



Modulo 2 – defesa e monitoramento

by: Danilo Gomes

Passos seguidos

1. Introdução

2. Nmap

3. SQLI + XSS

4. SQLI + XSS 403

5. Duzzle XSS - SQLI

6. Conclusão



Introdução

Segurança em sistemas, infraestrutura, cloud e resposta a incidentes.

Nmap

é para analisar a rede, e verificar quantas maquinas estão conectadas na rede... Retornou 1 maquina conectada

```
root@13274796db67: /

Session Actions Edit View Help
=> => sha256:e7337afcd762777aeb9c2f935c997aa0ce0b6cb7f7ab03f3bb4cfb2b4363343a 2.87kB / 2.87kB 0.0s
=> => sha256:95ad21363392ccdb56c3bb68dcc21628d6289a33616a8d66b740605891c15862 52.71MB / 52.71MB 1.2s
=> => sha256:ebd1cd63dfadb5a11ba51944d06d36965361642653d1644a5120a8307113425d 964B / 964B 0.0s
=> => sha256:fd8a53c7a8ac2d450966dfae9f57cbc3972bc93c4909e99e0c5309a8f3514a02 429B / 429B 0.0s
=> => extracting sha256:95ad21363392ccdb56c3bb68dcc21628d6289a33616a8d66b740605891c15862 2.3s
=> [kali_lab35 2/2] RUN apt-get update && DEBIAN_FRONTEND=noninteractive apt-get install -y 16.7s
=> [kali_lab35] exporting to image 0.6s
=> => exporting layers 0.6s
=> => writing image sha256:cb6002d6d141acf26f9bce80b6ea605218be0913f1be96a8e7e754ae7fd681d0 0.0s
=> => naming to docker.io/library/labs-kali_lab35 0.0s
=> [kali_lab35] resolving provenance for metadata file 0.0s
[+] Running 6/6
✓ kali_lab35 Built 0.0s
✓ Network labs_labnet35 Created 0.1s
✓ Container dozzle Started 0.6s
✓ Container dvwa Started 0.6s
✓ Container kali_lab35 Started 0.6s
✓ Container waf_modsec Started 0.9s

(kali@kali)-[~/../modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs]
$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS
PORTS         NAMES
f39af4e2cc63   owasp/modsecurity-crs:nginx-alpine  "/docker-entrypoint..." 9 minutes ago  Up 9 minutes
(healthy)    0.0.0.0:8080->8080/tcp, :::8080->8080/tcp  waf_modsec
c007dbaa1049   amir20/dozzle:latest               "/dozzle"                9 minutes ago  Up 9 minutes
0.0.0.0:9999->8080/tcp, [::]:9999->8080/tcp  dozzle
3613e5610f2e   vulnerables/web-dvwa                "/main.sh"               9 minutes ago  Up 9 minutes
80/tcp        dvwa
13274796db67   labs-kali_lab35                     "/bin/bash"              9 minutes ago  Up 9 minutes
kali_lab35

(kali@kali)-[~/../modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs]
$ curl -s http://localhost:8080 | head -5

(kali@kali)-[~/../modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs]
$ docker exec -it kali_lab35 /bin/bash
root@13274796db67: [/]
# nmap -sS -sV waf_modsec
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-17 22:58 UTC
Nmap scan report for waf_modsec (192.168.35.30)
Host is up (0.000010s latency).
rDNS record for 192.168.35.30: waf_modsec.labs_labnet35
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
8080/tcp  open  http    nginx
8443/tcp  open  ssl/http nginx
MAC Address: 02:42:C0:A8:23:1E (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.43 seconds

root@13274796db67: [/]
#
```

SQLi+ XSS

1. verifica, se esta ocorrendo algum ataque. depois aparece o status no log do duzzle.
2. Foi feito um teste de ataque, mas não foi bloqueado por isso retornou o status 302

```
(kali㉿kali)-[~/.../modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs]
$ docker exec kali_lab35 curl -s "http://waf_modsec:8080/vulnerabilities/sqli/?id=1'+OR+'1'='1'--+&Submit=Submit" \
-H "Host: dvwa" \
-H "Cookie: PHPSESSID=test; security=low" \
-w "Status: %{http_code}\n"
Status: 302

(kali㉿kali)-[~/.../modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs]
$ docker exec kali_lab35 curl -s "http://waf_modsec:8080/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%22XSS%22%29%3C/script%3E" \
-H "Host: dvwa" \
-H "Cookie: security=low" \
-w "Status: %{http_code}\n"
Status: 302

(kali㉿kali)-[~/.../modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs]
$
```


SQLi + XSS 403

1. verifica, se ainda está ocorrendo algum ataque. foi feito a verificação no status do log do duzzle.

2. Foi feito um teste de ataque novamente, porém retorna o status 403 dessa vez, que significa que o ataque foi bloqueado.

```
kali@kali: ~/Downloads/formacao-cybersec-main/modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs
Session Actions Edit View Help
(kali@kali)-[~/.../modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs]
$ docker compose up -d --force-recreate waf_modsec
WARN[0000] /home/kali/Downloads/formacao-cybersec-main/modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 2/2
  ✓ Container dvwa      Running      0.0s
  ✓ Container waf_modsec Started    0.7s

(kali@kali)-[~/.../modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs]
$ docker exec kali_lab35 curl -s "http://waf_modsec:8080/vulnerabilities/sqli/?id=1'+OR+'1'='1'--+&Submit=Submit" \
-H "Host: dvwa" \
-H "Cookie: PHPSESSID=test; security=low" \
-w "Status: %{http_code}\n"
<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>nginx</center>
</body>
</html>
Status: 403

(kali@kali)-[~/.../modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs]
$ docker exec kali_lab35 curl -s "http://waf_modsec:8080/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%22XSS%22%29%3C/script%3E" \
-H "Host: dvwa" \
-H "Cookie: security=low" \
-w "Status: %{http_code}\n"
<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>nginx</center>
</body>
</html>
Status: 403

(kali@kali)-[~/.../modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs]
$
```

Duzzle xss / sqli

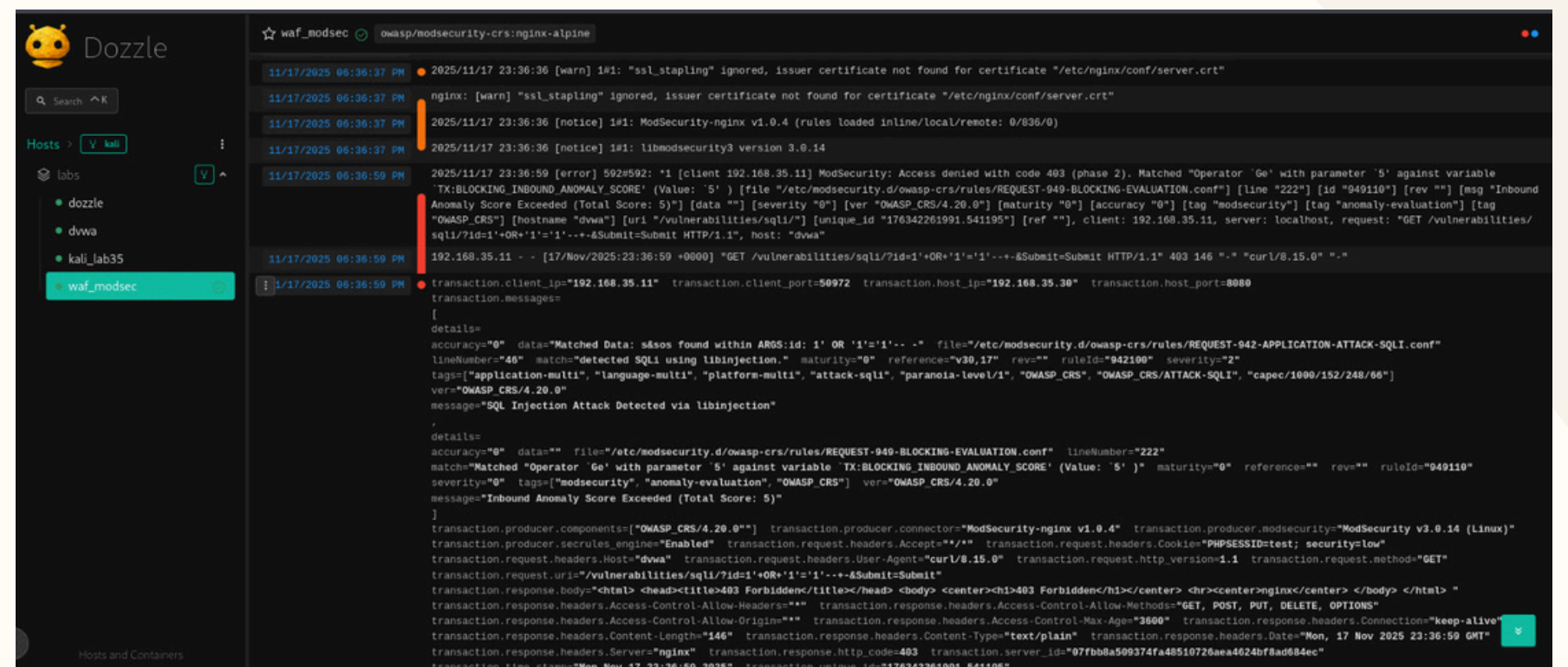
resultado 1



The screenshot shows the Duzzle interface with a sidebar on the left containing a search bar and a list of hosts: dozzle, dvwa, kali_lab35, and waf_modsec. The main panel displays a log for the waf_modsec container. The log shows a 403 status code response from ModSecurity, indicating an XSS attack was detected. The log entry includes details such as the client IP (192.168.35.11), the request path (/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%22XSS%22%3C/script%3E HTTP/1.1), and the response body (403 Forbidden). The log also shows the transaction details, including the client IP, client port, host IP, and host port.

Este é o log duzzle com a mensagem 403, que informa o bloqueio do ataque - XSS

resultado 2

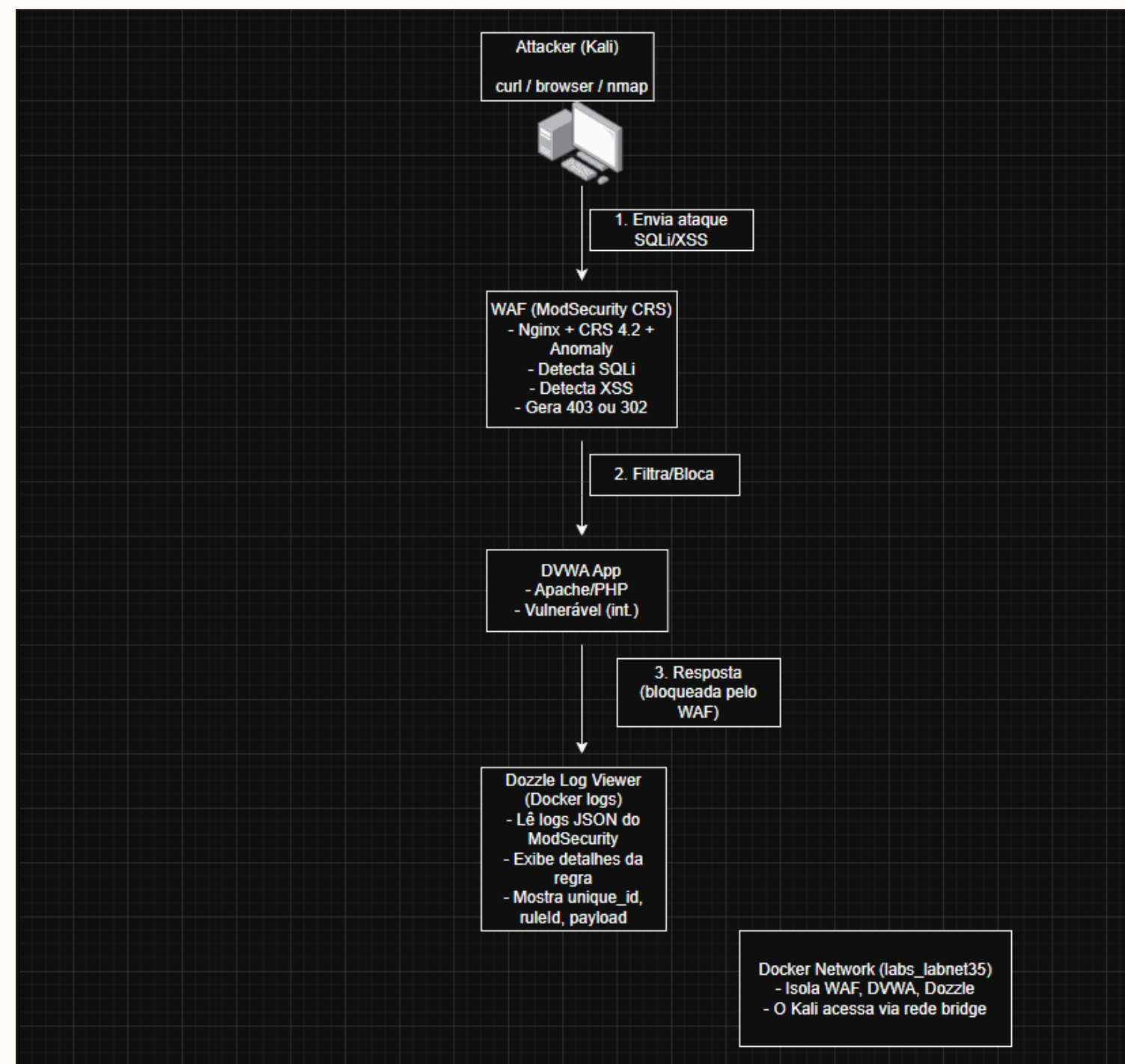


The screenshot shows the Duzzle interface with a sidebar on the left containing a search bar and a list of hosts: dozzle, dvwa, kali_lab35, and waf_modsec. The main panel displays a log for the waf_modsec container. The log shows a 403 status code response from ModSecurity, indicating an SQLi attack was detected. The log entry includes details such as the client IP (192.168.35.11), the request path (/vulnerabilities/sql/7?id=1'OR'1'='1'---&Submit=Submit HTTP/1.1), and the response body (403 Forbidden). The log also shows the transaction details, including the client IP, client port, host IP, and host port.

A mesma informação porém - SQLi

Arquitetura (Diagrama)

diagrama ASCII claro e completo
mostrando o fluxo Attacker → WAF
→ DVWA → Logs, exatamente como
ocorre no seu ambiente.



Recomendações

80/20

1. Reduzir o Paranoia Level do OWASP CRS para PL1

Diminui falsos positivos e mantém proteção essencial. É só alterar uma linha no arquivo de configuração.

2. Ajustar o Anomaly Score Threshold (ex.: de 5 para 10)

Evita bloqueios desnecessários e mantém detecção eficaz. Impacto alto com mudança mínima.

3. Centralizar os logs do WAF (ex.: enviar para Dozzle/ELK)

Facilita análise, auditoria e monitoramento. Aproveita infraestrutura já existente.

4. Isolar containers em redes Docker dedicadas

Reduz superfícies internas e limita movimentos laterais. Simples de aplicar com docker network.

5. Criar regras de exceção básicas (whitelists por rota)

Permite que rotas conhecidas passem sem acionar o CRS. Evita ruído e melhora a precisão do WAF.

Resposta a Incidente (NIST IR)

Detecção

O WAF identificou e alertou tentativas de SQLi e XSS, registrando tudo nos logs.

Contenção

As requisições maliciosas foram imediatamente bloqueadas com código 403.

Erradicação

Não houve comprometimento; apenas revisão das regras do WAF para evitar falsos positivos.

Recuperação

Serviços continuaram íntegros; apenas confirmação do funcionamento normal do ambiente.

Lições Aprendidas

Reforçar uso do WAF, centralizar logs e ajustar configurações para melhorar precisão e proteção.

Conclusão

Foi feito um reconhecimento do Nmap executado em scan, foi verificado que após configurar um WAF em modo de detecção, foi constatado que estava acontecendo um ataque. Logo em seguida foi feita uma configuração na WAF, para bloquear o ataque.