

TryHackMe: Snort Challenges - Live Attack

Scenario 1: Brute Force

Primeiro, é pedido para usarmos o snort em modo sniffer para achar a origem do ataque, serviço, e porta. Depois disso, escreveremos uma regra IPS para parar o ataque de força bruta. Caso o ataque seja impedido por 1 minuto, uma flag será recebida no desktop.

Vamos tentar o sniffer com o modo -X para recebermos todo o pacote

Comando 1: `sudo snort -X -v -l .`

```
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
05/07-20:51:05.103913 10.10.140.29:22 -> 10.10.245.36:46466
05/07-20:51:05.114473 10.10.245.36:46466 -> 10.10.140.29:22
05/07-20:51:05.268396 10.10.245.36:46466 -> 10.10.140.29:22
05/07-20:51:05.278957 10.10.140.29:22 -> 10.10.245.36:46466
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
05/07-20:51:05.334309 10.10.140.29:22 -> 10.10.245.36:46458
05/07-20:51:05.513839 10.10.140.29:22 -> 10.10.245.36:46466
05/07-20:51:05.526952 10.10.245.36:46458 -> 10.10.140.29:22
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
05/07-20:51:05.545828 10.10.245.36:46458 -> 10.10.140.29:22
05/07-20:51:05.556388 10.10.140.29:22 -> 10.10.245.36:46458
05/07-20:51:05.566948 10.10.140.29:22 -> 10.10.245.36:46458
05/07-20:51:05.746487 10.10.140.29:22 -> 10.10.245.36:46456
05/07-20:51:05.765359 10.10.140.29:22 -> 10.10.245.36:46458
05/07-20:51:05.783590 10.10.245.36:46466 -> 10.10.140.29:22
05/07-20:51:05.794148 10.10.140.29:22 -> 10.10.245.36:46466
05/07-20:51:05.812387 10.10.245.36:46456 -> 10.10.140.29:22
```

Figura 1 - Pacotes capturados pelo comando 1

Na Figura 1, observamos que o IP 10.10.140.29, está recebendo diversos pacotes na porta 22 de um mesmo IP que está fazendo milhares de requests.

Com isso podemos assumir que o atacante é 10.10.245.36, está atacando nosso serviço SSH pela porta 22.

Agora que temos essas informações, vamos tentar parar este ataque com regras IPS.

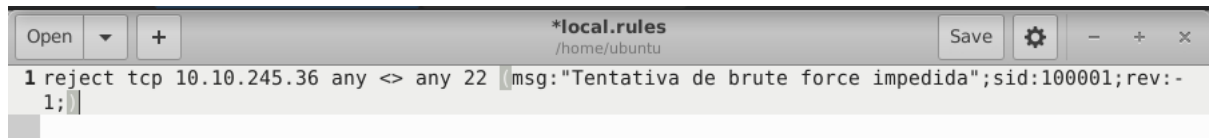


Figura 2 - Regra para impedir comunicação com o IP do atacante

O jeito mais simples seria bloquear qualquer pacote tentando conexão SSH oriundo do IP deste atacante. Testando com o modo console abaixo, podemos ver que ele está de fato funcionando.

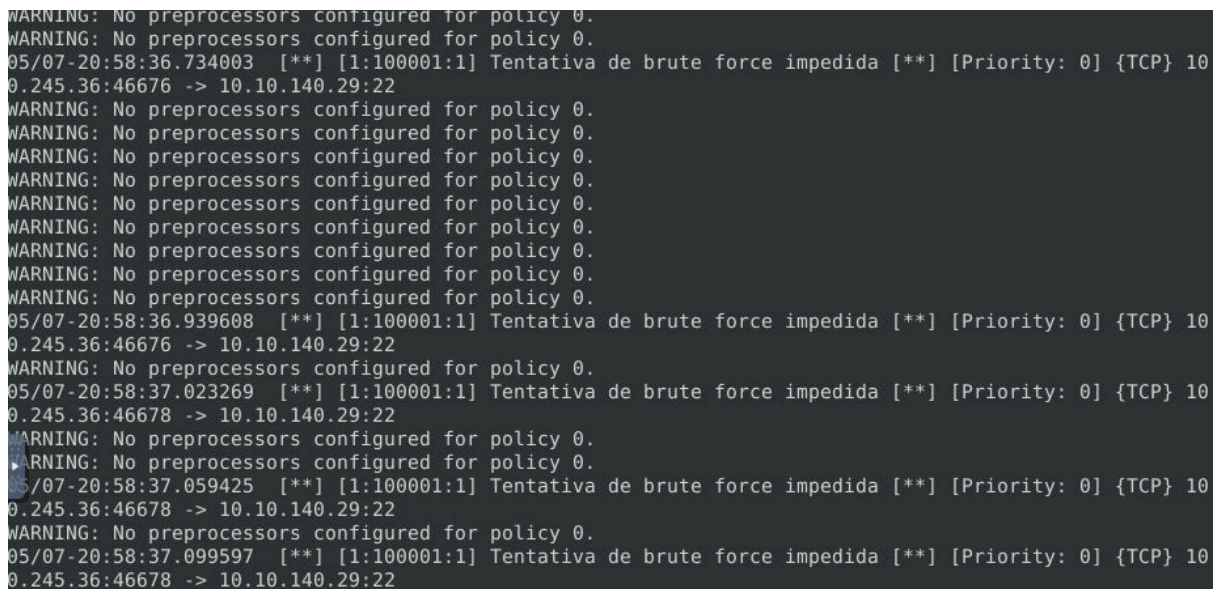


Figura 3 - Recorte do output do modo console

Seguros de que nossa regra é efetiva, agora é necessário colocá-la em prática, executando o snort em modo full para impedir o ataque completamente.

Comando 2: **sudo snort -c \$REGRA -q -Q --daq afpacket -i eth0:eth1 -A full**

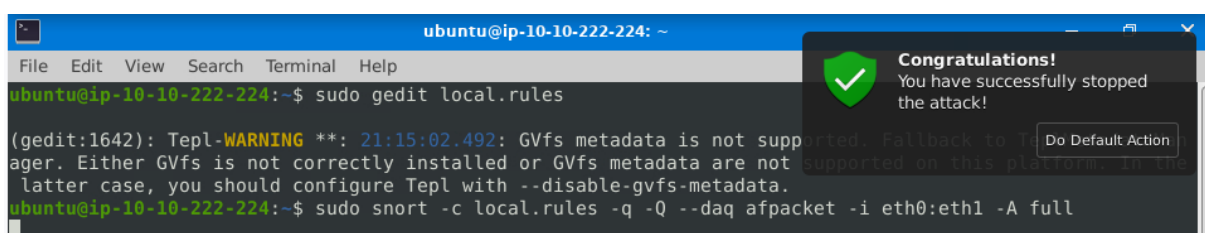


Figura 4 - Alerta gerado pelo Comando 2, com mensagem de sucesso

A regra foi bem sucedida e recebemos o arquivo de flag no Desktop.

```
ubuntu@ip-10-10-113-27:~$ cd Desktop/  
ubuntu@ip-10-10-113-27:~/Desktop$ ls  
flag.txt  
ubuntu@ip-10-10-113-27:~/Desktop$ sudo cat flag.txt  
THM{81b7fef657f8aaa6e4e200d616738254}ubuntu@ip-10-10-113-27:~/Desktop$
```

Figura 5 - Flag 1

Flag: THM{81b7fef657f8aaa6e4e200d616738254}

Qual o nome do serviço sofrendo o ataque?

R: SSH

Qual o protocolo/porta usados no ataque?

R: TCP/22

Scenario 2: Reverse Shell

Primeiro, é pedido para usarmos o snort em modo sniffer para achar a origem do ataque, serviço, e porta. Depois disso, escreveremos uma regra IPS para parar o ataque de força bruta. Caso o ataque seja impedido de vez, uma flag será recebida no desktop.

Desta vez, sabemos que é um reverse shell, portanto o tráfego para fora da máquina deve ser impedido. Vamos utilizar o Comando 1 novamente para observar os pacotes na rede. Vasculhando os logs, a porta 4444 chama atenção.

```
WARNING: No preprocessors configured for policy 0.  
05/07-21:22:01.387521 10.10.196.55:54242 -> 10.10.144.156:4444  
05/07-21:22:01.387531 10.10.144.156:4444 -> 10.10.196.55:54242  
05/07-21:22:01.407530 10.10.196.55:54242 -> 10.10.144.156:4444  
05/07-21:22:01.419540 10.10.144.156:4444 -> 10.10.196.55:54242  
05/07-21:22:01.440167 10.10.196.55:54242 -> 10.10.144.156:4444  
05/07-21:22:01.459577 10.10.144.156:4444 -> 10.10.196.55:54242  
05/07-21:22:01.461290 10.10.196.55:54242 -> 10.10.144.156:4444  
WARNING: No preprocessors configured for policy 0.  
WARNING: No preprocessors configured for policy 0.  
WARNING: No preprocessors configured for policy 0.  
WARNING: No preprocessors configured for policy 0.  
05/07-21:22:01.471843 10.10.144.156:4444 -> 10.10.196.55:54242  
05/07-21:22:01.491674 10.10.196.55:54242 -> 10.10.144.156:4444  
05/07-21:22:01.494245 10.10.196.55:54306 -> 10.10.144.156:4444  
05/07-21:22:01.511811 10.10.144.156:4444 -> 10.10.196.55:54306  
05/07-21:22:01.516669 10.10.196.55:54306 -> 10.10.144.156:4444
```

Figura 6 - Recorte dos logs procurando porta 4444

Podemos observar o mesmo IP usando portas diferentes para se comunicar com o 10.10.144.156 na porta 4444. Vamos vasculhar ainda mais os logs, desta vez utilizando um grep procurando o IP 10.10.144.156:

```
05/07-21:22:01.151059 10.10.144.156:4444 -> 10.10.196.55:54172
05/07-21:22:01.151535 10.10.196.55:54172 -> 10.10.144.156:4444
05/07-21:22:01.183183 10.10.196.55:54172 -> 10.10.144.156:4444
05/07-21:22:01.185777 10.10.144.156:4444 -> 10.10.196.55:54172
05/07-21:22:01.199166 10.10.196.55:54172 -> 10.10.144.156:4444
05/07-21:22:01.211181 10.10.144.156:4444 -> 10.10.196.55:54172
05/07-21:22:01.231207 10.10.196.55:54172 -> 10.10.144.156:4444
05/07-21:22:01.251215 10.10.144.156:4444 -> 10.10.196.55:54172
05/07-21:22:01.251843 10.10.144.156:4444 -> 10.10.196.55:54172
05/07-21:22:01.271281 10.10.196.55:54242 -> 10.10.144.156:4444
05/07-21:22:01.275524 10.10.144.156:4444 -> 10.10.196.55:54242
05/07-21:22:01.291360 10.10.196.55:54242 -> 10.10.144.156:4444
05/07-21:22:01.309764 10.10.196.55:54242 -> 10.10.144.156:4444
05/07-21:22:01.323462 10.10.144.156:4444 -> 10.10.196.55:54242
05/07-21:22:01.331364 10.10.144.156:4444 -> 10.10.196.55:54242
05/07-21:22:01.351493 10.10.196.55:54242 -> 10.10.144.156:4444
05/07-21:22:01.352804 10.10.196.55:54242 -> 10.10.144.156:4444
05/07-21:22:01.367525 10.10.144.156:4444 -> 10.10.196.55:54242
05/07-21:22:01.387521 10.10.196.55:54242 -> 10.10.144.156:4444
05/07-21:22:01.387531 10.10.144.156:4444 -> 10.10.196.55:54242
05/07-21:22:01.407530 10.10.196.55:54242 -> 10.10.144.156:4444
05/07-21:22:01.419540 10.10.144.156:4444 -> 10.10.196.55:54242
05/07-21:22:01.440167 10.10.196.55:54242 -> 10.10.144.156:4444
05/07-21:22:01.459577 10.10.144.156:4444 -> 10.10.196.55:54242
05/07-21:22:01.461290 10.10.196.55:54242 -> 10.10.144.156:4444
05/07-21:22:01.471843 10.10.144.156:4444 -> 10.10.196.55:54242
05/07-21:22:01.491674 10.10.196.55:54242 -> 10.10.144.156:4444
05/07-21:22:01.494245 10.10.196.55:54306 -> 10.10.144.156:4444
05/07-21:22:01.511811 10.10.144.156:4444 -> 10.10.196.55:54306
05/07-21:22:01.516669 10.10.196.55:54306 -> 10.10.144.156:4444
05/07-21:22:01.547934 10.10.196.55:54306 -> 10.10.144.156:4444
05/07-21:22:01.550885 10.10.144.156:4444 -> 10.10.196.55:54306
05/07-21:22:01.568002 10.10.144.156:4444 -> 10.10.196.55:54306
```

Figura 7 - Recorte dos logs procurando o IP 10.10.144.156

Todas as comunicações com esse IP são pela porta 4444, e vem de um mesmo IP com portas diferentes. Isto é um forte indicativo de que encontramos nosso reverse shell. o IP do atacante é 10.10.196.55, a porta de ataque é 4444, que é normalmente usado pelo metasploit (<https://www.speedguide.net/port.php?port=4444>).

Com tudo isso em consideração, é hora de escrever nossa regra de IPS.

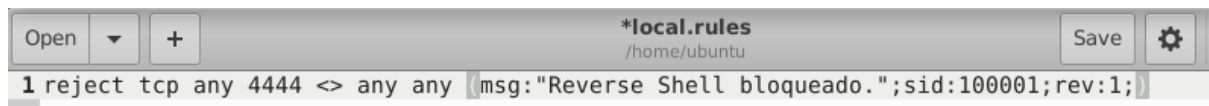


Figura 8 - Regra contra conexões saindo da porta 4444

Agora testando nossa regra com o console, vemos que ela captura parte do tráfego.

```
WARNING: No preprocessors configured for policy 0.
05/07-21:44:37.698408  [**] [1:100001:1] Reverse Shell bloqueado. [**] [Priority: 0] {TCP} 10.10.144.150
:4444 -> 10.10.196.55:54116
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
05/07-21:44:37.746626  [**] [1:100001:1] Reverse Shell bloqueado. [**] [Priority: 0] {TCP} 10.10.144.150
:4444 -> 10.10.196.55:54116
WARNING: No preprocessors configured for policy 0.
05/07-21:44:37.747201  [**] [1:100001:1] Reverse Shell bloqueado. [**] [Priority: 0] {TCP} 10.10.144.150
:4444 -> 10.10.196.55:54116
WARNING: No preprocessors configured for policy 0.
05/07-21:44:37.782796  [**] [1:100001:1] Reverse Shell bloqueado. [**] [Priority: 0] {TCP} 10.10.144.150
:4444 -> 10.10.196.55:54116
WARNING: No preprocessors configured for policy 0.
05/07-21:44:37.802812  [**] [1:100001:1] Reverse Shell bloqueado. [**] [Priority: 0] {TCP} 10.10.144.150
:4444 -> 10.10.196.55:54116
WARNING: No preprocessors configured for policy 0.
05/07-21:44:37.842967  [**] [1:100001:1] Reverse Shell bloqueado. [**] [Priority: 0] {TCP} 10.10.144.150
:4444 -> 10.10.196.55:54116
```

Figura 9 - Recorte do output do modo console 2

Utilizando o Comando 2, vamos colocar a regra em prática.

```
ubuntu@ip-10-10-111-208: ~  
File Edit View Search Terminal Help  
GRE IP6 Ext: 0 ( 0.000%)  
GRE PPTP: 0 ( 0.000%)  
GRE ARP: 0 ( 0.000%)  
GRE IPX: 0 ( 0.000%)  
GRE Loop: 0 ( 0.000%)  
MPLS: 0 ( 0.000%)  
ARP: 0 ( 0.000%)  
IPX: 0 ( 0.000%)  
Eth Loop: 0 ( 0.000%)  
Eth Disc: 0 ( 0.000%)  
IP4 Disc: 536 ( 15.723%)  
IP6 Disc: 0 ( 0.000%)  
TCP Disc: 0 ( 0.000%)  
UDP Disc: 0 ( 0.000%)  
ICMP Disc: 0 ( 0.000%)  
All Discard: 536 ( 15.723%)  
Other: 0 ( 0.000%)  
Bad Chk Sum: 820 ( 24.054%)  
Bad TTL: 0 ( 0.000%)  
S5 G 1: 0 ( 0.000%)  
S5 G 2: 0 ( 0.000%)  
Total: 3409  
=====
```

Action Stats:		
Alerts:	720	(21.121%)
Logged:	720	(21.121%)
Passed:	0	(0.000%)

```
Limits:  
Match: 0  
Queue: 0  
Log: 0  
Event: 0  
Alert: 0  
Verdicts:  
Allow: 3409 ( 98.955%)  
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-10-10-111-208:~$ sudo snort -c local.rules -q -Q --daq afpacket -i eth0:eth1 -A full
```


**Congratulations!**
You have successfully stopped the attack!
Do Default Action

Figura 10 - Segundo alerta gerado pelo Comando 2, com mensagem de sucesso

Mais uma regra bem sucedida, mais uma regra no desktop.

```
ubuntu@ip-10-10-111-208:~$ cd Desktop/  
ubuntu@ip-10-10-111-208:~/Desktop$ sudo cat flag.txt  
THM{0ead8c494861079b1b74ec2380d2cd24}ubuntu@ip-10-10-111-208:~/Desktop$
```

Figura 11 - Flag 2

R: THM{0ead8c494861079b1b74ec2380d2cd24}

Qual o protocolo/porta usados no ataque?

R: TCP/4444

Qual ferramenta é associada com esse número de porta específico?

R: metasploit