

Universidade do Minho  
Departamento de Informática

Mestrado Integrado em Engenharia Informática

# Segurança de Redes



---

## TP5 – Práticas sobre Detecção de Intrusões em Rede

---

Grupo 5  
A86617 Gonçalo Nogueira  
A74806 João Amorim  
A75876 Jorge Cardoso  
A78566 Marcos Silva  
A82529 Carlos Afonso

Braga  
Dezembro, 2020

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Conceitos Importantes</b>	<b>3</b>
2.1	Firewall . . . . .	3
2.2	Sistema de Detecção de Intrusões . . . . .	3
2.3	Snort . . . . .	3
2.4	ARP-poisoning . . . . .	3
<b>3</b>	<b>Tarefas</b>	<b>4</b>
3.1	Tarefa 2 - O Snort como sniffer . . . . .	4
3.1.1	Exercício 2: Execute o Snort com a opção -? e verifique a lista das opções que pode utilizar na linha de comando. Em particular anote a função das opções -v, -d,-e,-l,-c e -t . . . . .	4
3.1.2	Exercício 3 a): Execute o comando snort -vde. Anote o que observa no ecrã, procurando justificar a atividade. . . . .	5
3.1.3	Exercício 4: No computador de teste, execute o comando ping para o endereço do computador alvo. a) Anote o que observa no ecrã do computador alvo, procurando justificar a atividade. . . . .	6
3.1.4	Exercício 5: No computador alvo, interrompa o Snort. a) Anote o que observa no ecrã. . . . .	7
3.2	Tarefa 4 - Iniciar o Snort e testar a configuração . . . . .	8
3.2.1	Exercício 6: No computador alvo, interrompa a execução do Snort. Observe com atenção o relatório apresentado pelo Snort e anote, em particular: . . . .	8
3.2.2	Exercício 7: Copie para o relatório o conteúdo do ficheiro de alertas, juntamente com o conteúdo do ficheiro "snort.log.*"e comente o mesmo. . . . .	9
3.3	Tarefa 5 - Criar uma configuração que utiliza a <i>detection engine</i> e testá-la . . . . .	11
3.3.1	Exercício 6: No computador de teste, inicie uma sessão Telnet com o servidor; siga as instruções do seu servidor para o ativar, registando o processo no logbook. . . . .	11
3.3.2	Exercício 7: No computador alvo, interrompa a execução do Snort com Ctrl+C . . . . .	12
3.3.3	Exercício 8: No computador alvo, observe o conteúdo do ficheiro de alertas. Anote todos os alertas detetados e justifique-os à luz da atividade criada. Consegue localizar os alertas gerados pelo ataque ARP poisoning? Justifique. . . . .	13
3.4	Tarefa 6 - Criar uma configuração que utiliza a <i>detection engine</i> , uma regra específica e testá-la . . . . .	14
3.4.1	Exercício 2: Utilizando o seu editor preferido crie um ficheiro com o nome subseven.rules e acrescente-lhe uma linha com o texto "alert tcp anyday -> any 27374". Qual o objetivo desta regra? . . . . .	14
3.4.2	Exercício 6: No computador alvo, interrompa a execução do Snort. . . . .	14
3.4.3	Exercício 7: Observe o conteúdo do ficheiro de alertas. Deverá identificar aí um ou mais alertas. . . . .	15
<b>4</b>	<b>Conclusão</b>	<b>16</b>
<b>5</b>	<b>Bibliografia</b>	<b>17</b>

# 1 Introdução

Nos trabalhos práticos anteriores abordámos um vasto conjunto de tecnologias, na sua maioria destinadas a proteger a informação, ou a evitar que algum hacker assuma algum tipo de controlo sobre o nosso equipamento informático.

Tempo agora para uma abordagem mais cautelosa e preventiva, em torno daquilo que pode ser feito, caso sejamos alvo de um ataque bem sucedido.

Existem duas alternativas para a deteção de ataques: através da verificação regular dos logs do sistema, ou pela instalação e configuração de um Sistema de Deteção de Intrusões para redes, o chamado NIDS.

Neste trabalho prático, iremos aprofundar os nossos conhecimentos nesta segunda alternativa, com recurso a uma ferramenta de seu nome Snort (um software open-source).

## 2 Conceitos Importantes

### 2.1 Firewall

Basicamente, trata-se de um dispositivo que filtra o tráfego de redes, protegendo a rede interna de um mundo exterior não confiável. *Firewalls* e *Access Control* evitam atividade maliciosa, mas a má utilização legítima deve ser detetada, daí o surgimento de Sistemas de Detecção de Intrusões.

### 2.2 Sistema de Detecção de Intrusões

É um conjunto de componentes, de software ou de hardware, que tem a função de detetar, identificar e responder a actividades não autorizadas ou anormais num sistema alvo, isto é, detectar e contrariar intrusões.

Uma intrusão é qualquer conjunto de acções com o intuito de comprometer a integridade, a confidencialidade ou a disponibilidade de um recurso.

Neste caso, iremos abordar o tipo de IDS que é baseado em Redes, **NIDS**. Este monitora o tráfego de rede num segmento ou dispositivo, e analisa a rede e a atividade dos protocolos para identificar comportamentos suspeitos.

### 2.3 Snort

O Snort é constituído por quatro módulos:

- **Sniffer**, semelhante ao tcpdump, este módulo é responsável pela captura do tráfego que chega à placa de rede local, a funcionar no modo promíscuo;
- **Preprocessor**, executa várias funções, sendo uma delas a detecção de pacotes mal formados, como respostas anormais a pedidos ARP;
- **Detection Engine**, verifica os dados que lhe chegam contra um conjunto de regras predefinidas;
- **Alerts**, envia uma mensagem de alerta, para um ficheiro, por email, ou para o sistema de logs - conforme configurado -, sempre que é detetado um pacote suspeito, que verificou algumas das regras.

### 2.4 ARP-poisoning

É uma técnica em que o intruso envia mensagens ARP(Address Resolution Protocol) para uma rede local, com o objetivo de associar o seu endereço MAC com o endereço IP do alvo, fazendo com que qualquer tráfego destinado ao alvo, seja enviado para o responsável pelo ataque.

## 3 Tarefas

### 3.1 Tarefa 2 - O Snort como sniffer

- 3.1.1 Exercício 2: Execute o Snort com a opção -? e verifique a lista das opções que pode utilizar na linha de comando. Em particular anote a função das opções -v, -d,-e,-l,-c e -t

```
-A      Set alert mode: fast, full, console, test or none (alert file alerts only)
        "unsock" enables UNIX socket logging (experimental).
-b      Log packets in tcpdump format (much faster!)
-B <mask> Obfuscate IP addresses in alerts and packet dumps using CIDR mask
-c <rules> Use Rules file <rules>
-C      Print out payloads with character data only (no hex)
-d      Dump the Application Layer
-D      Run Snort in background (daemon) mode
-e      Display the second layer header info
-f      Turn off fflush() calls after binary log writes
-F <bpf> Read BPF filters from file <bpf>
-g <gname> Run snort gid as <gname> group (or gid) after initialization
-G <gid> Log Identifier (to uniquely id events for multiple snorts)
-h <hsh> Set home network = <hsh>
        (for use with -l or -B, does NOT change $HOME_NET in IDS mode)
-H      Make hash tables deterministic.
-l <lf> Listen on interface <lf>
-I      Add Interface name to alert output
-k <mode> Checksum mode (all,noip,notcp,noudp,noicmp,none)
-K <mode> Logging mode (pcap[default],ascii,none)
-l <ld> Log to directory <ld>
-L <file> Log to this tcpdump file
-M      Log messages to syslog (not alerts)
-m <unmask> Set unmask = <unmask>
-n <cnt> Exit after receiving <cnt> packets
-N      Turn off logging (alerts still work)
-O      Obfuscate the logged IP addresses
-p      Disable promiscuous mode sniffing
-P <snap> Set explicit snaplen of packet (default: 1514)
-q      Quiet. Don't show banner and status report
-Q      Enable inline mode operation.
-r <tf> Read and process tcpdump file <tf>
-R <ld> Include 'id' in snort_intf-id.pid file name
-s      Log alert messages to syslog
-S <env> Set rules file variable n equal to value v
-t <dir> Chroots process to <dir> after initialization
-T      Test and report on the current Snort configuration
-u <uname> Run snort uid as <uname> user (or uid) after initialization
-U      Use UTC for timestamps
-v      Be verbose
-V      Show version number
-X      Dump the raw packet data starting at the link layer
-x      Exit if Snort configuration problems occur
-y      Include year in timestamp in the alert and log files
-Z <file> Set the performonitor preprocessor file path and name
-?      Show this information
```

Figura 1: Opções Snort

Resposta:

- -v - output mais detalhado;
- -d - largar a camada de aplicação;
- -e - apresentar o cabeçalho da informação da segunda camada;
- -l - ficheiro log para a diretoria;
- -c - usar o ficheiro de regras;
- -T - testar e reportar a atual configuração do Snort.

3.1.2 Exercício 3 a): Execute o comando snort -vde. Anote o que observa no ecrã, procurando justificar a atividade.

```
WARNING: No preprocessors configured for policy 0.
12/15-23:53:41.982554 1C:AB:C0:EE:D3:86 -> 08:00:27:90:D4:82 type:0x800 len:0x42
157.240.212.16:443 -> 192.168.1.5:36462 TCP TTL:89 TOS:0x0 ID:13457 Iplen:20 Dgnlen:52 DF
***A**** Seq: 0xA0B0C82A Ack: 0xA416ADA2 Wln: 0x181 Tcplen: 32
TCP Options (3) => NOP NOP TS: 1062105724 4293755402
*****

WARNING: No preprocessors configured for policy 0.
12/15-23:53:41.980862 1C:AB:C0:EE:D3:86 -> 08:00:27:90:D4:82 type:0x800 len:0x42
157.240.212.16:443 -> 192.168.1.5:36462 TCP TTL:89 TOS:0x0 ID:13458 Iplen:20 Dgnlen:52 DF
***A**** Seq: 0xA0B0C82A Ack: 0xA416ADD8 Wln: 0x181 Tcplen: 32
TCP Options (3) => NOP NOP TS: 1062105728 4293755404
*****

WARNING: No preprocessors configured for policy 0.
12/15-23:53:42.050046 1C:AB:C0:EE:D3:86 -> 08:00:27:90:D4:82 type:0x800 len:0x5E
157.240.212.16:443 -> 192.168.1.5:36462 TCP TTL:89 TOS:0x0 ID:13459 Iplen:20 Dgnlen:80 DF
***AP*** Seq: 0xA0B0C82A Ack: 0xA416ADD8 Wln: 0x181 Tcplen: 32
TCP Options (3) => NOP NOP TS: 1062105790 4293755404
17 03 03 00 17 05 B3 20 6E D9 5F A8 BF 60 76 02 .....n...v.
54 91 A2 0F 23 1F 56 0A C4 A9 79 37 .....T...#.V...y7
*****

12/15-23:53:42.050056 08:00:27:90:D4:82 -> 1C:AB:C0:EE:D3:86 type:0x800 len:0x42
192.168.1.5:36462 -> 157.240.212.16:443 TCP TTL:64 TOS:0x0 ID:49361 Iplen:20 Dgnlen:52 DF
***A**** Seq: 0xA416ADD8 Ack: 0xA0B0C846 Wln: 0x1F5 Tcplen: 32
TCP Options (3) => NOP NOP TS: 4293755485 1062105790
*****

WARNING: No preprocessors configured for policy 0.
12/15-23:53:42.053265 1C:AB:C0:EE:D3:86 -> 08:00:27:90:D4:82 type:0x800 len:0x60
157.240.212.16:443 -> 192.168.1.5:36462 TCP TTL:89 TOS:0x0 ID:13460 Iplen:20 Dgnlen:82 DF
***AP*** Seq: 0xA0B0C846 Ack: 0xA416ADD8 Wln: 0x181 Tcplen: 32
TCP Options (3) => NOP NOP TS: 1062105795 4293755404
17 03 03 00 19 30 89 01 8C 91 97 CA 00 FB B9 27 .....0.....'
05 DB 80 C5 14 7E 32 94 EF DF 1D EE E5 B8 .....-2.....
*****

12/15-23:53:42.053273 08:00:27:90:D4:82 -> 1C:AB:C0:EE:D3:86 type:0x800 len:0x42
192.168.1.5:36462 -> 157.240.212.16:443 TCP TTL:64 TOS:0x0 ID:49362 Iplen:20 Dgnlen:52 DF
***A**** Seq: 0xA416ADD8 Ack: 0xA0B0C864 Wln: 0x1F5 Tcplen: 32
TCP Options (3) => NOP NOP TS: 4293755488 1062105795
*****
```

Figura 2: Resultado do comando snort -vde

**Resposta:**

Tráfego na Rede captado pelo Snort

- 3.1.3 Exercício 4: No computador de teste, execute o comando ping para o endereço do computador alvo. a) Anote o que observa no ecrã do computador alvo, procurando justificar a atividade.

```
+++++
WARNING: No preprocessors configured for policy 0.
12/15-23:55:10.580017 08:00:27:60:02:F5 -> 08:00:27:90:D4:82 type:0x800 len:0x62
192.168.1.8 -> 192.168.1.5 ICMP TTL:64 TOS:0x0 ID:5775 IpLen:20 DgnLen:84 DF
Type:8 Code:0 ID:56603 Seq:24 ECHO
DE 4C D9 5F 00 00 00 00 E8 39 08 00 00 00 00 00 .L_.....9.....
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F .....
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F !"#%&'()*+,-./
30 31 32 33 34 35 36 37 01234567

+++++
WARNING: No preprocessors configured for policy 0.
12/15-23:55:10.580033 08:00:27:90:D4:82 -> 08:00:27:60:02:F5 type:0x800 len:0x62
192.168.1.8 -> 192.168.1.8 ICMP TTL:64 TOS:0x0 ID:63898 IpLen:20 DgnLen:84
Type:0 Code:0 ID:56603 Seq:24 ECHO REPLY
DE 4C D9 5F 00 00 00 00 E8 39 08 00 00 00 00 00 .L_.....9.....
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F .....
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F !"#%&'()*+,-./
30 31 32 33 34 35 36 37 01234567

+++++
WARNING: No preprocessors configured for policy 0.
12/15-23:55:11.580601 08:00:27:60:02:F5 -> 08:00:27:90:D4:82 type:0x800 len:0x62
192.168.1.8 -> 192.168.1.5 ICMP TTL:64 TOS:0x0 ID:5926 IpLen:20 DgnLen:84 DF
Type:8 Code:0 ID:56603 Seq:25 ECHO
DF 4C D9 5F 00 00 00 00 43 3E 08 00 00 00 00 00 .L_.....C>.....
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F .....
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F !"#%&'()*+,-./
30 31 32 33 34 35 36 37 01234567

+++++
WARNING: No preprocessors configured for policy 0.
12/15-23:55:11.580617 08:00:27:90:D4:82 -> 08:00:27:60:02:F5 type:0x800 len:0x62
192.168.1.8 -> 192.168.1.8 ICMP TTL:64 TOS:0x0 ID:64072 IpLen:20 DgnLen:84
Type:0 Code:0 ID:56603 Seq:25 ECHO REPLY
DF 4C D9 5F 00 00 00 00 43 3E 08 00 00 00 00 00 .L_.....C>.....
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F .....
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F !"#%&'()*+,-./
30 31 32 33 34 35 36 37 01234567

+++++
```

Figura 3: Resultado do ping para o computador alvo

Resposta:

O Snort deteta o ping(ICMP), feito a partir do computador teste.

3.1.4 Exercício 5: No computador alvo, interrompa o Snort. a) Anote o que observa no ecrã.

```
=====
Run time for packet processing was 58.592011 seconds
Snort processed 458 packets.
Snort ran for 0 days 0 hours 0 minutes 58 seconds
Pkts/sec: 7
=====
Memory usage summary:
Total non-mapped bytes (arena): 786432
Bytes in mapped regions (hblkhd): 13180928
Total allocated space (uordblks): 681264
Total free space (fordblks): 105168
Topmost releasable block (keepcost): 102832
=====
Packet I/O Totals:
Received: 463
Analyzed: 458 ( 98.920%)
Dropped: 0 ( 0.000%)
Filtered: 0 ( 0.000%)
Outstanding: 5 ( 1.080%)
Injected: 0
=====
Breakdown by protocol (includes rebuilt packets):
Eth: 458 (100.000%)
VLAN: 0 ( 0.000%)
IP4: 448 ( 96.070%)
Frag: 0 ( 0.000%)
ICMP: 116 ( 25.328%)
UDP: 23 ( 5.022%)
TCP: 300 ( 65.502%)
IP6: 0 ( 0.000%)
IP6 Ext: 0 ( 0.000%)
IP6 Opts: 0 ( 0.000%)
Frag6: 0 ( 0.000%)
ICMP6: 0 ( 0.000%)
UDP6: 0 ( 0.000%)
TCP6: 0 ( 0.000%)
Teredo: 0 ( 0.000%)
ICMP-IP: 0 ( 0.000%)
IP4/IP4: 0 ( 0.000%)
IP4/IP6: 0 ( 0.000%)
IP6/IP4: 0 ( 0.000%)
IP6/IP6: 0 ( 0.000%)
GRE: 0 ( 0.000%)
GRE Eth: 0 ( 0.000%)
GRE VLAN: 0 ( 0.000%)
GRE IP4: 0 ( 0.000%)
GRE IP6: 0 ( 0.000%)
GRE IP6 Ext: 0 ( 0.000%)
```

Figura 4: Snort interrompido

**Resposta:**

Resultado do modo Sniffer do Snort.



## 3.2 Tarefa 4 - Iniciar o Snort e testar a configuração

### 3.2.1 Exercício 6: No computador alvo, interrompa a execução do Snort. Observe com atenção o relatório apresentado pelo Snort e anote, em particular:

- a) Quantos pacotes o Snort recebeu e qual a distribuição desses pacotes por protocolos fundamentais (TCP, UDP, ICMP e ARP)? Justifique a observação com base na atividade criada.

```
Breakdown by protocol (includes rebuilt packets):
  Eth:          320 (100.000%)
  VLAN:         0 ( 0.000%)
  IP4:         298 ( 93.125%)
  Frag:         0 ( 0.000%)
  ICMP:         3 ( 0.938%)
  UDP:         70 ( 21.875%)
  TCP:        218 ( 68.125%)
  IP6:          0 ( 0.000%)
  IP6 Ext:      0 ( 0.000%)
  IP6 Opts:     0 ( 0.000%)
  Frag6:        0 ( 0.000%)
  ICMP6:        0 ( 0.000%)
  UDP6:         0 ( 0.000%)
  TCP6:         0 ( 0.000%)
  Teredo:       0 ( 0.000%)
  ICMP-IP:      0 ( 0.000%)
  IP4/IP4:      0 ( 0.000%)
  IP4/IP6:      0 ( 0.000%)
  IP6/IP4:      0 ( 0.000%)
  IP6/IP6:      0 ( 0.000%)
  GRE:          0 ( 0.000%)
  GRE Eth:      0 ( 0.000%)
  GRE VLAN:     0 ( 0.000%)
  GRE IP4:      0 ( 0.000%)
  GRE IP6:      0 ( 0.000%)
  GRE IP6 Ext:  0 ( 0.000%)
  GRE PPTP:     0 ( 0.000%)
  GRE ARP:      0 ( 0.000%)
  GRE IPX:      0 ( 0.000%)
  GRE Loop:     0 ( 0.000%)
  MPLS:         0 ( 0.000%)
  ARP:         22 ( 6.875%)
  IPX:          0 ( 0.000%)
  Eth Loop:     0 ( 0.000%)
  Eth Disc:     0 ( 0.000%)
  IP4 Disc:     7 ( 2.188%)
  IP6 Disc:     0 ( 0.000%)
  TCP Disc:     0 ( 0.000%)
  UDP Disc:     0 ( 0.000%)
  ICMP Disc:    0 ( 0.000%)
  All Discard:  7 ( 2.188%)
  Other:        0 ( 0.000%)
  Bad Chk Sum:  2 ( 0.625%)
  Bad TTL:      0 ( 0.000%)
  S5 G 1:       1 ( 0.312%)
  S5 G 2:       5 ( 1.562%)
  Total:       320
```

Figura 5: Relatório do Snort

### Resposta:

Relativamente aos pacotes TCP, UDP e ICMP, estes podem ser justificados por tráfego na rede não relevante, tal como aceder a algum website, stream, pings, entre outros. Os pacotes ARP detetados resultam da tentativa de arp-poisoning por parte do computador teste.

- b) Quantos alertas foram registados (se tudo correu bem, devem ter sido registados vários alertas)?

```
Action Stats:
Alerts:      7 ( 2.188%)
Logged:      7 ( 2.188%)
Passed:      0 ( 0.000%)
```

Figura 6: Alertas registados

### Resposta:

Foram detetados 7 alertas.

### 3.2.2 Exercício 7: Copie para o relatório o conteúdo do ficheiro de alertas, juntamente com o conteúdo do ficheiro "snort.log.\*"e comente o mesmo.

```
[**] [1:469:3] ICMP PING NMAP [**]
[Classification: Attempted Information Leak] [Priority: 2]
12/18-22:14:14.181998 192.168.1.1 -> 192.168.1.5
ICMP TTL:64 TOS:0x0 ID:32487 IpLen:20 DgmLen:28
Type:8 Code:0 ID:32487 Seq:32487 ECHO
[Xref => http://www.whitehats.com/info/IDS162]

[**] [1:384:5] ICMP PING [**]
[Classification: Misc activity] [Priority: 3]
12/18-22:14:14.181998 192.168.1.1 -> 192.168.1.5
ICMP TTL:64 TOS:0x0 ID:32487 IpLen:20 DgmLen:28
Type:8 Code:0 ID:32487 Seq:32487 ECHO

[**] [1:408:5] ICMP Echo Reply [**]
[Classification: Misc activity] [Priority: 3]
12/18-22:14:14.182034 192.168.1.5 -> 192.168.1.1
ICMP TTL:64 TOS:0x0 ID:12388 IpLen:20 DgmLen:28
Type:0 Code:0 ID:32487 Seq:32487 ECHO REPLY

[**] [1:408:5] ICMP Echo Reply [**]
[Classification: Misc activity] [Priority: 3]
12/18-22:14:14.182072 192.168.1.1 -> 192.168.1.5
ICMP TTL:64 TOS:0x0 ID:29214 IpLen:20 DgmLen:28
Type:0 Code:0 ID:32487 Seq:32487 ECHO REPLY

[**] [112:2:1] (spp arpspoof) Ethernet/ARP Mismatch request for Source [**]
12/18-22:14:30.261416

[**] [112:2:1] (spp arpspoof) Ethernet/ARP Mismatch request for Source [**]
12/18-22:14:31.282369

[**] [112:2:1] (spp arpspoof) Ethernet/ARP Mismatch request for Source [**]
12/18-22:14:32.302641
```

Figura 7: Ficheiro de alertas

As primeiras quatro entradas são as tentativas de PING do computador teste para o computador alvo. Por outro lado, as três últimas representam o alerta de *ARP-poisoning*.

```
Breakdown by protocol (includes rebuilt packets):
```

Eth:	7 (100.000%)
VLAN:	0 ( 0.000%)
IP4:	4 ( 57.143%)
Frag:	0 ( 0.000%)
ICMP:	4 ( 57.143%)
UDP:	0 ( 0.000%)
TCP:	0 ( 0.000%)
IP6:	0 ( 0.000%)
IP6 Ext:	0 ( 0.000%)
IP6 Opts:	0 ( 0.000%)
Frag6:	0 ( 0.000%)
ICMP6:	0 ( 0.000%)
UDP6:	0 ( 0.000%)
TCP6:	0 ( 0.000%)
Teredo:	0 ( 0.000%)
ICMP-IP:	0 ( 0.000%)
IP4/IP4:	0 ( 0.000%)
IP4/IP6:	0 ( 0.000%)
IP6/IP4:	0 ( 0.000%)
IP6/IP6:	0 ( 0.000%)
GRE:	0 ( 0.000%)
GRE Eth:	0 ( 0.000%)
GRE VLAN:	0 ( 0.000%)
GRE IP4:	0 ( 0.000%)
GRE IP6:	0 ( 0.000%)
GRE IP6 Ext:	0 ( 0.000%)
GRE PPTP:	0 ( 0.000%)
GRE ARP:	0 ( 0.000%)
GRE IPX:	0 ( 0.000%)
GRE Loop:	0 ( 0.000%)
MPLS:	0 ( 0.000%)
ARP:	3 ( 42.857%)
IPX:	0 ( 0.000%)
Eth Loop:	0 ( 0.000%)
Eth Disc:	0 ( 0.000%)
IP4 Disc:	0 ( 0.000%)
IP6 Disc:	0 ( 0.000%)
TCP Disc:	0 ( 0.000%)
UDP Disc:	0 ( 0.000%)
ICMP Disc:	0 ( 0.000%)
All Discard:	0 ( 0.000%)
Other:	0 ( 0.000%)
Bad Chk Sum:	0 ( 0.000%)
Bad TTL:	0 ( 0.000%)
S5 G 1:	0 ( 0.000%)
S5 G 2:	0 ( 0.000%)
Total:	7

Figura 8: Pacotes que originam os alertas

Mais uma vez, os três pacotes ARP, presentes na figura, referenciam os três alertas provocados por ARP-poisoning.

### 3.3 Tarefa 5 - Criar uma configuração que utiliza a *detection engine* e testá-la

3.3.1 Exercício 6: No computador de teste, inicie uma sessão Telnet com o servidor; siga as instruções do seu servidor para o ativar, registrando o processo no logbook.

```
(testpc@kali)-[~]
$ telnet 192.168.1.5 23
Trying 192.168.1.5...
Connected to 192.168.1.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
carlos-VirtualBox login: carlos
Password:

Login incorrect
carlos-VirtualBox login: carlos
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-58-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

14 as atualizações podem ser instaladas imediatamente.
0 destas atualizações são atualizações de segurança.
Para ver as actualizações adicionais corre o comando: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Fri Dec 18 19:36:58 WET 2020 from kali.home on pts/2
carlos@carlos-VirtualBox:~$ logout
Connection closed by foreign host.
```

Figura 9: Início de sessão Telnet

É necessário ter em atenção que ao usar as regras que ditam o ficheiro “snort\_detection.conf” foram detetados 0 alertas. Posto isto, o ficheiro utilizado passou a ser o ficheiro original, “snort.conf”

### 3.3.2 Exercício 7: No computador alvo, interrompa a execução do Snort com Ctrl+C

- a) Quantos pacotes recebeu?

```
Packet I/O Totals:
Received:          537
Analyzed:          535 ( 99.628%)
Dropped:           0 (  0.000%)
Filtered:          0 (  0.000%)
Outstanding:       2 (  0.372%)
Injected:          0
```

Figura 10: Pacotes recebidos pelo Snort

**Resposta:**

O Snort recebeu 537 pacotes.

- b) Qual o número de pacotes TCP?

```
TCP:              391 ( 72.542%)
```

Figura 11: Pacotes TCP recebidos pelo Snort

**Resposta:**

391 pacotes TCP.

- c) Quantos alertas foram gerados?

```
Action Stats:
Alerts:           18 (  3.340%)
Logged:           18 (  3.340%)
Passed:           0 (  0.000%)
```

Figura 12: Alertas gerados pelo Snort

**Resposta:**

Foram gerados 18 alertas.

**3.3.3 Exercício 8:** No computador alvo, observe o conteúdo do ficheiro de alertas. Anote todos os alertas detetados e justifique-os à luz da atividade criada. Consegue localizar os alertas gerados pelo ataque ARP poisoning? Justifique.

```
[**] [112:2:1] (spp arpspoof) Ethernet/ARP Mismatch request for Source [**]  
12/18-22:20:05.217474  
  
[**] [112:2:1] (spp arpspoof) Ethernet/ARP Mismatch request for Source [**]  
12/18-22:20:06.238456  
  
[**] [112:2:1] (spp arpspoof) Ethernet/ARP Mismatch request for Source [**]  
12/18-22:20:07.259112
```

Figura 13: Alertas relativos ao ARP-poisoning

```
[**] [129:12:1] Consecutive TCP small segments exceeding threshold [**]  
[Classification: Potentially Bad Traffic] [Priority: 2]  
12/18-22:20:17.104022 192.168.1.9:49746 -> 192.168.1.5:23  
TCP TTL:64 TOS:0x10 ID:15200 IpLen:20 DgmLen:55 DF  
***AP*** Seq: 0xE0E6ADCF Ack: 0xDABA5D51 Win: 0x1F6 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 454910449 1764055971  
  
[**] [129:12:1] Consecutive TCP small segments exceeding threshold [**]  
[Classification: Potentially Bad Traffic] [Priority: 2]  
12/18-22:20:19.179841 192.168.1.9:49746 -> 192.168.1.5:23  
TCP TTL:64 TOS:0x10 ID:15216 IpLen:20 DgmLen:53 DF  
***AP*** Seq: 0xE0E6ADD5 Ack: 0xDABA5D81 Win: 0x1F6 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 454912524 1764057848 |  
  
[**] [129:12:1] Consecutive TCP small segments exceeding threshold [**]  
[Classification: Potentially Bad Traffic] [Priority: 2]  
12/18-22:20:25.439181 192.168.1.9:49746 -> 192.168.1.5:23  
TCP TTL:64 TOS:0x10 ID:15230 IpLen:20 DgmLen:53 DF  
***AP*** Seq: 0xE0E6ADDA Ack: 0xDABA5D90 Win: 0x1F6 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 454918784 1764063688  
  
[**] [129:12:1] Consecutive TCP small segments exceeding threshold [**]  
[Classification: Potentially Bad Traffic] [Priority: 2]  
12/18-22:20:30.034602 192.168.1.9:49746 -> 192.168.1.5:23  
TCP TTL:64 TOS:0x10 ID:15241 IpLen:20 DgmLen:53 DF  
***AP*** Seq: 0xE0E6ADDF Ack: 0xDABA5DBE Win: 0x1F6 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 454923380 1764067409  
  
[**] [129:12:1] Consecutive TCP small segments exceeding threshold [**]  
[Classification: Potentially Bad Traffic] [Priority: 2]  
12/18-22:20:30.755419 192.168.1.9:49746 -> 192.168.1.5:23  
TCP TTL:64 TOS:0x10 ID:15254 IpLen:20 DgmLen:53 DF  
***AP*** Seq: 0xE0E6ADE3 Ack: 0xDABA5DC2 Win: 0x1F6 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 454924100 1764069469  
  
[**] [129:12:1] Consecutive TCP small segments exceeding threshold [**]  
[Classification: Potentially Bad Traffic] [Priority: 2]  
12/18-22:20:33.462111 192.168.1.9:49746 -> 192.168.1.5:23  
TCP TTL:64 TOS:0x10 ID:15266 IpLen:20 DgmLen:53 DF  
***AP*** Seq: 0xE0E6ADE8 Ack: 0xDABA5DD0 Win: 0x1F6 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 454926807 1764072206  
  
[**] [129:12:1] Consecutive TCP small segments exceeding threshold [**]  
[Classification: Potentially Bad Traffic] [Priority: 2]  
12/18-22:20:34.306628 192.168.1.9:49746 -> 192.168.1.5:23  
TCP TTL:64 TOS:0x10 ID:15270 IpLen:20 DgmLen:53 DF  
***AP*** Seq: 0xE0E6ADEC Ack: 0xDABA5DD0 Win: 0x1F6 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 454927651 1764072960
```

Figura 14: Alertas relativos à conexão Telnet

### 3.4 Tarefa 6 - Criar uma configuração que utiliza a *detection engine*, uma regra específica e testá-la

#### 3.4.1 Exercício 2: Utilizando o seu editor preferido crie um ficheiro com o nome `subseven.rules` e acrescente-lhe uma linha com o texto "alert tcp anyday -> any 27374". Qual o objetivo desta regra?

**Resposta:**

Alertar, quando existir tráfego TCP, de qualquer porta para a porta 27374.

#### 3.4.2 Exercício 6: No computador alvo, interrompa a execução do Snort.

- a) Quantos pacotes recebeu o Snort?

```
Packet I/O Totals:
Received:          1968
Analyzed:          1965 ( 99.848%)
Dropped:           0 ( 0.000%)
Filtered:           0 ( 0.000%)
Outstanding:        3 ( 0.152%)
Injected:           0
```

Figura 15: Pacotes recebidos

**Resposta:**

Recebeu 1968 pacotes.

- b) Quantos alertas foram gerados?

```
Alerts:              4 ( 0.204%)
```

Figura 16: Alertas gerados

**Resposta:**

Foram registados quatro alertas.

### 3.4.3 Exercício 7: Observe o conteúdo do ficheiro de alertas. Deverá identificar aí um ou mais alertas.

- a) Anote todos os alertas detetados.

```
[**] [1:999:0] "SubSeven Connection Attempt" [**]
[Priority: 0]
12/18-19:21:43.327259 192.168.1.37:51888 -> 192.168.1.12:27374
TCP TTL:64 TOS:0x10 ID:31000 IpLen:20 DgmLen:60 DF
*****S* Seq: 0xDDF12B79 Ack: 0x0 Win: 0xFAF0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 3642443297 0 NOP WS: 7

[**] [1:999:0] "SubSeven Connection Attempt" [**]
[Priority: 0]
12/18-19:21:46.340623 192.168.1.37:51890 -> 192.168.1.12:27374
TCP TTL:64 TOS:0x10 ID:958 IpLen:20 DgmLen:60 DF
*****S* Seq: 0x81FF03B8 Ack: 0x0 Win: 0xFAF0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 3642446311 0 NOP WS: 7

[**] [1:999:0] "SubSeven Connection Attempt" [**]
[Priority: 0]
12/18-19:21:47.860745 192.168.1.37:51892 -> 192.168.1.12:27374
TCP TTL:64 TOS:0x10 ID:60119 IpLen:20 DgmLen:60 DF
*****S* Seq: 0xE116BDA8 Ack: 0x0 Win: 0xFAF0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 3642447831 0 NOP WS: 7

[**] [1:999:0] "SubSeven Connection Attempt" [**]
[Priority: 0]
12/18-19:21:49.797125 192.168.1.37:51894 -> 192.168.1.12:27374
TCP TTL:64 TOS:0x10 ID:36494 IpLen:20 DgmLen:60 DF
*****S* Seq: 0xB09CEA1D Ack: 0x0 Win: 0xFAF0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 3642449767 0 NOP WS: 7
```

Figura 17: Alertas detetados

- b) Consegue localizar os alertas gerados pela atividade que acabou de gerar?

**Resposta:**

Todos os alertas apresentados acima são relativos à tentativa, sem sucesso, de conexão telnet para a porta 27374.

- c) Acha que o alerta gerado está de acordo com o objectivo da regra, anteriormente enunciado (ver 2, acima)?

**Resposta:**

Sim.

- d) Classificaria este alerta como um “Falso Positivo”?

**Resposta:**

Sim, uma vez que uma tentativa de conexão para uma porta fechada não deve ser relevante.



## 4 Conclusão

Este foi um trabalho bastante importante, no sentido em que nos permitiu enriquecer os nossos conhecimentos acerca da temática de sistemas de deteção de intrusões. Consideramos que atingimos os objetivos propostos à partida, uma vez que somos agora capazes de trabalhar, com algum conforto, com a ferramenta Snort, através da qual analisámos os alertas gerados pela mesma, criámos regras específicas para um determinado ataque e configurámos o pré processador e o *detection engine*.

## 5 Bibliografia

"Intrusion Detection and Correlation: challenges and solutions", de Kruegel, 2005, da página 17 à 28

<http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node1.html>