Universidade do Minho
Mestrado Integrado em Engenharia Informática
Segurança de Sistemas Informáticos
Trabalho Prático 2

1 *- Instruções*

- O trabalho prático deverá ser feito em dupla;
- A submissão deverá ser feita por apenas um dos integrantes da dupla, exclusivamente, via *blackboard*;
- Para todas as questões propostas, inclua imagens que demonstrem a forma como chegou aos resultados obtidos;
- A entrega consiste em um ficheiro pdf que inclua toda a análise proposta neste enunciado;
- O prazo de submissão será 23h59 do dia **22/12/2020**

2 - *Objetivo*

Este trabalho prático é dividido em duas partes independentes, a primeira parte consiste no uso de técnicas para coleta passiva de informação como ferramenta de análise da postura de segurança em sistemas e infra-estruturas reais. Na segunda parte, será configurado um ambiente de testes no qual técnicas e ferramentas de varredura activa (i.e., scanning) serão usadas como estratégia de identificação de vulnerabilidades e fraquezas de um sistema remoto.

As duas partes deste trabalho estão enquadradas na fase de *footprinting* da atividade de testes de penetração detalhadas nos *slides* da Aula 03.

3 - *Parte A*

Escolha duas empresas com serviços *on-line* (uma grande corporação e um negócio local) e utilize técnicas de busca passiva de informação que permitam identificar detalhes sobre os seus sistemas e infra-estrutura. Descreva as estratégias usadas, os resultados obtidos e as possíveis diferenças de postura adoptadas pelos administradores dos domínios estudados. Enriqueça a sua análise apontando estratégias destinadas a fortalecer a postura de segurança destes domínios, especificamente, como resposta a técnicas e ferramentas de busca passiva.

4 - *Parte B*

Para esta parte do trabalho prático, certifique-se que tem o ambiente de testes instalado e configurado de acordo com as instruções da Sec. 5 deste enunciado. Todas as tarefas listadas nesta parte do trabalho deverão usar ferramentas de varredura activa instaladas no *Sistema Auditor* e terá como alvo, **apenas**, o *Sistema Mestasploitable 3*.

Com base neste ambiente, forneça respostas detalhadas para as questões abaixo.

Q1: Selecione um conjunto de ferramentas e técnicas de varredura activa para identificar e detalhar vulnerabilidades e fraquezas para as quais o *Sistema Metasploitable 3* está exposto.
A sua resposta deverá listar os serviços a correr neste sistema e as vulnerabilidades e/ou fraquezas relacionados a cada um. Para os serviços com diferentes vulnerabilidades, escolha a mais recente ou a mais grave.
Importante: Para esta questão, **não será permitido o uso de Scanners de Vulnerabilidades** (por exemplo, OpenVAS ou Nessus). Uma lista abrangente de ferramentas pode ser consultada em www.sectools.org

****

Para as questões Q2 a Q5, certifique-se que tem o *Scanner de vulnerabilidade* (e.g., OpenVAS ou Nessus), o Sistema de Deteção de Intrusão - IDS (e.g., Snort ou Suricata) e um analisador de tráfego (e.g., Wireshark) de sua escolha corretamente configurados no *Sistema Auditor*. Efetue uma varredura ao *Sistema Metasploitable 3* tendo, durante o processo, o IDS ativo e o analisador de tráfego a capturar todos os pacotes trocados entre os dois sistemas.
Use o resultado da varredura, os alertas gerados pelo IDS e o tráfego capturado para responder as seguintes questões.
****

Q2: Discuta os resultados globais do processo de varredura activa ao *Sistema Mestasploitable 3*. Avalie também as diferenças entre o resultado do sistema automático de identificação de vulnerabilidades e o resultado que obteve no item Q1 da Parte B deste enunciado.

Q3: Examine o *output* do IDS e escolha dois eventos identificados como tráfego anómalo. Para cada evento escolhido, identifique o respetivo tráfego capturado via *Analisador de tráfego* e o descreva. Se possível, inclua o CVE da vulnerabilidade e o método de identificação usado pelo *scanner*.

Q4: Observe que algumas notificações do IDS não possuem vulnerabilidade correspondente no relatório do *Scanner de vulnerabilidades*. Apresente e discuta as possíveis razões para estas diferenças.

Q5: Escolha três vulnerabilidades identificadas pelo *Scanner de vulnerabilidades,* sendo, pelo menos, uma classificada como *High*/*Critical* e uma classificada como *Medium*. Pesquise a documentação referente às formas de corrigir a fonte do problema e efetue os procedimentos necessários para tal. Ao final dos procedimentos escolhidos para cada vulnerabilidade, execute uma nova varredura para garantir que estas já não são identificadas.
Discuta a solução dada e inclua os ficheiros resultantes da varredura antes e depois das respectivas correções.

5 - *Descrição do ambiente de testes*

O ambiente de testes será composto por dois sistemas principais:

i) *Sistema alvo* (*i.e, Sistema Metasploitable*): corresponde a uma instância do projeto *Metasploitable 3* mantido pela *Rapid7.* Há duas formas de configurar este sistema em

uma máquina virtual, a primeira delas está descrita no link do projeto (https://github.com/rapid7/metasploitable3). A segunda consiste no uso de uma imagem pré-configurada que pode ser importada via *VirtualBox* ou *VMWare*. Uma imagem deste tipo pode ser encontrada no link (https://drive.google.com/u/0/uc?export=download&confirm=c8XR&id=1-cDEpDRl5_-QWBU8Ckpp_Zep-1-9-EY4), contudo, é importante notar que esta pode não ser a versão mais recente do sistema e tende a exigir mais espaço de armazenamento disponível.[1] A solução adoptada será da escolha do grupo.

ii) *Sistema Auditor*: corresponde ao sistema usado para executar varreduras activas ao *Sistema Metasploitable 3*. Aqui, há três alternativas principais que o grupo pode adoptar: *i)* instalar individualmente todas as ferramentas necessárias no sistema operativo nativo ou em uma máquina virtual previamente criada; *ii)* instalar a *framework* Kali no sistema operativo nativo ou em uma máquina virtual previamente criada; ou *iii)* usar uma máquina virtual Kali Linux previamente configurada (https://www.kali.org/). Para qualquer alternativa, será necessário instalar e configurar um IDS, um Scanner de Vulnerabilidades e um analisador de tráfego (já instalado na VM Kali Linux).

As instruções seguintes correspondem a um pequeno tutorial de como instalar e configurar um ambiente referência que consiste no *Sistema Metasploitable 3* e *Sistema Auditor* a correr em máquinas virtuais. Para as demais alternativas, consulte a documentação correspondente.

Scanner de Vulnerabilidade - ***Nessus***

O *Nessus* é uma ferramenta para identificação automática 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* [2].

Para instalar o Nessus é necessário baixar o pacote específico para o Linux, uma vez que a versão mais recente ainda não está 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 Nessus
```

[1] Isso não impede a realização do trabalho.

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

```
/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
```

Uma vez instalado, o *Nessus* deverá ser configurado através de sua interface web (https://localhost:8834). Após adicionar a licença e criar um utilizador o Nessus fará o download da sua base de conhecimento, que permitirá a varredura ao *Sistema Metasploitable* em busca de vulnerabilidades.

Sistema de Deteção de Intrusão (IDS) - ***Snort***

Sistema capaz de detetar atividade maliciosa através da monitorização constante da rede ou de chamadas de sistema em um sistema operativo. Este tipo de ferramenta pode ser classificada 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 rede (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. Esta abordagem 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 sistema em busca de eventos que coincidam com alguma assinatura. Apesar de produzir um baixo índice de falsos positivos, esta classe não deteta 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 sobre um segmento de rede através de um dispositivo ligado diretamente a este segmento. Desta forma, monitora o tráfego no segmento de rede ao qual a interface de monitorização está ligada. Um IDS baseado em rede é também conhecido como NIDS (*Network Intrusion Detection System*). A depender de sua configuração, um NIDS será capaz de detetar atividade suspeita em uma rede inteira.

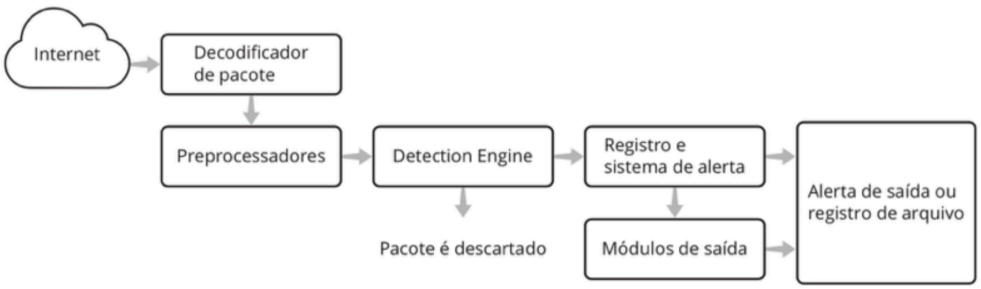
Sistemas Reativos: agem após detetar um evento malicioso, por exemplo, inserindo regras em um firewall ligado ao mesmo segmento de rede. Em alguns ataques, contudo, a reação pode ser tardia.

Sistemas Passivos: efetuam apenas registos dos eventos e geram alertas para os administradores ou outros sistemas. 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 (IPS).

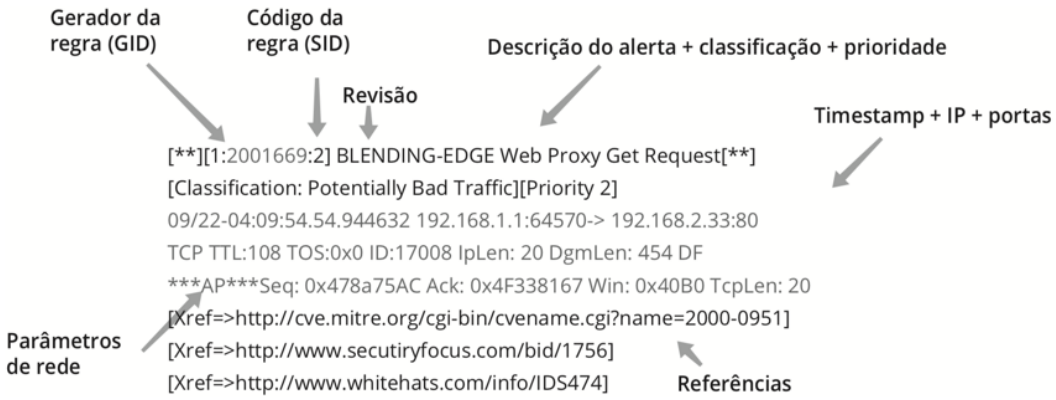
O Snort é um NIDS *open source*, baseado em assinaturas. A base de dados de assinaturas deve ser constantemente atualizada para que continue efetivo na deteção de novos ataques.

Este sistema 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. A Figura abaixo ilustra o funcionamento base 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 sobre estes 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 analisas cada pacote contra essas regras. O registo e sistema de alerta geram os registos do Snort e envia os respetivos alertas.

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



Instalação

Para instalar o Snort no Kali (ou alguma outra distribuição do Linux baseada em Debian) é possível utilizar o *apt*. Para isso, siga as seguintes instruções:

```
~#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 serviço, execute: “/etc/init.d/snort start”.

Para testar se o mesmo está a correr use 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 se 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 registado.

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

-u: indica o utilizador que executará o Snort.

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

-c: indica o caminho do ficheiro 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 [172.20.x.0/24]
- var EXTERNAL_NET any

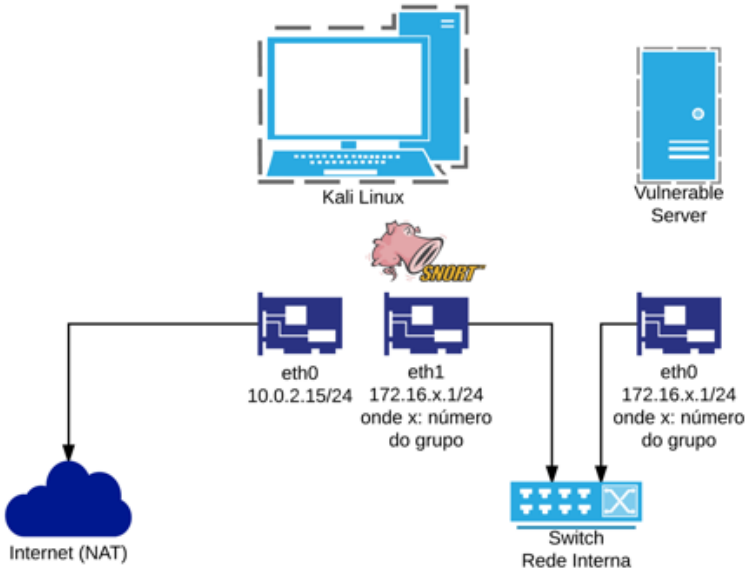
Na Secção 6 de plugins de saída é recomendável adicionar a seguinte linha de configuração:

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

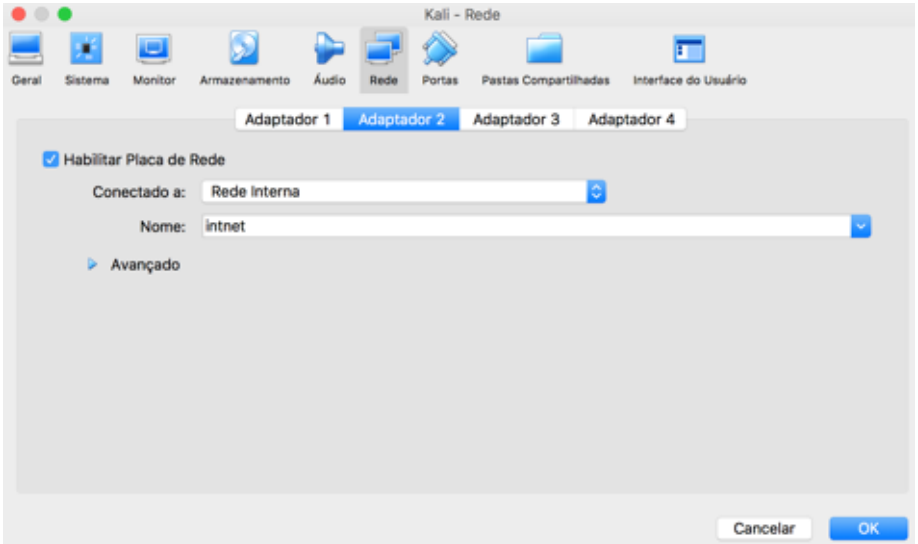
Isto se faz necessário para a geração de logs adicionais que facilitarão o entendimento do alerta gerado.

Comunicação entre as máquinas virtuais

Com o objetivo de manter o ambiente de teste 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 ao menu de configuração, a máquina virtual deve estar desligada. Para isolar o ambiente de testes da rede local, ligue ambas as máquinas virtuais (i.e., Kali Linux e



Metasploitable 3) via um configuração de “rede interna”.

Após a configuração do VirtualBox, o Kali Linux e a VM Metasploitable 3 deverão ser configuradas para atribuírem os ip’s: **172.20.x.1/24 (Kali)** e **172.20.x.2/24 (Metasploitable)**, onde **x = número do grupo** (ver em https://bit.ly/2H4fan4).

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