Ferramentas de Segurança com RPi: Snort e Kismet[1]

Adriana V. M. Azevedo, Marcos G. A. Oliveira, Robson L. M. Rodrigues, Susanny B. Oliveira, Vitor H. Barros

Instituto Metrópole Digital – Universidade Federal do Rio Grande do Norte (UFRN)
Av. Cap. Mor Gouveia, 3733 – Lagoa Nova, Natal – RN – Brazil

dricavma@hotmail.com, marcos_gabriel_araujo@hotmail.com,
laelrodrigues7@gmail.com, sboliveira@outlook.com.br,
vitorbarros@ufrn.edu.br

Abstract. *This meta-paper describes the implementation of the snort and kismet network traffic monitoring tools installed on Raspberry PI 2. Describing the configuration of Raspberry and installing the tools on it. The data capture process is described in the two tools.*

Resumo. *Este meta-artigo descreve a implementação das ferramentas de monitoramento de tráfego de rede snort e kismet, instaladas no Raspberry PI 2. Sendo descrito a configuração do Raspberry e a instalação das ferramentas no mesmo. O processo de captura dos dados é descrito nas duas ferramentas.*

1. Introdução

Este documento relata o processo de instalação e utilização dos sistemas de detecção de intrusão, Snort e Kismet.

A proposta de trabalho é "Ferramenta de segurança com RPi: Snort / Suricata e Kismet". O Snort, Suricata e Kismet têm código fonte aberto, são sniffers e sistemas de detecção de intrusão. A ferramenta Kismet funciona num ambiente wireless, já as ferramentas Snort e Surica trabalham em ambiente Ethernet. Tanto o Snort como Suricata são multiplataformas e têm o mesmo conjunto de vantagens para detecção de intrusão de rede.

O Suricata é um mecanismo que trabalha com multithread, oferecendo maior velocidade e eficiência na análise do tráfego de rede, pois tem uma capacidade utilização dos recursos mais atuais dos equipamentos. Porém, como é uma ferramenta mais recente, disponibilizado em 2010, não tem documentação tão ampla e detalhada como o snort.

O grupo analisou e foi decidido que iria ser implementado o snort, devido documentação de apoio ampla e detalhada.

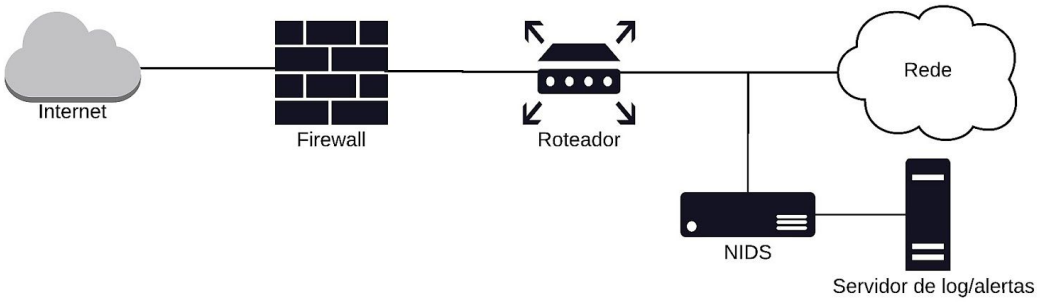
---

[1] Este trabalho é requisito parcial para obtenção de aprovação na disciplina Redes de Computadores do semestre 2017.2, no Curso de Bacharelado em Tecnologia da Informação, da Universidade Federal do Rio Grande do Norte. Professor da disciplina: Wellington Silva de Souza.

Nas seções seguintes haverá uma breve apresentação sobre Sistema de Detecção de Intrusão e Raspberry PI 2. O Snort e Kismet são apresentados nas seções 4 e 5, respectivamente, com a descrição de suas instalações.

2. Sistema de Detecção de Intrusão na Rede (NIDS - Network Intrusion Detection System)

Em uma Rede de computadores corporativa ou doméstica existe uma gama de dados, pacotes, que trafegam continuamente na rede. Esse tráfego pode ser monitorado e analisado em busca de irregularidades. O NIDS monitora continuamente uma rede de computadores ou um segmento de redes em busca de intrusão (falhas). É importante frisar que uma intrusão configura-se quando um ataque a rede ocorre e é bem sucedido.



O sistema de detecção realiza o monitoramento do tráfego, com a interface de rede atuando em modo "promíscuo" (monitoramento). O software consegue classificar o tráfego como suspeito ou legítimo pela coleta, análise e armazenamento contínuo das informações trafegadas, depois ocorre a classificação desse tráfego como suspeito ou legítimo. São definidos assinaturas ou padrões conhecidos e através deles, os dados capturados são analisados, seus cabeçalhos e conteúdos. Existem diversas ferramentas que implementam este sistema, tais como, Snort, Suricata, RealSecure, Kismet entre outras. O Snort e o Suricata são baseados em assinaturas, que fazem a inspeção dos eventos atuais e a comparação das suas assinaturas, regras, com os dados da rede.

3. Raspberry PI 2

O Raspberry PI 2 é um computador de placa desenvolvido pela Raspberry Pi Foundation. Ele pode ser conectado a um monitor ou TV e dispõe de entradas USB para mouse e teclado. Seu hardware é composto por um processador ARM Cortex-A7 quad-core 900MHz, 1GB de RAM e uma unidade de processamento gráfico VideoCore IV.

O desenvolvimento do processo ocorreu primeiramente pela instalação e configuração da Raspberry PI 2. Nesse processo, o primeiro passo foi a formatação do cartão de memória do Raspberry utilizando o programa SDformatter. Na documentação oficial do Raspberry foi feito o download e instalação do Raspbian, porém devido a documentação oficial do snort não dar suporte ao Raspbian, foi instalado o Ubuntu Mate Server. A configuração do ocorreu com o auxílio de um monitor com entrada HDMI,

teclado USB e mouse USB. Após a instalação do Ubuntu Mate o serviço SSH foi instalado e configurado, habilitando a porta 22, para poder trabalhar com acesso remoto. O pacote Net-tools, conjunto de ferramentas do linux, foi instalado para dar suporte à rede.

Com o Raspberry configurado começou a busca pela documentação das ferramentas para instalação, porém ocorreram algumas dificuldades e foi decidido fazer a instalação da ferramenta snort inicialmente em uma máquina virtual (Virtual Machine - VM). Não foi possível instalar o Snort no Raspberry devido a problema de dependências.

Devido aos problemas apresentados na instalação do Snort no Raspberry e incompatibilidade de instalar o Snort e o Kismet no mesmo equipamento, instalou-se o Kismet aproveitando a mobilidade que o Raspberry permite.

4. Snort

Criado por Martin Roesch em 1999, o Snort é um sistema aberto de detecção e prevenção de intrusos no tráfego de rede. O sistema pode realizar análises de tráfego em tempo real e log dos pacotes em rede do protocolo IP. Pode também realizar análises de protocolos, pesquisas de conteúdo e correspondência, como também faz detecção de ataques ou sondas, incluindo, tentativas de impressão digital do sistema operacional, ataques de URL, semântica, estouro de buffer, sondas de bloco de mensagens do servidor e varredura de portas.

O Snort tem o logotipo de um porco, pois ele fareja e captura todos os pacotes que trafegam na rede, realizando uma comparação com o conjunto de regras existente que foram gravadas no banco de dados. Os pacotes são filtrados com as regras, com a finalidade de procurar assinaturas de ataques conhecidas. Funciona tanto no sistema operacional Unix como no Windows, mas necessita da biblioteca Libcap (Linux) e Winpcap (Windows).

Uma limitação da ferramenta é que os ataques identificados são aqueles que ocorrem através de pacotes, ou em um fluxo, porém quando o ataque acontece numa falha a nível de aplicação a ferramenta não consegue identificá-lo, como no caso do Tomcat e Java.

O snort pode ser configurado em três modos distintos:

1. Sniffer mode (modo farejador): simplesmente captura os pacotes da rede e os mostra num fluxo contínuo. O Snort pode ser executado:

   ./snort

   com as seguintes opções:

Opção	Descrição
-v	Mostrará o IP e os Cabeçalhos TCP / UDP / ICMP na tela
-vd	Mostrará o IP e os Cabeçalhos TCP / UDP / ICMP na tela e os dados do

	aplicativo em trânsito
-vde	Mostrará o IP e os Cabeçalhos TCP / UDP / ICMP na tela e os dados do aplicativo em trânsito juntamente com os cabeçalhos da camada de link de dados (par

2. Packet Logger mode (modo base de logs): grava os pacotes no disco. Para ser possível gravar os pacotes em disco é necessário especificar um diretório, usualmente chama-se de log, onde ficam gravados os pacotes após a execução do seguintes comando:

   ./snort -dev -l ./log

   Todos os pacotes que são coletados serão colocados em uma hierarquia de diretórios com base no endereço IP de um dos hosts no datagrama, quando o endereço de rede não é especificado o Snort pode utilizar o endereço do computador remoto ou o host local, caso queira especificar um rede doméstica é preciso determiná-la. Abaixo tabela de opções de atributos para utilização do Snort no modo packet logger.

Opção	Descrição
-h <net address>	Informará que deseja registrar os pacotes relativo a essa rede especificada.
-b	Realizará o log no modo binário, pacotes mais compactos. (Comando: ./snort -l ./log -b)
-r	Modo de reprodução, ler pacotes binários. (Comando: ./snort -dv -r packet.log)

3. Network Intrusion Detection System mode (modo de detecção de intrusão): realiza detecção e análise no tráfego de redes. Há várias maneiras de implementar o Snort nesse modo, o padrão dos mecanismos de registro e alerta é fazer o login no formato ASCII decodificado e usar alertas completos. Os modos de alerta são mais complexos e estão listados no quadro:

Opção	Descrição
-A fast	Modo alerta rápido. Emite um alerta no formato simples com a marcação do tempo, a mensagem de alerta, fonte e destino IP's/portas
-A full	Modo de alerta total. É o modo de alerta padrão

-A unsock	Envia um alerta ao socket UNIX que outro programa pode estar escutando
-A none	Desabilitar os alertas
-A console	Envia um alerta “fast-style” para a tela
-A cmg	Gera uma alerta “cmg style”

O registro de pacotes pode ser completamente desativado pela flag -N e pode-se enviar alertas ao syslog com o -s.

4.1. Instalação do Snort

O Snort pode funcionar sem qualquer software de suporte, porém para um melhor funcionamento é interessante instalar três software de suporte:

- Barnyard2: interpretador de código aberto para arquivos de saída binários Unified2 do Snort que grava o resultado em um banco de dados SQL. O uso principal é permitir que o Snort escreva no disco de forma eficiente, deixando a tarefa de analisar dados binários em vários formatos para o barnyard, assim o Snort não perde o tráfego da rede.
- PulledPork: gerenciador de regras para o Snort. Ele ajuda a automatizar o processo de download e instalação/atualização das regras do Snort.
- Base: interface gráfica baseada na web para visualização e limpeza de eventos snort. A interface permite a classificação de alertas de grupo, a exibição de diagramas e a busca de alertas de acordo com diferentes critérios.

Um pré-requisito para instalação do Snort no Ubuntu é a instalação da biblioteca de compressão zlibg, existem mais quatro bibliotecas opcionais que melhoram o funcionamento do Snort, que são liblzma-dev, openssl e libssl-dev. Também necessita das bibliotecas HTTP / 2 C que implementa HPAC, que faz a compressão do cabeçalho.

Todo processo de instalação do Snort ocorreu com o auxílio de um tutorial [1] que seguimos passo a passo.

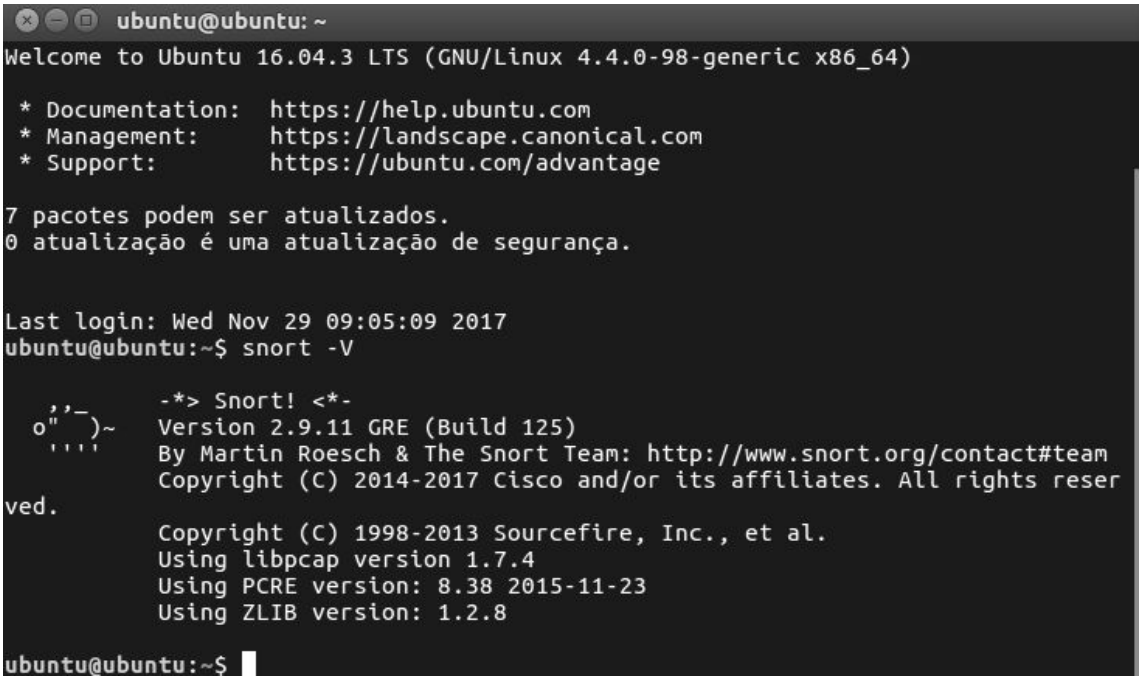
Foram instalados todas as bibliotecas recomendadas e no início do download foi necessário descobrir qual a versão do Snort atual, pois no tutorial a versão era anterior. Realizamos a instalação com a versão para 2.9.11.

```
cd ~/snort_src
wget https://snort.org/downloads/snort/snort-2.9.11.tar.gz
tar -xvzf snort-2.9.11.tar.gz
cd snort-2.9.11
./configure --enable-sourcefire
make
```

sudo make install

É importante atualizar as bibliotecas compartilhadas, pois quando tenta executar o Snort sem fazer essa atualizar apresenta-se erro.

Foi testado para ver se o Snort estava executando corretamente e obteve a seguinte tela:



Para garantir que o snort funcione de maneira correta é necessário verificar o HOME_NET e EXTERNAL_NET no /etc/snort/snort.conf, o HOME_NET precisa está com endereço IP da rede com sua máscara a qual o Snort está monitorando, caso seja alterado a rede onde será executado o Snort esse arquivo de configuração precisa ser atualizado com o novo endereço de rede. Essa informação se encontra no arquivo de configuração, snort.conf, na linha 45.

No processo de instalação do Snort foi definido que seria utilizada uma regra simples para confirmar o funcionamento da ferramenta e então foi seguida a orientação do tutorial para habilitar o arquivo de local.rules (snort.conf linha 546).

Com o Snort funcionando realizou-se a instalação do modo NIDS, para esse modo poder funcionar corretamente é necessário que o Snort não seja executado como root, e para isso ocorre precisa-se criar uma conta e um grupo não privilegiados onde o Daemon possa ser executado (snort:snort). Foram criados arquivos e diretórios com permissões, que são exigências de instalação do Snort.

Conteúdos	Path

Configurações e arquivos de regras	/etc/snort
Alertas	/var/log/snort
Regras compiladas	/usr/local/lib/snort_dynamicrules

Os arquivos de configurações são:

- classification.config
- file magic.conf
- reference.config
- snort.conf
- threshold.conf
- attribute table.dtd
- gen-msg.map
- unicode.map

A árvore de diretórios do Snort fica assim:

```
ubuntu@ubuntu:~$ tree /etc/snort
/etc/snort
├── attribute_table.dtd
├── barnyard2.conf
├── classification.config
├── disablesid.conf
├── dropsid.conf
├── enablesid.conf
├── file_magic.conf
├── gen-msg.map
├── modifysid.conf
├── preproc_rules
├── pulledpork.conf
├── reference.config
├── rules
│   ├── iplists
│   │   ├── black_list.rules
│   │   ├── default.blacklist
│   │   └── white_list.rules
│   ├── local.rules
│   └── snort.rules
├── sid-msg.map
├── snort.conf
├── so_rules
├── threshold.conf
└── unicode.map

4 directories, 20 files
```

Para verificar se ocorreu corretamente a validação das configurações usa-se o seguinte comando num usuário:

sudo snort -T -i eth0 -c /etc/snort/snort.conf (1)

Sendo o -T para testar o arquivo de configuração, a flag -i especifica qual interface o snort vai monitorar e -c qual arquivo de configuração vai ser utilizado.

Após o Ubuntu 16 os nomes das interfaces de rede mudaram e são específicas de cada sistema, é necessário descobrir a sua e fazer a alteração (no nome da interface aparece utilizando o comando *ifconfig* no terminal).

No processo de instalação foi criado uma regra simples para teste, o Snort gera um alerta para quando ocorresse um ping ICMP (o tutorial contém a regra). Adiciono-a ao:

/etc/snort/rules/local.rules

É importante verificar o arquivo de configuração todas as vezes que for feito uma alteração, tal verificação ocorre pelo comando de validação das configurações, visto anteriormente (1).

Com a regra instalada e as configurações concluídas podemos verificar o Snort monitorando o tráfego através do seguinte comando:

sudo /usr/local/bin/snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0

Na execução desse comando a flag -A console imprime alertas de modo rápido, a a flag -q executa no modo silencioso, sem banner e relatório de status, a flag -u <nomeusuario> executa com o usuário snort, -g <nomegrupo> com grupo snort, a flag -c <path> onde vai encontrar o arquivo de configuração e por último a flag -i <interface> informa qual interface vai escutar. Executar esse comando sozinho não gera nenhum saída, pois o Snort ficará executando, monitorando os pacotes com a regra estabelecida, para aparecer alguma saída é necessário de outro computador fazer um ping para o IP do computador que está na mesma rede do Snort.

Após a verificação do funcionamento, foram instalados o Barnyard2, o PulledPork e o visualizador web Base.

Como explicado anteriormente o Barnyard2 irá armazenar os eventos gerados pelo Snort num banco de dados, no nosso exemplo utilizamos o MySQL, permitindo assim que sejam analisados posteriormente. O Snort produz eventos em forma binária numa pasta (nessa instalação, */var/log/snort*) em seguida o Barnyard2 lê esses eventos de forma assíncrona e os insere no banco de dados criado.

O comando utilizado para realizar o download do Barnyard2 permite que seja feito a versão atual e não uma versão especifica.

Seguindo os passos do tutorial o Barnyard2 foi instalado, durante o processo criou-se um banco de dados e informou no arquivo de configuração do Barnyard as informações necessárias para utilização do banco pelo Barnyard, como usuário e senhas, por isso é importante que tal conhecimento não possa ser acessado por todos.

Para testar se o Snort está escrevendo os eventos no arquivo de log binário de forma correta e que o Barnyard2 está lendo esses logs e escrevendo no banco de dados é importante executar os dois programas no modo daemon e gerar alguns eventos de ping

na interface, porém devem ser executados um de cada vez, primeiro executa o Snort para depois o Barnyard2, em terminais distintos.

Comando Snort:

sudo /usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i eth0

Comando Barnyard2:

sudo barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -w /var/log/snort/barnyard2.waldo -g snort -u snort

Importante salientar que caso utilize o recurso de colar copiar dos comandos do tutorial as quebras precisam ser retiradas, porque a ferramenta não reconhece, foi observado durante o desenvolvimento algumas situações que ocorreram esse problema.

Quando o Barnyard2 começar a funcionar, às vezes demora um pouco, serão gerados alertas no /var/log/snort/snort.u2.nnnnnnnnn, os arquivos gerados tem essa estrutura onde n dependerá no instante em que ocorre o registro.

Utilizando Crtl+c os processos serão interrompidos.

Para auxiliar a instalar e atualizar regras no Snort é instalado o PulledPork, caso não queira instalar, pode ser feito a instalação manualmente das regras, no tutorial que foi usado de guia existe um apêndice que auxilia, Installing Snort Rules. Existem algumas regras gratuitas que podem ser baixadas, porém caso tenha interesse baixar as regras e a documentação regular é necessário criar uma conta gratuita em http://snort.org, obtendo o Oinkcode exclusivo que permite baixar as regras mais recentes com sua documentação.

No atual projeto não foram baixadas as regras mais recentes.

Na visualização do monitoramento foi utilizado o último software de suporte o BASE, que é um Guia Web que permite a visualização do tráfego. A atenção que tem que ser tomada é a utilização de uma PPA no Ubuntu, pois o Base requer PHP 5, que não tem no Ubuntu 16, que foi utilizado no projeto.

O Base foi configurado para funcionar com Apache. Após todo processo de configuração coloca-se no browser http://ServerIP/base/index.php onde ServerIP é o endereço de IP do servidor que será monitorado.

Com o Snort funcionando com detector do processo de varredura de porta,s que é o método utilizado nos primeiros passos de um ataque.

O Snort funcionando no módulo sfPortscan, foi utilizado o Nmap, uma das ferramentas de varredura de portas mais comum. O sfPortscan foi projetado para ser capaz de detectar os diferentes tipos de digitalizações que o Nmap pode produzir.

Uma regra nova foi adicionada, scan.rules:

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Nmap scan FYN";flow:stateless;flags:F,12;reference:arachnids,27;classtype: attempted-recon;gid:1000020;sid:1000020;rev:4;)

A regra não pode ter espaço entre os comandos, e a mensagem “Nmap scan FYN” só tem que ser escrita dessa forma, a diferença é devido ao editor do relatório.

Regra essa criada para permitir que o Snort perceba o varredura das portas realizada pelo Nmap. Também foi adicionado no arquivo de configuração do Snort

/etc/snort/snort.conf

o novo caminha da nova regra, abaixo da linha 546. Uma outra solução seria no arquivo de local.rules acrescentar a nova regra.

O processo ocorreu com um servidor ligado na mesma rede do Snort e uma máquina externa atacando o servidor com o Nmap, utilizando o comando:

nmap -v - sF IPServidor

O Base registrou o ataque

5. Kismet

O Kismet é NIDS e um detector de redes wireless, capaz de analisar a rede. Ele trabalha com redes Wi-Fi (IEEE 802.11), mas também pode trabalhar com outros tipos de rede, através de plug-ins. Kismet é compatível com os sistemas operacionais Linux, Mac OS X, FreeBSD, NetBSD e OpenBSD. Há um cliente para Windows mas é necessário utilizar um servidor externo.

O Kismet pode ser usado tanto para checar a segurança da rede wireless quanto para checar a presença de outras redes próximas, o que possibilita descobrir os canais que estão mais congestionados ou, até mesmo, invadir redes. Essa ferramenta identifica redes coletando passivamente os pacotes, o que possibilita a detecção de redes ocultas. Ela também é capaz de descobrir pontos de acessos que não divulgam o ESSID (Extended Service Set Identifier), quando sua placa wireless está em modo de monitoramento, além disso, é possível integrá-la a um GPS.

Algumas das Informações que o kismet gera sobre uma determinada rede wireless:

- Name - Nome da rede detectada
- Manuf - Modelo do roteador
- Encryption - Tipo de criptografía usado
- BSSID - Identificação do MAC do roteador
- Channel - Canal de frequência de operação da rede
- Packets - Numero de pacotes capturados

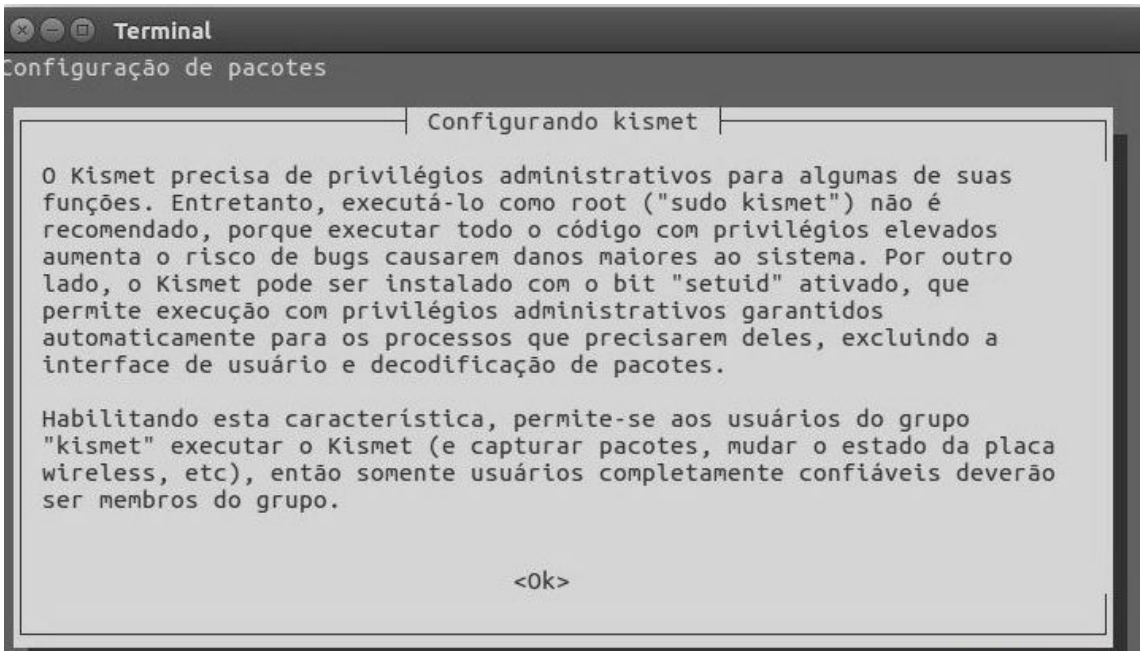
5.1. Instalação e configuração do Kismet

O processo de instalação foi feito na Raspberry Pi 2, descrita na seção 3, e é bem simples.

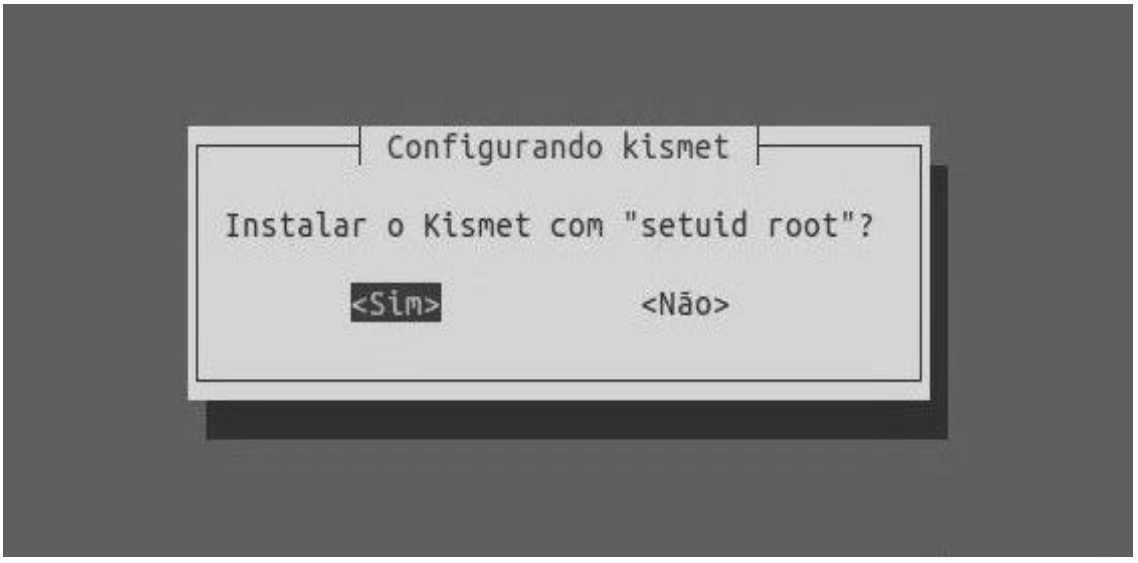
O comando para instalação:

 # sudo apt-get install kismet

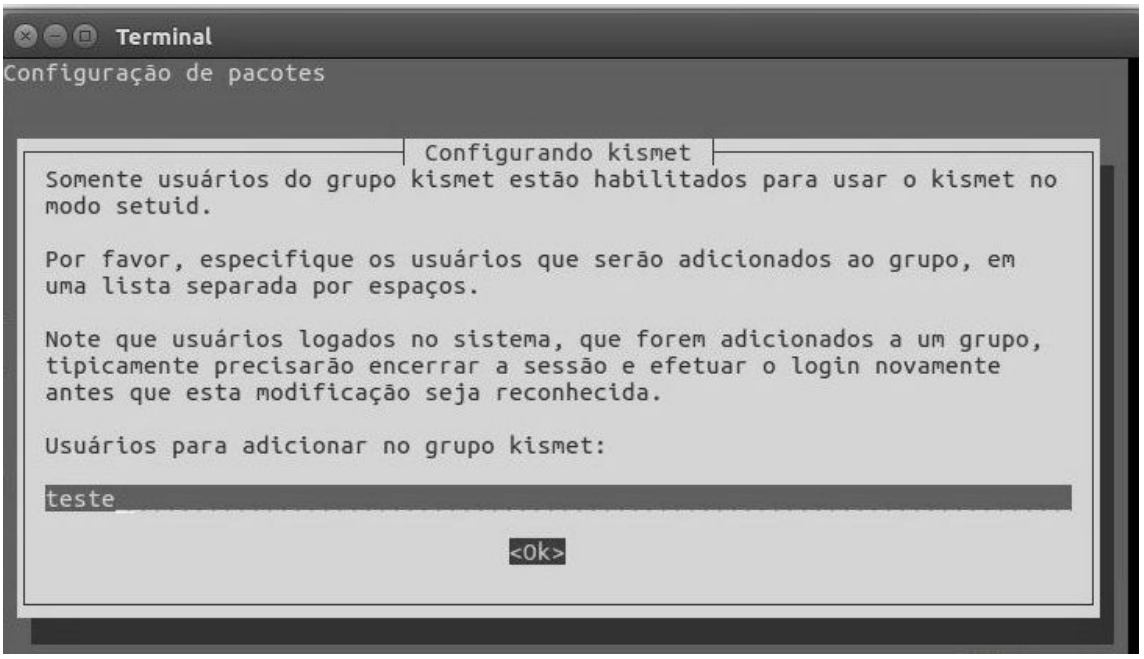
Após o comando anterior foi exibido um tela de configuração



Na imagem anterior é descrito que o kismet precisa de privilégios administrativos para algumas de sua funções. Então foi teclado ‘enter’ para confirmar.



Confirmou-se teclando ‘Enter’ para permitir a execução com privilégios administrativos garantidos automaticamente para os processos que precisarem deles.



O kismet solicitou um nome de usuário do sistema para adicionar ao grupo kismet, então foi digitado o usuário ‘teste’ e em seguida confirmou-se teclando ‘enter’ e a instalação e configuração foi finalizada .

6. Referências

[1] Dietrich, Noah. Snort 2.9.9.x on Ubuntu 14 and 16 with Barnyard2, PulledPork, and BASE. Disponível em: https://s3.amazonaws.com/snort-org-site/production/document_files/files/000/000/122/original/Snort_2.9.9.x_on_Ubuntu_14-16.pdf?AWSAccessKeyId=AKIAIXACIED2SPMSC7GA&Expires=1511966097&Signature=Lkd7sYsPQluGcF59C1%2F10xtv6qo%3D. Acesso em: 6 novembro 2017.

Barnyard2. Disponível em: http://www.forensicswiki.org/wiki/Barnyard2. Acesso em: 20 novembro 2017.

Pulledpork. Disponível em: https://www.aldeid.com/wiki/Pulledpork. Acesso em 20 novembro 2017.

KERSHAW, Mike. Documentation Kismet 2016-01-R1. Disponível em: https://www.kismetwireless.net/documentation.shtml. Acesso em: 28 novembro 2017.

Wifi - Utilizando kismet e aircrack. Disponível em: http://www.peotta.com/wiki/index.php?title=Wifi_-_Utilizando_kismet_e_aircrack#Capturando_pacotes. Acesso em: 9 novembro 2017.

Oinkcodes. Disponível em: https://www.snort.org/oinkcodes. Acesso em: 21 novembro 2017.

The Snort Project. SNORT Users Manual 2.9.11. Disponível em: https://s3.amazonaws.com/snort-org-site/production/document_files/files/000/000/129/original/snort_manual.pdf?AWSAccessKeyId=AKIAIXACIED2SPMSC7GA&Expires=1511967401&Signature=SGdaqxmZWNMT0Yya8qGNqLnkmBY%3D. Acesso em: 18 novembro 2017