras do servidor.

del rule N: Remove apenas uma regra, com **rule_id** N.

add rule RULE: Adiciona uma nova regra no servidor, RULE deve ser informado na sintaxe das regras: **protocol**, **dst_port** e **content**. Exemplo: **add rule 6 80 attackstring**

3.5 Analisando os logs

No arquivo de configuração do daemon, como visto anteriormente, você pode configurar se o daemon irá ou não enviar logs para o servidor de logs, **syslog**, caso esta opção esteja habilitada, os logs do daemon, dependendo de como estiver configurado o seu **syslog**, vão para `/var/log/daemon.log`.

Todos os logs do módulo do kernel, vão para o arquivo `/var/log/kern.log`.