Universidade do Minho
Mestrado Integrado em Engenharia Informática
Tecnologia de Segurança
TP 2 - Instalação e configuração de ambiente de *Pentest*

Objectivo

O principal objectivo deste trabalho é a instalação e configuração de um ambiente laboratorial para as atividades de *Penetration Testing* do trabalho prático 2. É importante que todos os aspectos abordados neste documento sejam cuidadosamente considerados, sob risco de exposição da rede local e de seus terminais a ataques externos.

1 - O sistema operativo Kali Linux[1]

O Kali Linux[2] é um sistema operativo de código aberto mantido pela *Offensive Security*, organização que disponibiliza treinamentos e serviços na área de segurança da informação e testes de invasão (*pentest*). O Kali Linux é um sistema voltado para utilizadores da área de segurança da informação e contém utilitários para diversas finalidades, como: *pentest*, auditoria, computação forense e redes *WiFi*. O sistema é bastante flexível, escalável e simples de usar, principalmente por ser derivado do Debian[3] Linux.

Verifique se possui o Sistema Operativo Kali instalado no seu computador. Caso isso não aconteça descarregue-o e instale a partir de: https://www.kali.org/ (ou a partir de uma imagem para VirtualBox ou VMWare: https://www.kali.org/downloads/). O uso de uma máquina virtual é a opção mais recomendada[4].

Após a instalação, verifique se o Kali Linux possui a última versão do ***nmap*** instalado (*i.e.,* versão 7.80). Caso não possua, siga as instruções do link https://nmap.org/book/install.html

Além do *nmap*, será necessário também instalar (caso ainda não o tenha) um analisador de tráfego (*sniffer*). As alternativas mais populares são o ***Wireshark*** (https://www.wireshark.org/) e o ***TCPDUMP*** (http://www.tcpdump.org/). Aqui também poderão optar por qualquer uma das alternativas, embora a primeira opção forneça um maior número de funcionalidades do interesse para as tarefas do TP 2.

2 - Sistemas de Pentest

---

[1] Uma alternativa ao Kali Linux é o projeto ParrotSec (https://www.parrotsec.org/). Para as atividades seguintes do TP2, qualquer uma das opções é adequada e, portanto, aceite.

[2] https://www.kali.org/

[3] https://www.debian.org/

[4] O procedimento para o ParrotSec é semelhante.

Sistemas de *pentest* são ferramentas que analisam *hosts* de rede em busca de vulnerabilidades presentes em sua base de conhecimento (*Knowledge Base – KB*).

Estes sistemas possuem pelo menos dois módulos: *Scan* e *Report*. O módulo de *Scan* efetua varreduras para descobrir se o *host* está ativo, e após a confirmação procura descobrir o sistema operativo e serviços em execução no *host*. Após descobrir o sistema operativo e os serviços, o sistema de *pentest* analisa sua base de conhecimento em busca de vulnerabilidades conhecidas para efetuar testes nos *hosts* que fazem parte do *Scan*. Após analisar todos os *hosts* detetados, o sistema, através do módulo *Report,* gera relatórios detalhados contendo todos os dispositivos em que foram encontradas vulnerabilidades. Este relatório contém diversas informações, como por exemplo o CVE, criticidade das vulnerabilidades encontradas, entre outras informações disponibilizadas pela KB do sistema.

O ***Nessus*** é uma ferramenta de análise de vulnerabilidades, atualmente mantida pela empresa *Tenable Network Security*. Apesar de originalmente ser uma ferramenta *open source*, hoje a sua licença permite o uso gratuito apenas residencial e para fins didáticos. O uso comercial necessita da aquisição de uma licença especifica. Por conta dessas mudanças, foi criado um novo produto, a partir da última versão livre do Nessus, atualmente conhecido como ***OpenVAS***. Para as atividade do TP 2 - Parte B, poderão também pelo uso do Nessus ou OpenVAS.

Abaixo, instruções sobre a instalação e configuração de ambas as ferramentas.

2.1 - Nessus

Instruções de instalação e configuração podem ser encontradas no link http://www.openvas.org/

Para instalar o Nessus é necessário baixar o pacote específico para o Linux, uma vez que a versão mais recente ainda não se encontra disponível nos repositórios para instalação com *apt-get*. O download do Nessus pode ser feito no seguinte endereço: https://www.tenable.com/products/nessus-home.

A instalação segue os seguintes passos:

```
#Efetuar o download do ficheiro .dpkg (ou .deb) através do link disponibilizado acima

#instalação através do dpkg do Kali:

dpkg –i nome_do_ficheiro.dpkg

# iniciar o nesses

/etc/init.d/nessusd start
```

Um guião de instalação do Nessus no Kali pode ser encontrado em.:

```
https://www.tenable.com/blog/getting-started-with-nessus-on-kali-linux
```

Após a instalação do Nessus, o mesmo deve ser configurado através de sua interface web (https://localhost:8834). Após adicionar a licença e criar o usuário o

Nessus fará o download da sua base de conhecimento, que permitirá a varredura da rede em busca de vulnerabilidades.

2.2 - OpenVAS

Instruções de instalação e configuração pode ser encontradas no link http://openvas.org/

A partir do Kali Linux é possível instalar e configurar o OpenVAS de forma simples, seguindo os passos listados abaixo:

```
#apt-get update
#apt-get disc-upgrade
#apt-get install openvas
```

Para a configuração:

```
#openvas-setup
```

Este processo levará algum tempo, já que será atualizada a base de conhecimento para o posterior scan. Ao final, será gerado um link para acesso local e uma password, que deverá ser guardada para futuros acessos à ferramenta. Sem ela, será necessário reinstalar o sistema.

Ao aceder o link fornecido, será necessário incluir uma exceção de segurança no browser, uma vez que o respectivo certificado não é válido.

Para iniciar o OpenVAS via terminal:

```
#openvas-start
```

3 - Sistemas de Deteção de Intrusão (IDS) - Snort

Sistema capaz de detetar atividade maliciosa através do monitoramento constante da rede ou de chamadas de sistema em um sistema operativo. Podem ser classificados de acordo com as seguintes características:

- Centralizados x Distribuídos
- Quanto ao modo de funcionamento:
    - Detetores de anomalias.
    - Detetores baseados em regras.
- Quanto ao local de atuação:
    - Baseados em host (HIDS).
    - Baseados em redes (NIDS).
- Quanto à forma de atuação:

- Reativos.
- Passivos.
- Ativos (IPS).

Detetor de anomalias: utiliza funções estatísticas ou rede neuronal para definir um perfil de utilização da rede. Em seguida, analisa constantemente o perfil atual da rede com o perfil aprendido. Caso ocorra alguma variação acima de um limiar, considera que houve uma tentativa de intrusão. Possibilita a deteção de ataques desconhecidos, mas pode gerar falsos positivos.

Detetor baseado em regras: possui funcionamento similar com o de um antivírus. Através de um conjunto de assinaturas, o IDS monitora o ambiente em busca de eventos que coincidam com alguma assinatura. Possui baixo índice de falsos positivos. Não detetam ataques desconhecidos e dependem de atualização das assinaturas por parte do fabricante ou da comunidade.

Detetor baseado em host: atua em cima de uma única maquina. Instalado na própria maquina que se deseja proteger. Monitora chamadas do sistema operativo ou atividades de uma aplicação específica. Também chamados de Host-based IDS (HIDS).

Detetor baseado em redes: atua em cima de um segmento de rede. É instalado em um servidor que faz parte do segmento de rede. Monitora o tráfego no segmento de rede do qual a interface de monitoramento faz parte. IDS baseado em rede é também conhecido como NIDS (Network Intrusion Detection System). Um NIDS é capaz de detetar atividade suspeita em uma rede inteira.

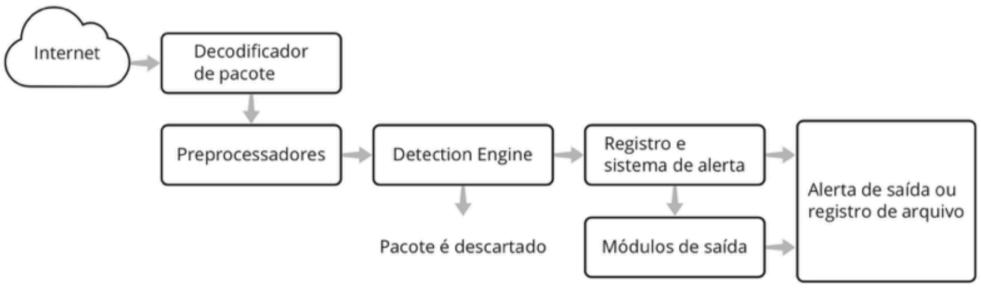
Sistemas Reativos agem após detetar um evento malicioso. Podem inserir regras em um firewall acoplado. Em alguns ataques, a reação pode ser tardia demais. Existem ataques em que um pacote é suficiente para causar algum tipo de estrago, como em ataques de negação de serviço (DoS – *Denial-of-Service*). Nesses casos, no momento em que o IDS reagir ao ataque, será tarde demais.

Sistemas Passivos: efetuam apenas registros dos eventos e geram alertas para os administradores. Uma vantagem de um IDS passivo é que ele não causa nenhuma interrupção na rede caso falhe.

Sistemas Ativos: agem ativamente em caso de evento malicioso. Os sistemas ativos são chamados de Sistemas de Prevenção de Intrusos ou IPS.

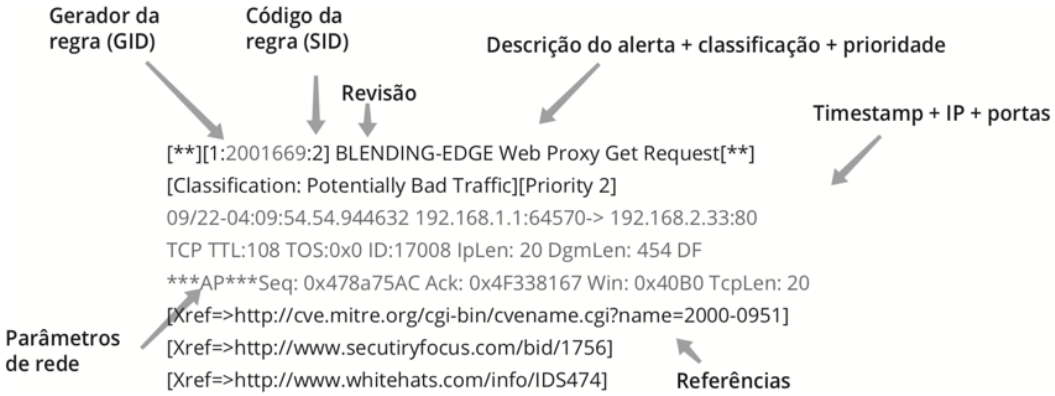
O Snort é um NIDS open source, baseado em assinaturas. A base de dados de assinaturas deve ser constantemente atualizada para continuar sendo efetivo e detetar novos ataques.

O Snort possui uma estrutura modular altamente customizável, de modo que diversos plugins e utilitários podem ser usados para expandir suas funcionalidades, como a possibilidade de reagir a um alerta, a atualização automática das suas assinaturas e o gerenciamento de diversos sensores espalhados em uma ou mais redes. Por ter o código-fonte aberto, o Snort foi portado para plataformas como Linux e Windows. A Figura abaixo demonstra o funcionamento do Snort.



O decodificador de pacote é responsável pela obtenção dos pacotes no segmento de rede monitorado. Os pré processadores realizam diversos tipos de processamento em cima dos pacotes, com o objetivo de obter tráfego normalizado. Questões como fragmentação, uso de codificações diferentes e ofuscação de pacotes são tratadas nessa etapa. A seguir, o *detection engine* é responsável por compilar as regras (assinaturas) e testar os pacotes contra essas regras. O registro e sistema de alerta gera os registros do Snort e envia os alertas.

Por fim, os módulos de saída exportam os alertas e registros para um arquivo ou banco de dados. A Figura a seguir apresenta um exemplo de alerta gerado pelo Snort.



3.1 - Instalação e configuração do SNORT

Para instalar o Snort no Kali (ou alguma outra distribuição do Linux baseada em Debian) é possível utilizar o *apt*.

```
~#apt-get update

~#apt-get install snort

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following additional packages will be installed:
```

```
   libdaq2 oinkmaster snort-common snort-common-libraries snort-
rules-default

Suggested packages:

  snort-doc

The following NEW packages will be installed:

  libdaq2 oinkmaster snort snort-common snort-common-libraries

  snort-rules-default

0 upgraded, 6 newly installed, 0 to remove and 675 not upgraded.

Need to get 2230 kB of archives.

After this operation, 7325 kB of additional disk space will be
used.

Do you want to continue? [Y/n]
```

Ao final da execução do comando, o Snort estará instalado. Para iniciar o Snort precisamos executar: “/etc/init.d/snort start”.

Para testar se o mesmo está executando podemos executar o comando: “ps aux | grep snort”, ou ainda “/etc/init.d/snort status”.

```
snort       2772   0.0   7.4  595636  152012 ?          Ssl   09:24
0:00 /usr/sbin/snort -m 027 -D -d -l /var/log/snort -u snort -g
snort -c /etc/snort/snort.conf -S HOME_NET=[0.0.0.0/0] -i eth0

root      2972  0.0  0.0  12860   936 pts/0    S+   10:22   0:00
grep snort
```

A saída do comando grep demonstra que o Snort está a executar, assim como os parâmetros utilizado na sua execução, que significam:

-D: modo daemon, executa o Snort como um serviço.

-d: instrui o Snort a incluir os dados da camada de aplicação no pacote que será registrado.

-l: indica o diretório onde os logs do Snort serão armazenados. Neste exemplo, o diretório /var/log/snort conterá os registros de alertas e pacotes.

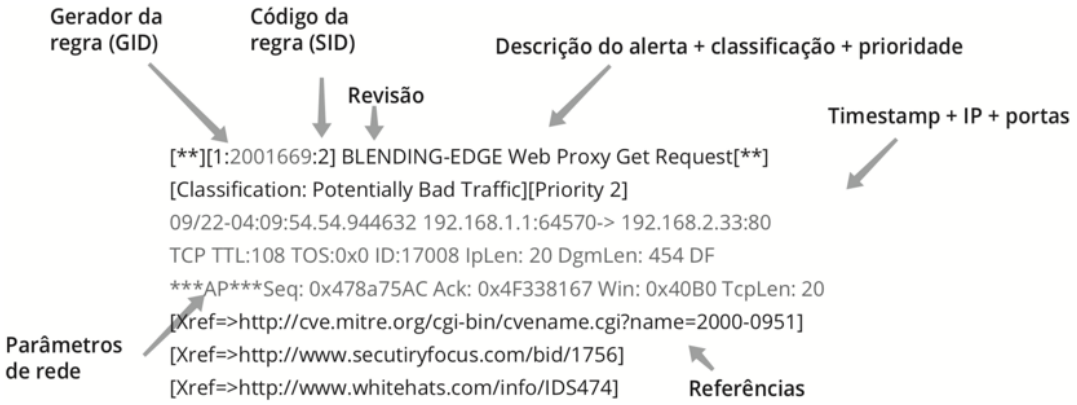
-u: indica o usuário que será utilizado para executar o Snort.

-g: indica o grupo utilizado para executar o processo do Snort.

-c: indica o caminho do arquivo de configuração.

-S: variável=valor ajusta a variável para o valor definido. Permite alteração em linha de comando de parâmetros do ficheiro de configuração. No exemplo acima, o parâmetro está ajustando a variável HOME_NET para o valor 0.0.0.0/0, definido durante a instalação.

-



i:
indica a interface que será utilizada para a captura de trafego.

Configuração do Snort:

`Ficheiro /etc/snort/snort.conf`:

Parâmetros importantes:

- `var RULE_PATH`
  - `Indica o caminho onde os ficheiros de regras (assinaturas) se encontram.`
- `include $RULE_PATH/<arquivo>.rules`
  - `Inclui um arquivo de regras.`
- `var HOME_NET [192.168.x.0/24]`
- `var SMTP_SERVERS 172.16.1.20 2`
- `var EXTERNAL_NET any`

Na Secção 6 de plugins de saída é recomendável adicionar a seguinte linha de configuração:

```
output alert_full: alert.full
```

Isto se faz necessário para a geração de logs adicionais, para facilitar o entendimento do alerta gerado. A Figura abaixo exibe um exemplo do log gerado por um alerta.

4 - Metasploitable 2

A *metasploitable 2* (ou a versão mais recente, i.e., metasploitable 3) é uma instalação do sistema operacional Linux intencionalmente vulnerável para fins de formação/treino. Poderão baixar uma versão com extensão *ova* (compatível com VMWare e VirtualBox) através do link https://goo.gl/4AAg5B

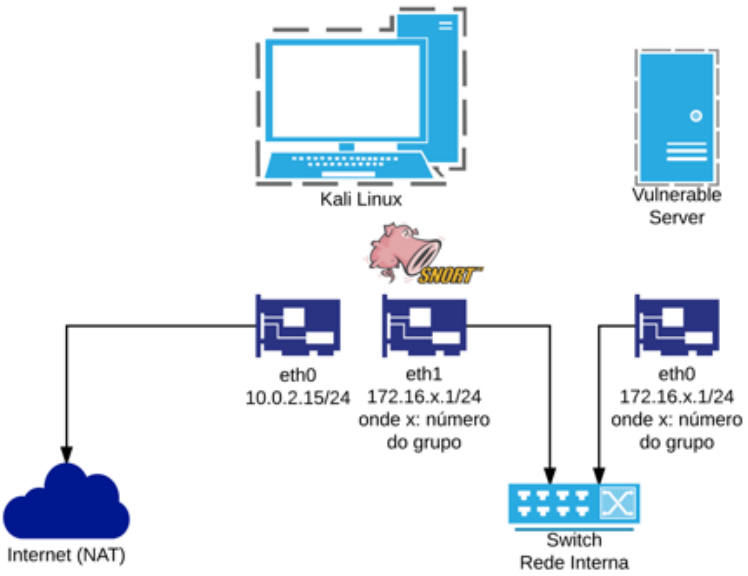
Credenciais - Login/Password: ***msfadmin***

Para o processo de instalação e uso, siga cuidadosamente as instruções da Secção 5. kill

Metasploitable 3 Link com o OVA - https://github.com/brimstone/metasploitable3/releases/tag/0.1.4

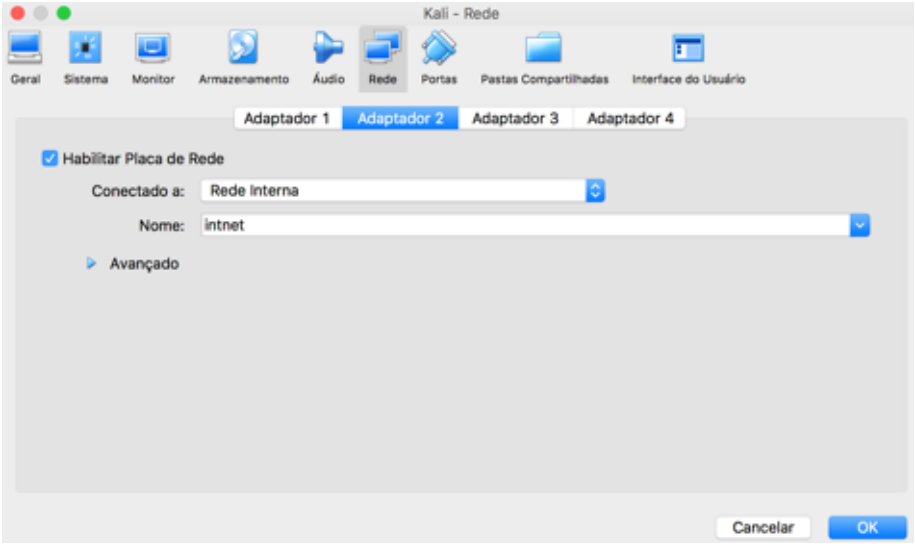
5 - Configuração do ambiente de testes

Com o objetivo de manter um ambiente de Penetration Test isolado da rede local e, portando, evitando riscos de ataques externos, deverá ser configurada uma rede virtual através do VMWare ou do VirtualBox. As imagens de configuração abaixo referem-se à segunda opção, porém, para a primeira, os passos são semelhantes. A topologia de rede para o ambiente de testes deve seguir a seguinte arquitetura:



Note que para aceder as configurações a máquina virtual deve estar desligada. Aceder as configurações da máquina virtual e modificar as opções necessárias. Para

isolar o ambiente de testes da rede local, ligue ambas as máquinas virtuais (i.e., Kali Linux e Metasploitable 2) via um configuração de “rede interna”.



Após a configuração do VirtualBox, o Kali Linux e a VM Metasploitable2 deverão ser configuradas para atribuírem os ip’s: **172.16.x.1/24 (Kali)** e **172.16.x.2/24 (Metasploitable)**, onde **x = número do grupo**.

Efetue testes para verificar que toda o ambiente de testes é funcional.