Sistemas de detección de intrusos

Juan Pablo Donoso

Servidores Web de Altas Prestaciones

Diapositivas disponibles en github.com/juanpablodonoso/IDS

Definición

Un Sistema de Detección de Intrusos se engarga de registrar toda la actividad que pueda ser maliciosa en un sistema o red, es decir la cuál amenace la **D**isponibiliad, **C**onfidencialidad e **I**ntegridad de la red.

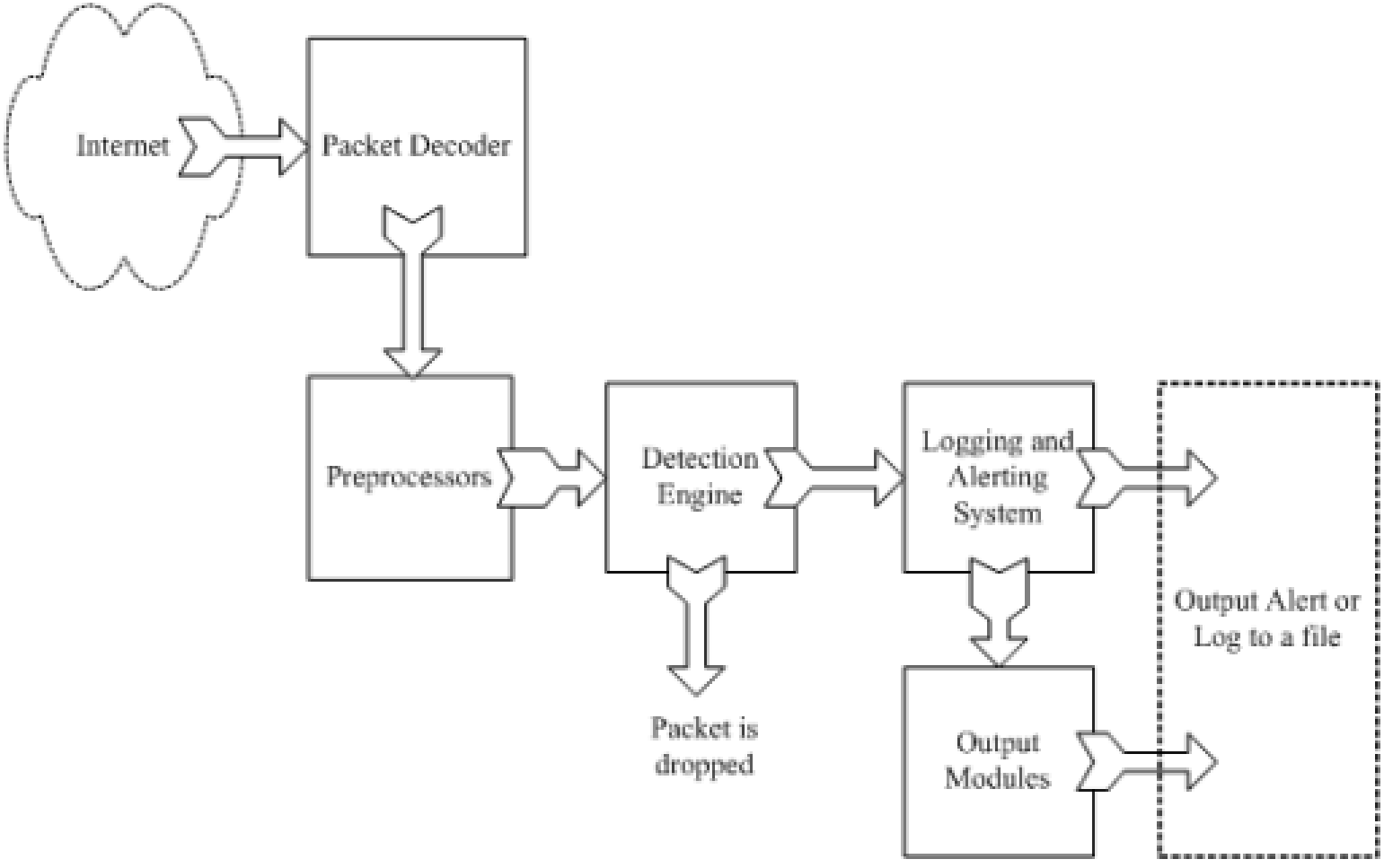
Esta actvidad es llamada intento de intrusión (**N**etwork **I**ntrusion **A**ttempt) donde un evento, paquete o agente es un intruso.

Comportamiento

Cuando un *NIA* es detectado se produce una **alerta** al administrados del sistema, que pasa a formar parte de un sistema de **logs**.

El sistema de alertas es el encargado de gestionar falsas alarmas o **falsos positivos**.

Arquitectura (Snort)

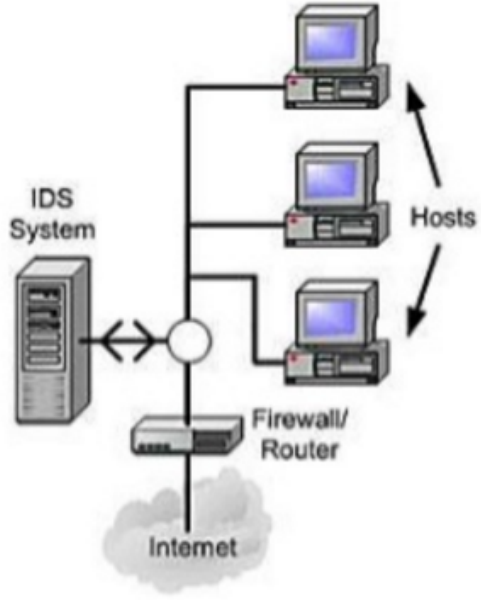


Comportamiento (II)

En función del comportamiento del IDS podemos realizar una primera clasificación

- Actividad analizada
  - **Basados en red**
  - Basado en host
- Método de detección
  - Basados en firmas (*Snort*)
  - Basados en anomalías

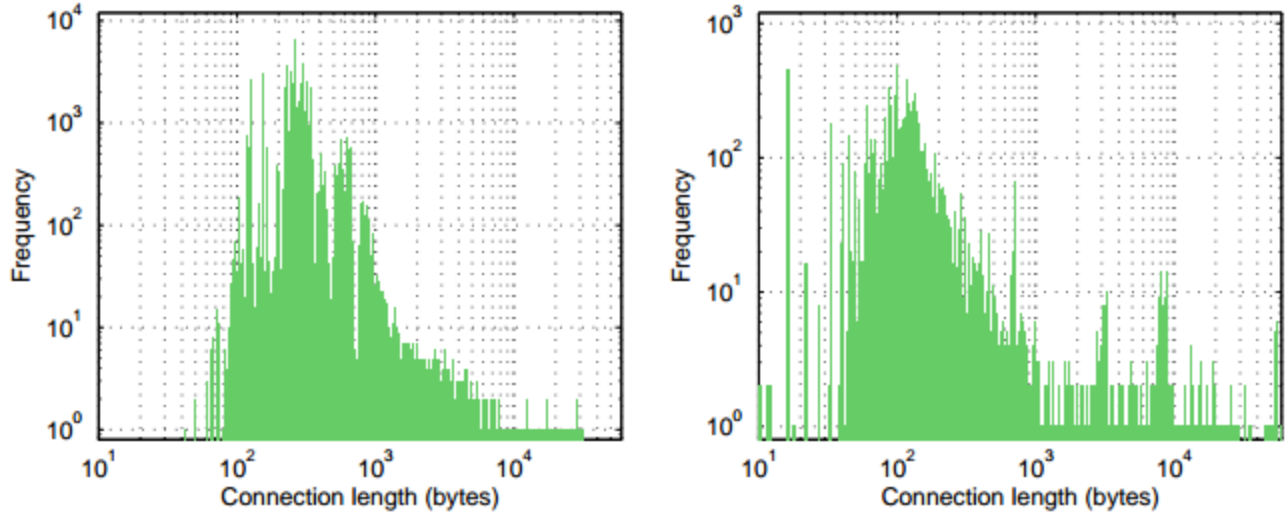
Deteción de intrusos basada en red (HIDS)



Analizan segmentos de la subred realizando un matching de los elementos de los paquetes de datos que viajan por la red con una base de datos o reglas.

Deteción de intrusos basada en red

Los dispositivos en este tipo de sisteman han de estar en modo **primíscuo** lo que nos permite analizar todas las conexiones de una forma completa. Por ejempl la duración de las conexiones

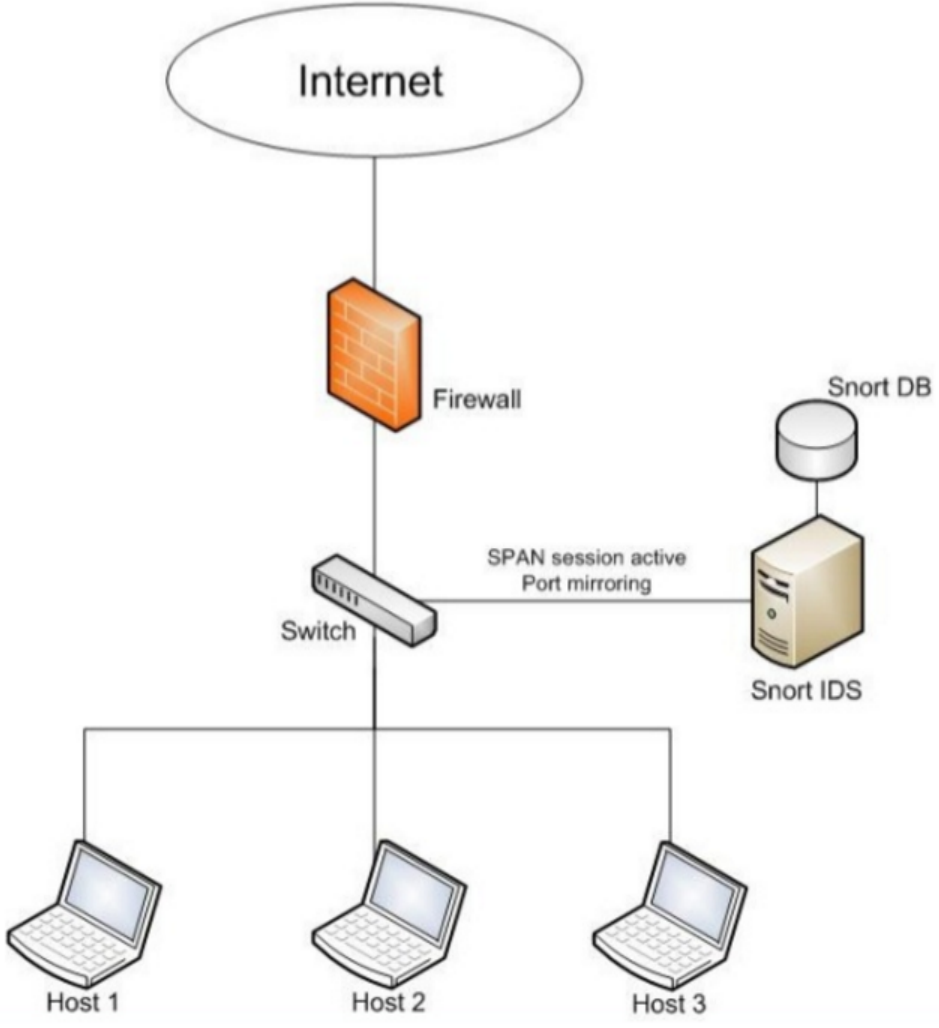


(a) HTTP data set

(b) FTP data set

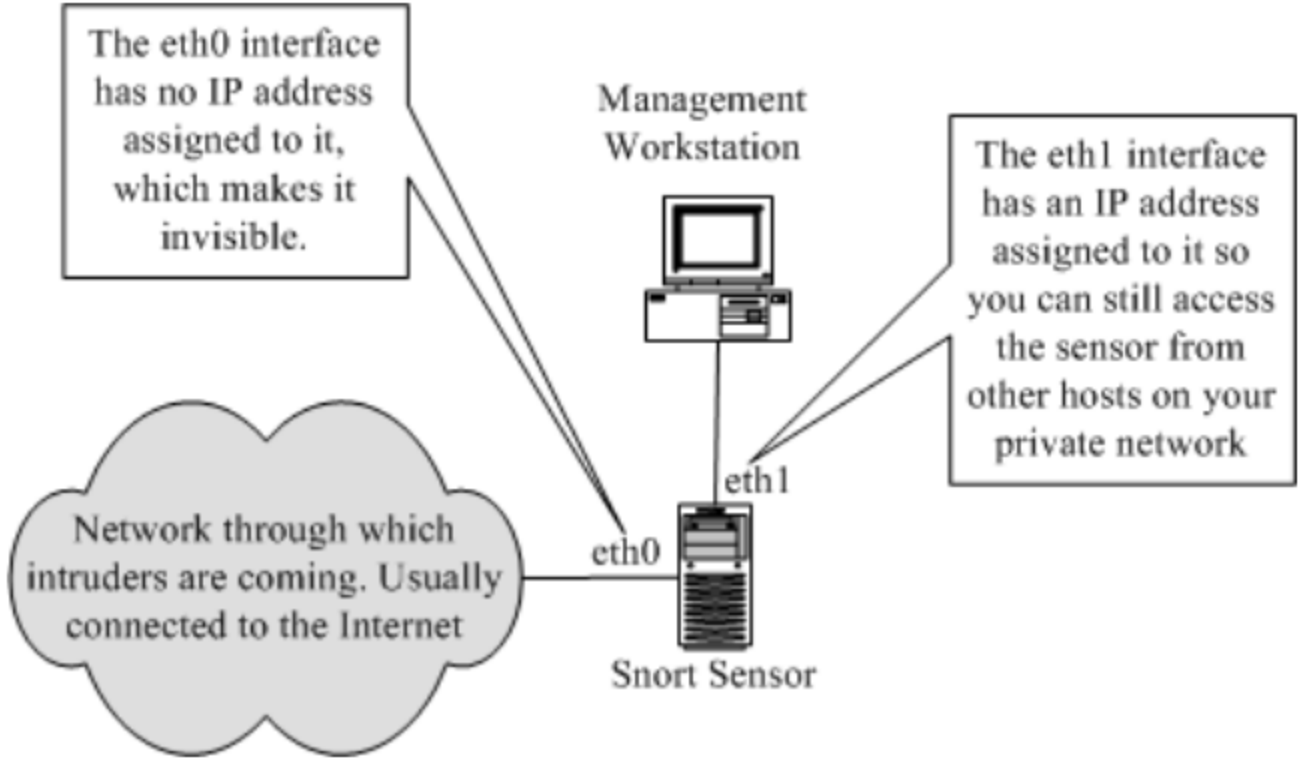
Deteción de intrusos basada en red

Un ejemplo de este tipo de arquitectura podría ser el siguiente

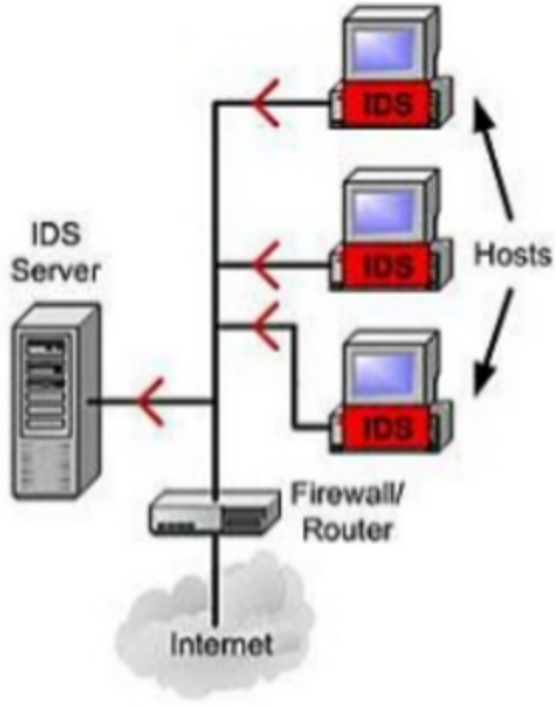


Deteción de intrusos basada en red

Es el tipo más común, ya que nos permite colocar **sensores** físicos o virtuales en la red, permitiendonos aislar segmentos de esta



Detección de intrusos basada en host



Comportamiento (II)

En función del comportamiento del IDS podemos realizar una primera clasificación

- Actividad analizada
  - Basados en red
  - Basado en host
- Método de detección
  - **Basados en firmas (*Snort*)**
  - Basados en anomalías

Métodos de detección

Los IDS presentan un comportamiento my diferente en función del método de detección de intentos de intrusión.

El método decide qué está bien o qué está mal en nuestra red.

Detección basasa en reglas (Signature-based)

Este tipo de detección está baso en el matching de una cadena extraída de un paquete de datos del segmento de red con expresiones o estructuras almacenadas en una base de conocimiento.

Detección basasa en reglas (Signature-based)

Las firmas estarán presentes en diferentes partes del datagrama dependiendo de la naturaleza del ataque. Por ejemplo podemos encontrar firmas en la cabecera IP, el la cabecera de la capa de transporte (cabeceras TCP o UPD) y en la cabecera de la capa de aplicación o payload. En IDS como Snort, las firmas son actualizadas por el usuario mientras que en otro tipo de sistemas son responsabilidad del vendedor del software.

Detección basasa en reglas (Signature-based)

Ejemplo 1 - Parser de protocolo de apliacación

```
Request          = Request-Line *(Header CRLF) CRLF Message-Body
Request-Line     = Method SP Request-URI SP HTTP-Version CRLF
Method           = "OPTIONS" | "GET" | "HEAD" | "POST" | ...
Request-URI      = * | absoluteURI | abs_path | authority
```

Detección basasa en reglas (Signature-based)

Ejemplo 1 - Parser de protocolo de aplicación

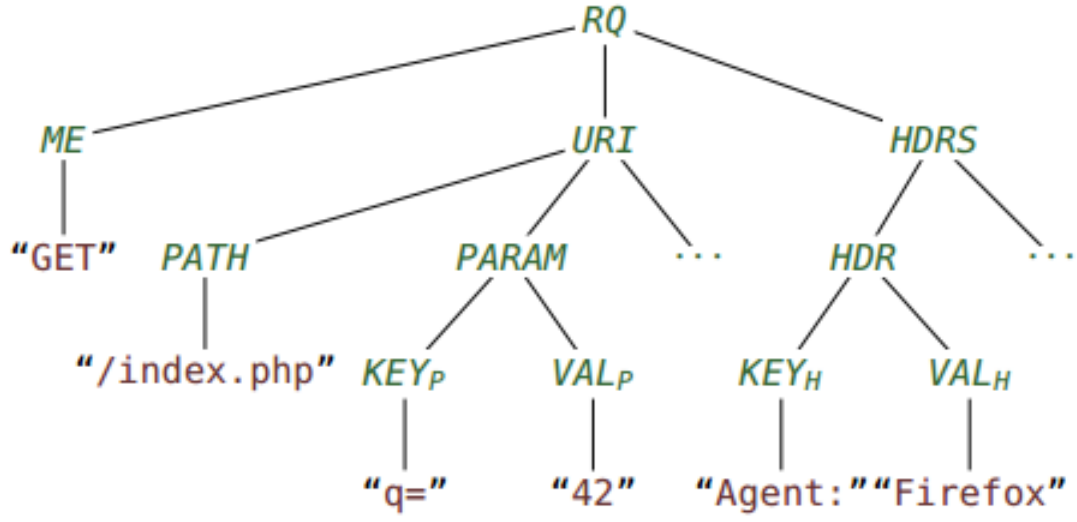
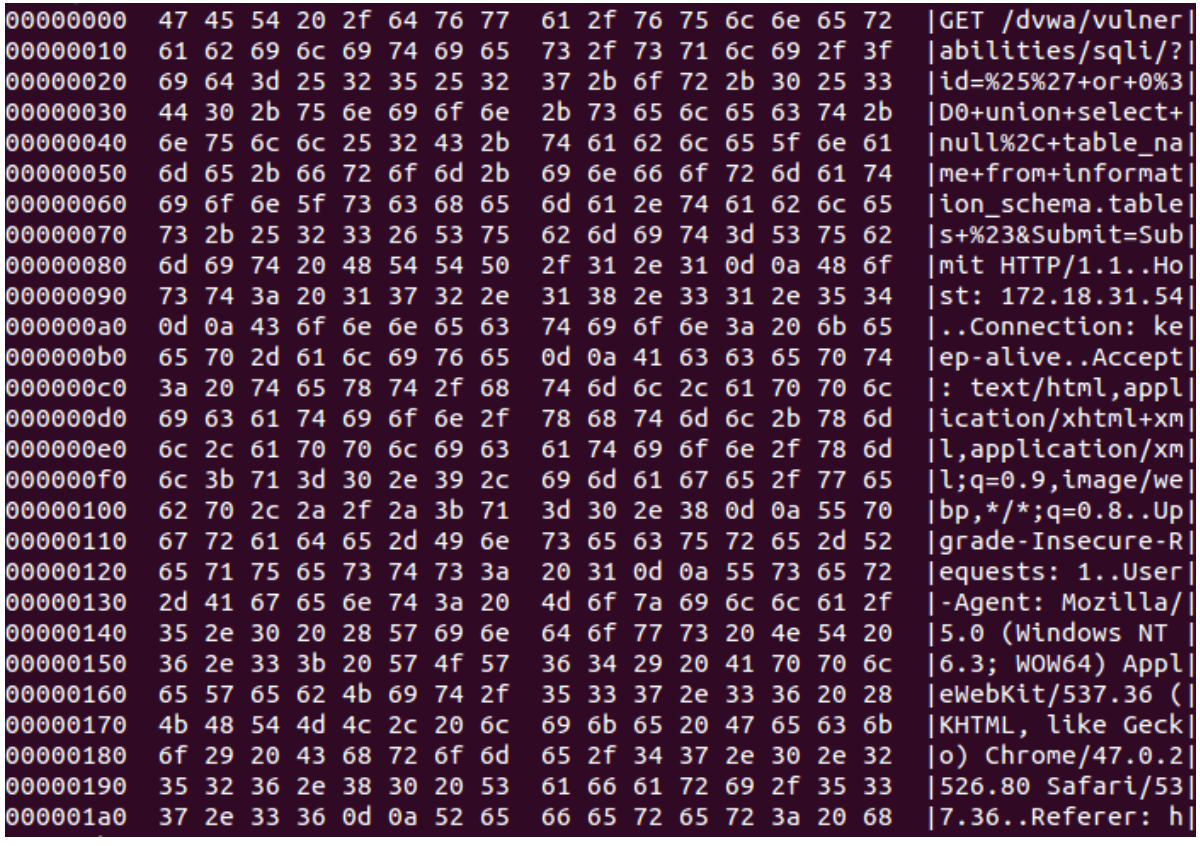


Figure 2.3: Simplified parse tree for an HTTP request. Abbreviations: request (*RQ*), method (*ME*), URI parameters (*PARAM*), URI parameter key and value (KEY_P, VAL_P), headers (*HDRS*), header (*HDR*), header key and value (KEY_H, VAL_H).

Detección basasa en reglas (Signature-based)

Ejemplo 2 - Detección comandos



```
00000000  47 45 54 20 2f 64 76 77  61 2f 76 75 6c 6e 65 72  |GET /dvwa/vulner|
00000010  61 62 69 6c 69 74 69 65  73 2f 73 71 6c 69 2f 3f  |abilities/sqli/?|
00000020  69 64 3d 25 32 35 25 32  37 2b 6f 72 2b 30 25 33  |id=%25%27+or+0%3|
00000030  44 30 2b 75 6e 69 6f 6e  2b 73 65 6c 65 63 74 2b  |D0+union+select+|
00000040  6e 75 6c 6c 25 32 43 2b  74 61 62 6c 65 5f 6e 61  |null%2C+table_na|
00000050  6d 65 2b 66 72 6f 6d 2b  69 6e 66 6f 72 6d 61 74  |me+from+informat|
00000060  69 6f 6e 5f 73 63 68 65  6d 61 2e 74 61 62 6c 65  |ion_schema.table|
00000070  73 2b 25 32 33 26 53 75  62 6d 69 74 3d 53 75 62  |s+%23&Submit=Sub|
00000080  6d 69 74 20 48 54 54 50  2f 31 2e 31 0d 0a 48 6f  |mit HTTP/1.1..Ho|
00000090  73 74 3a 20 31 37 32 2e  31 38 2e 33 31 2e 35 34  |st: 172.18.31.54|
000000a0  0d 0a 43 6f 6e 6e 65 63  74 69 6f 6e 3a 20 6b 65  |..Connection: ke|
000000b0  65 70 2d 61 6c 69 76 65  0d 0a 41 63 63 65 70 74  |ep-alive..Accept|
000000c0  3a 20 74 65 78 74 2f 68  74 6d 6c 2c 61 70 70 6c  |: text/html,appl|
000000d0  69 63 61 74 69 6f 6e 2f  78 68 74 6d 6c 2b 78 6d  |ication/xhtml+xm|
000000e0  6c 2c 61 70 70 6c 69 63  61 74 69 6f 6e 2f 78 6d  |l,application/xm|
000000f0  6c 3b 71 3d 30 2e 39 2c  69 6d 61 67 65 2f 77 65  |l;q=0.9,image/we|
00000100  62 70 2c 2a 2f 2a 3b 71  3d 30 2e 38 0d 0a 55 70  |bp,*/*;q=0.8..Up|
00000110  67 72 61 64 65 2d 49 6e  73 65 63 75 72 65 2d 52  |grade-Insecure-R|
00000120  65 71 75 65 73 74 73 3a  20 31 0d 0a 55 73 65 72  |equests: 1..User|
00000130  2d 41 67 65 6e 74 3a 20  4d 6f 7a 69 6c 6c 61 2f  |-Agent: Mozilla/|
00000140  35 2e 30 20 28 57 69 6e  64 6f 77 73 20 4e 54 20  |5.0 (Windows NT |
00000150  36 2e 33 3b 20 57 4f 57  36 34 29 20 41 70 70 6c  |6.3; WOW64) Appl|
00000160  65 57 65 62 4b 69 74 2f  35 33 37 2e 33 36 20 28  |eWebKit/537.36 (|
00000170  4b 48 54 4d 4c 2c 20 6c  69 6b 65 20 47 65 63 6b  |KHTML, like Geck|
00000180  6f 29 20 43 68 72 6f 6d  65 2f 34 37 2e 30 2e 32  |o) Chrome/47.0.2|
00000190  35 32 36 2e 38 30 20 53  61 66 61 72 69 2f 35 33  |526.80 Safari/53|
000001a0  37 2e 33 36 0d 0a 52 65  66 65 72 65 72 3a 20 68  |7.36..Referer: h|
```

```
%25%27+0%3D0+union+select+null%2C+table_name+from+informa
tion_schema.tables+%23
```

Detección basasa en reglas (Signature-based)

Ejemplo 2 - Detección comandos

```
%’ or 0=0 union select null, table_name from information_schema.tables #
```

Detección basasa en reglas (Signature-based)

Ejemplo 3 - Detección características en payloads

```
{ “/bin/sh”, “/etc/passwd”, “admin”, “cmd.exe”, “dll”, “script”, “root” }
```

Detección basasa en reglas (Signature-based) - Dónde están las reglas

Las firmas son establecidas por el proveedor o bien por el usuario en software configurable como Snort en el archivo de configuración

```
alert icmp any any -> any any (msg: "ICMP Packet found";)
```

Detección basasa en anomalías

Son sistemas opuestos a los anteriores, ya que intentan proteger al sistema de ataques que no se han producido en el pasado y de los cuales no se tienen datos. Implementan técnicas de Machine Learning

Detección basasa en anomalías

Se componen de una fase de entranamiento en la que se crea un perfil de "normalidad" de la red, seguida de una fase de test donde se realiza la comparativa del estado actual con el estado de entranamiento.

Detección basasa en anomalías

Nos permiten detectar ataques de los cuales no tenemos datos en nuestra base de conocimiento pero puede haber un gran número de **falsos positivos**, ya que cualquier evento fuera del estado normalidad se clasificará como malicioso

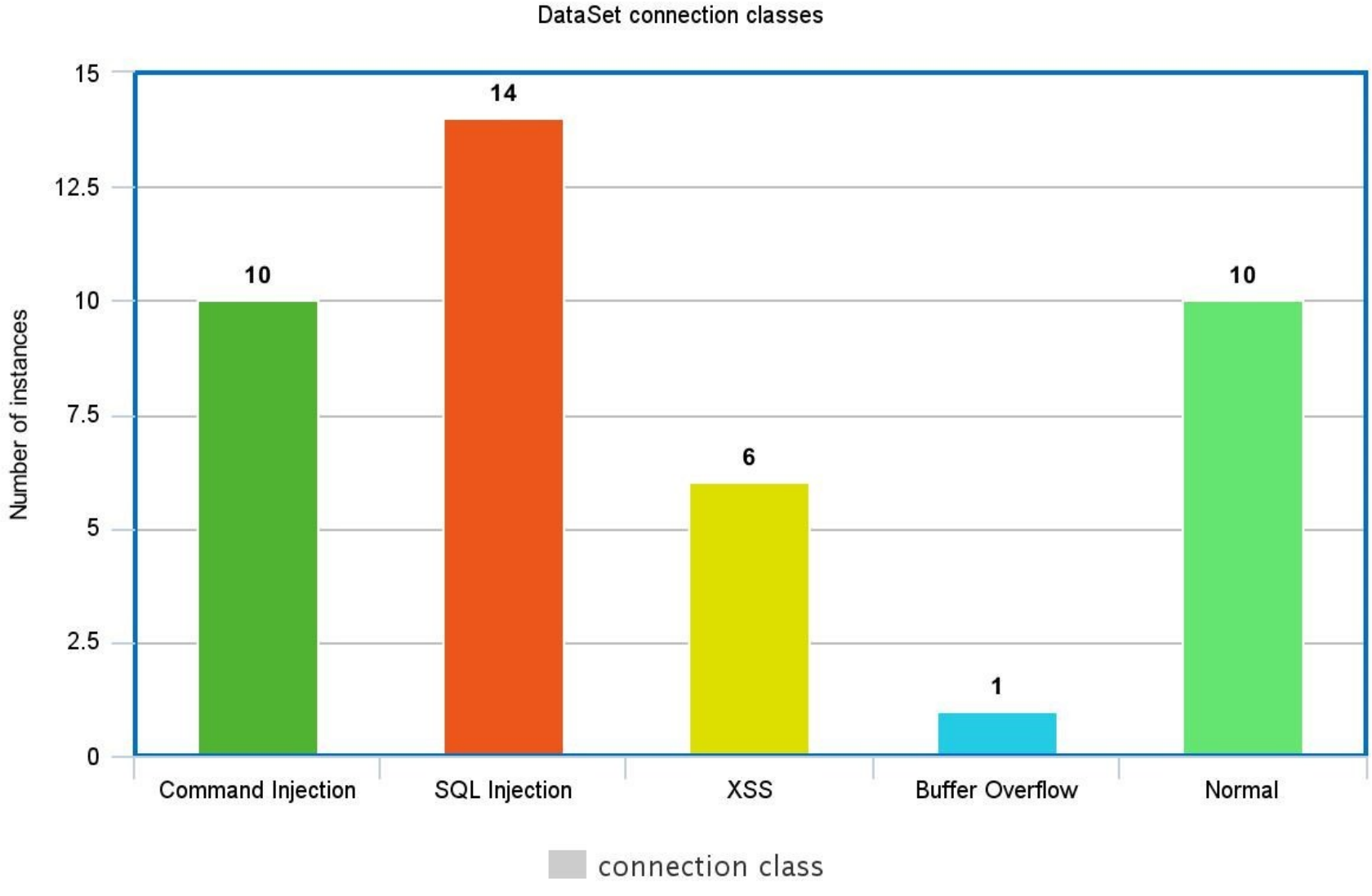
Aplicación de técnicas de Machine Learning en IDS

La aplicación de estas técnicas nos permite clasificar los tipos de ataques e intentar inferir los ataques que estań por llegar. Para ello es necesario un conjunto de entramiento a partir de los datos registrados en ataques previos a nuestra rerd. Con Snort podemos extraer el contenido de nuestra red en texto plano y procesarlo para así poder clasificar estos datos.

Aplicación de técnicas de Machine Learning en IDS

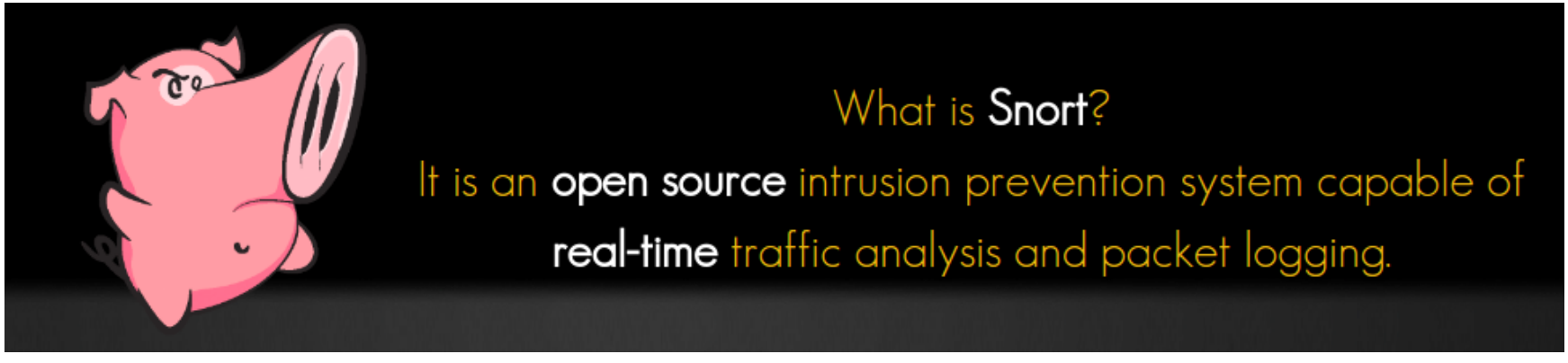
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	class	ts	uid	id.orig_h	id.orig	id.resp_h	id.resp	pro	ser	duration	JS	SQL	OS	entropy	nonPrintable	puntuation	securityKeyw	shellcode	Length
2	SQL Injection	###	C0bRaY12Aoi	172.18.1.	51044	172.18.31.54	80	tcp	http	6.70765	false	true	false	5.64	22	91	0	false	552
3	Normal	###	C37Uwl1AqnE	172.18.1.	51193	172.18.31.54	80	tcp	http	17.4053	false	false	false	5.59	110	428	0	false	2675
4	Normal	###	C3YmXMdh1o	172.18.1.	51040	172.18.31.54	80	tcp	http	10.0567	false	false	false	5.58	22	83	0	false	521
5	Command Injection	###	C5BfcwqQjmb	172.18.1.	51034	172.18.31.54	80	tcp	http	6.60796	false	false	true	5.57	30	106	0	false	668
6	Command Injection	###	C7fyI3KwnacH	172.18.1.	51035	172.18.31.54	80	tcp	http	10.0258	false	false	true	5.58	30	107	0	false	668
7	Buffer Overflow	###	C7gQnM1mWV	192.168.1	1049	192.168.1.9	8080	tcp	ftp	0.0271	false	false	false	6.32	25491	5923	1	yes	73802
8	Normal	###	Cbt8Cz1KEoJ	172.18.1.	51053	151.248.100	80	tcp	http	160.229	false	false	false	5.16	10	28	0	false	181
9	SQL Injection	###	CbWfka2CQH	172.18.1.	51147	172.18.31.54	80	tcp	http	10.0331	false	true	false	5.58	22	151	0	false	857
10	XSS	###	CCZ73W1fwQ	172.18.1.	51190	172.18.31.54	80	tcp	http	10.5381	true	false	false	5.66	118	515	0	false	3066
11	XSS	###	Cdig5tJs3fQz	172.18.1.	51192	172.18.31.54	80	tcp	http	10.0894	true	false	false	5.63	30	118	0	false	729
12	SQL Injection	###	CeeUGS1ScX	172.18.1.	51148	172.18.31.54	80	tcp	http	10.0303	false	true	false	5.60	22	145	0	false	833
13	Command Injection	###	CEzJ5h2mizK	172.18.1.	51036	172.18.31.54	80	tcp	http	6.49877	false	false	true	5.57	30	105	0	false	668
14	SQL Injection	###	Cg2lGsS04ZD	172.18.1.	51045	172.18.31.54	80	tcp	http	6.36746	false	true	false	5.64	22	91	0	false	551
15	Command Injection	###	CG7cHc4lLDO	172.18.1.	51031	172.18.31.54	80	tcp	http	6.01737	false	false	true	5.57	30	107	0	false	671
16	SQL Injection	###	CHDQza1Tnvr	172.18.1.	51070	172.18.31.54	80	tcp	http	10.0342	false	true	false	5.63	22	122	0	false	694
17	Normal	###	ChPWxxxLkltv	172.18.1.	51093	104.61.47.57	80	tcp	http	1.28782	false	false	false	5.63	18	69	0	false	454
18	SQL Injection	###	CJqlVK1bvaD	172.18.1.	51145	172.18.31.54	80	tcp	http	10.4676	false	true	false	5.60	22	138	0	false	789
19	Command Injection	###	CkAUwZFGFp	172.18.1.	51038	172.18.31.54	80	tcp	http	14.3013	false	false	true	5.58	60	211	0	false	1337
20	SQL Injection	###	Ckez4g8ThGe	172.18.1.	51066	172.18.31.54	80	tcp	http	10.0154	false	true	false	5.65	22	117	0	false	661
21	SQL Injection	###	CKfL3c3YvXS	172.18.1.	51050	172.18.31.54	80	tcp	http	10.0341	false	true	false	5.67	22	101	0	false	584
22	SQL Injection	###	CllgGV22xJ0n	172.18.1.	51067	172.18.31.54	80	tcp	http	10.0248	false	true	false	5.65	22	117	0	false	657
23	XSS	###	COZnXq4oSW	172.18.1.	51188	172.18.31.54	80	tcp	http	9.47139	true	false	false	5.65	22	96	0	false	575
24	SQL Injection	###	CPQSDZbba9	172.18.1.	51046	172.18.31.54	80	tcp	http	10.0153	false	true	false	5.64	22	93	0	false	553
25	Normal	###	CqOUnm19Ds	172.18.1.	51048	172.18.31.54	80	tcp	http	10.0261	false	false	false	5.61	22	87	0	false	540
26	Normal	###	Cr6K6a4TZN4	172.18.1.	51052	23.193.44.10	80	tcp	http	160.226	false	false	false	5.58	10	26	0	false	227
27	Normal	###	CrPQmp1OEU	172.18.1.	51187	172.18.31.54	80	tcp	http	9.50088	false	false	false	5.61	22	83	0	false	513
28	SQL Injection	###	CSVpc6OaQd	172.18.1.	51069	172.18.31.54	80	tcp	http	10.0269	false	true	false	5.65	22	119	0	false	663
29	Command Injection	###	CU8hin4Y2M5	172.18.1.	51029	172.18.31.54	80	tcp	http	5.22386	false	false	true	5.58	30	106	0	false	667
30	Normal	###	CuLPUP34Azt	172.18.1.	51104	94.142.38.30	80	tcp	http	86.4259	false	false	false	5.75	54	430	0	false	2631
31	Normal	###	CVtVbj3cBlbX	172.18.1.	51057	151.248.100	80	tcp	http	160.228	false	false	false	5.10	10	25	0	false	158
32	XSS	###	CvwFEh26lA5	172.18.1.	51184	172.18.31.54	80	tcp	http	10.0196	true	false	false	5.64	22	107	0	false	627

Aplicación de técnicas de Machine Learning en IDS



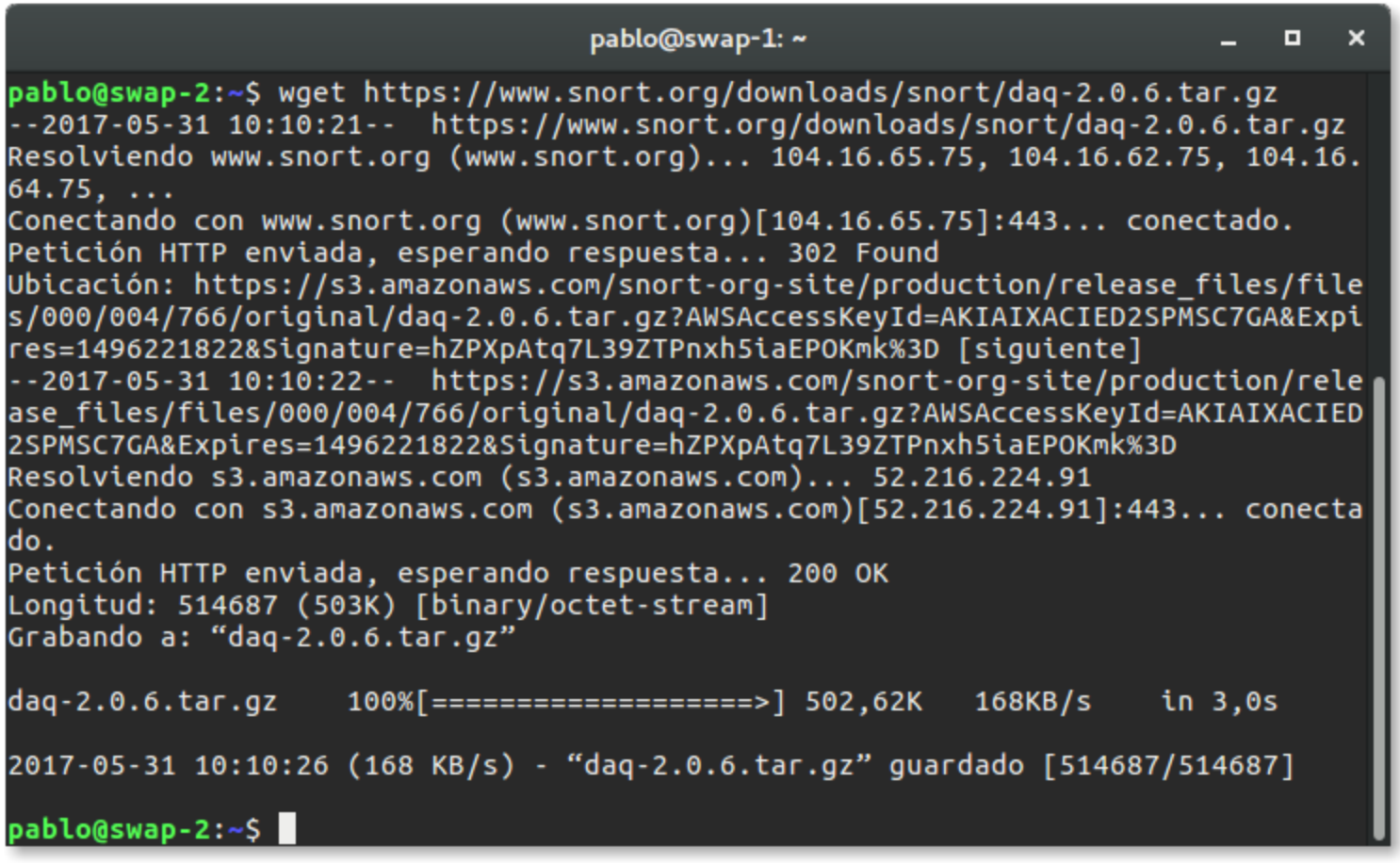
Snort

Snort es un software prevención de intrusos en red, que es capaz de analizar tráfico en tiempo real empleando software como `tcpdump` para el sniffing de paquetes o `flex` y `bison` para el análisis de expresiones dentro de estos paquetes.



***Intrusion Detection Systems with Snort Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID*, de Rafeeq Ur Rehman**

Ubuntu Server. Preparación del servidor



Para instalación detallada (mejor que doc. oficial)

https://www.upcloud.com/support/installing-snort-on-ubuntu/

Ubuntu Server. Preparación del servidor

En todos los servidores donde queramos ejecutar **Snort** debemos tener la siguiente configuración

Prerequisitos

- `build-essential`, `flex`, `bison`
- bibliotecas: `libpcap-dev`, `libpcre3-dev`, `libdumbnet-dev`, `zlib1g-dev` `libdnet`

```
# Ubuntu Server
sudo apt-get install build-essential libpcap-dev libpcre3
```

Para la configuración de Snort:

https://www.snort.org/documents/snort-2-9-9-x-on-ubuntu-14-16

Funcionamiento Snort

Método de detección

Snort está basado en reglas, que se emplean para la detección de NIAs.

Esta regla podría estar alojada en `/etc/snort/rules/sql.rules`

```
alert udp $EXTERNAL_NET any -> $HOME_NET 1434 (msg:"MS-SQL Worm
 propagation attempt"; \ content:"|04|"; depth:1; content:"|81 F1 03
 01 04 9B 81 F1 01|"; content:"sock"; content:"send"; \
 reference:bugtraq,5310; reference:bugtraq,5311;
 reference:cve,2002-0649; reference:nessus,11214; \
 reference:url,vil.nai.com/vil/content/v_99992.htm;
 classtype:misc-attack; sid:2003; rev:8;)
```

Base de datos para logs - Integración Mysql

```
mysql> show tables;
+--------------------+
| Tables_in_snort |
+--------------------+
| data |
| detail |
| encoding |
| event |
| flags |
| icmphdr |
| iphdr |
| opt |
| protocols |
| reference |
| reference_system |
| schema |
| sensor |
----> [...]
+--------------------+
19 rows in set (0.01 sec)
```

Base de datos para logs - Signatures

```
mysql> select * from sig_class;
+--------------+----------------------------+
| sig_class_id | sig_class_name |
+--------------+----------------------------+
| 9 | attempted-recon |
| 8 | misc-attack |
| 7 | bad-unknown |
| 6 | web-application-activity |
+--------------+----------------------------+
4 rows in set (0.00 sec)
```

Ejemplo de salida estándar

```
stu@ubuntu:~$ sudo snort -A console -q -c /etc/snort/snort.conf -i eth0
02/24-15:36:44.746693  [**] [1:1000008:1] Suspicious IP address [**] [Priority:
0] {TCP} 10.10.10.10:51471 -> 192.168.132.136:80
02/24-15:36:44.746950  [**] [1:1000008:1] Suspicious IP address [**] [Priority:
0] {TCP} 11.11.11.11:51471 -> 192.168.132.136:80
02/24-15:36:44.747222  [**] [1:1000008:1] Suspicious IP address [**] [Priority:
0] {TCP} 1.1.1.1:51471 -> 192.168.132.136:80
02/24-15:36:44.747226  [**] [1:1000008:1] Suspicious IP address [**] [Priority:
0] {TCP} 8.8.8.8:51471 -> 192.168.132.136:80
02/24-15:36:44.751845  [**] [1:1000008:1] Suspicious IP address [**] [Priority:
0] {TCP} 10.10.10.10:51471 -> 192.168.132.136:80
02/24-15:36:44.751846  [**] [1:1000008:1] Suspicious IP address [**] [Priority:
0] {TCP} 11.11.11.11:51471 -> 192.168.132.136:80
02/24-15:36:44.751847  [**] [1:1000008:1] Suspicious IP address [**] [Priority:
0] {TCP} 1.1.1.1:51471 -> 192.168.132.136:80
02/24-15:36:44.751848  [**] [1:1000008:1] Suspicious IP address [**] [Priority:
0] {TCP} 8.8.8.8:51471 -> 192.168.132.136:80
```