

Kensei CyberSec / Vai Na Web 2025

Formação CyberSec

Relatório técnico de defesa e monitoramento

Web Application Firewall (**WAF**) e

Damn Vulnerable Web Application (**DVWA**)

Aluna: Dalila Salvatierra - **Data:** 19/09/2025

Versão: Projeto final Modulo 2 - Opção 1 (Hands On)

Sumário:

Este laboratório teve como propósito avaliar a segurança de uma aplicação web vulnerável (**DVWA**) utilizando o **WAF ModSecurity** com as regras do **OWASP CRS**. A intenção principal foi verificar como o sistema consegue identificar e bloquear ataques, além de monitorar em tempo real e aplicar respostas a incidentes seguindo boas práticas de proteção cibernética.

O ambiente foi montado com **containers Docker**, garantindo um espaço isolado e seguro para testes, além de facilitar a configuração. Os containers utilizados incluíram:

- **Kali Linux:** container dedicado a ataques controlados, executando testes de **SQL Injection (SQLi)** e **Cross-Site Scripting (XSS)**.
- **WAF ModSecurity:** firewall de aplicação web, configurado primeiro em **modo Detecção**, para registrar ataques sem bloqueio, e depois em **modo Blocking**, para impedir ações maliciosas.
- **DVWA:** aplicação vulnerável usada como alvo, ajustada para nível de segurança "Low" para permitir testes de exploração.
- **Dozzle:** ferramenta de monitoramento, permitiu a visualização de logs em tempo real, possibilitando acompanhar todas as detecções e bloqueios do WAF.

Durante a bateria de testes, foram realizados dois tipos de ataques principais:

1. **SQL Injection (SQLi):** tentativas de manipular dados do banco de forma não autorizada, enviando comandos maliciosos pelos parâmetros da aplicação.
2. **Cross-Site Scripting (XSS):** inserção de scripts maliciosos que poderiam ser executados no navegador, explorando vulnerabilidades da aplicação.

Resultados obtidos:

- No **modo Detecção**, os ataques foram registrados nos logs do WAF, mas a aplicação ainda respondeu normalmente (HTTP 302), permitindo estudar o comportamento dos ataques sem interrupções.
- No **modo Blocking**, os ataques foram efetivamente bloqueados, retornando HTTP 403, o que demonstra a capacidade do WAF de prevenir ações maliciosas antes que atinjam a aplicação.
- A observação em tempo real com o **Dozzle** permitiu identificar os **Rule IDs 942100 (SQLi) e 941100 (XSS)**, confirmando que as regras do OWASP CRS foram aplicadas corretamente.
- Logs detalhados e capturas de tela foram coletados como evidência do processo de detecção, bloqueio e resposta.

Conclusões:

O laboratório comprovou que um **WAF bem configurado** é eficaz na proteção contra ataques comuns em aplicações web. A experiência demonstrou a importância de manter regras atualizadas, monitorar logs constantemente e responder rapidamente a incidentes. Observou-se também que o **modo Detecção** é útil para análise e aprendizado, enquanto o **modo Blocking** se destacou como essencial em ambientes reais.

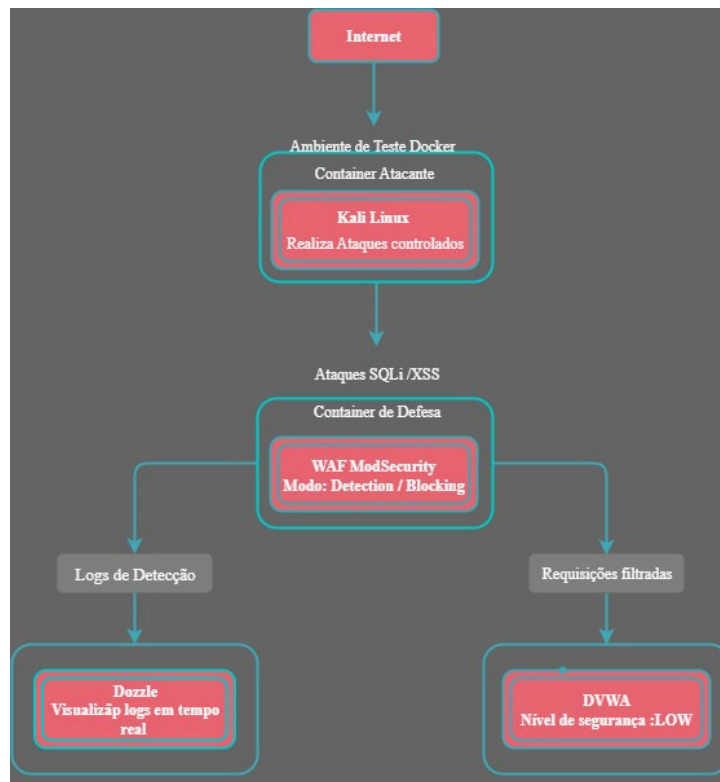
Objetivo e escopo:

Objetivo: Testar eficácia do WAF, monitoramento e resposta a incidentes.

Escopo:

- Protegido: DVWA (nível Low)
- Ataques: SQLi e XSS
- Limite: ambiente isolado em containers Docker

Arquitetura (Diagrama):



Metodologia:

A metodologia utilizada para este teste é baseada no tutorial “Lab de segurança WAF = DVWA” da Formação CyberSec, publicado no GitHub Kensei CyberSec

[formacao-cybersec/modulo2-defesa-monitoramento/projeto-final/opcao...](https://github.com/formacao-cybersec/modulo2-defesa-monitoramento/projeto-final/opcao...)

1. Preparar ambiente Docker e containers.
2. Reconhecimento com **Nmap** para identificar serviços e portas.
3. Testar ataques:
 - a. Modo Detecção → registros em logs
 - b. Modo Blocking → bloqueio efetivo
4. Monitorar logs em Dozzle
5. Coletar evidências (logs, prints)
6. Aplicar resposta a incidentes NIST IR

Execução e Evidências:

- **SQLi e XSS:**
 - Modo Detecção → HTTP 302
 - Modo Blocking → HTTP 403
- **Logs:**
 - logs_waf_evidencias.txt
 - Rule IDs: 942100 (SQLi), 941100 (XSS)
- **Monitoramento:** prints do Dozzle mostrando detecção e bloqueio

Resposta a Incidente (NIST IR)

Fase NIST IR	Descrição/Ação	Exemplos de Logs/Evidência
Detecção	WAF identifica tráfego suspeito ou padrões de ataque, sem bloquear.	2025/09/18 16:28:05 [alert] [client 192.168.35.10] ModSecurity: Warning. Matched XSS Attack via libinjection in ARGS:name TX:ANOMALY_INBOUND_ANOMALY_SCORE: 3
Contenção	Ativação do modo blocking no WAF; ataques passam a ser bloqueados automaticamente.	2025/09/18 16:30:12 [notice] ModSecurity: SecRuleEngine set to "On" Indica que regras de bloqueio estão ativas
Erradicação	Requisições maliciosas são bloqueadas efetivamente pelo WAF.	2025/09/18 16:30:45 [error] [client 192.168.35.10] ModSecurity: Access denied with code 403 (phase 2). Pattern match "SQL Injection Attack" TX:ANOMALY_INBOUND_ANOMALY_SCORE: 8
Recuperação	Garantir que a aplicação continue disponível e monitorar logs para confirmar que ataques não retornaram.	Logs posteriores mostram requisições legítimas sem bloqueios indevidos 2025/09/18 16:35:10 [notice] [client 192.168.35.11] Request processed successfully

- **Detecção:** ataques registrados pelo WAF em modo DetectionOnly
- **Contenção:** ativação do WAF em modo Blocking
- **Erradicação:** ataques interrompidos e aplicação protegida
- **Recuperação:** ambiente mantido operacional
- **Lições Aprendidas:** importância de monitoramento contínuo, revisão de regras e automatização de alertas

Recomendações (80/20)

1. Manter WAF ativo em **modo Blocking** nos ambientes de produção.
2. Revisar regras **OWASP CRS** regularmente.
3. Monitorar logs continuamente.
4. Testar outros tipos de ataques em ambientes seguros.
5. Treinar equipe para resposta rápida.

Conclusão:

A partir dos testes de laboratório realizados foi constatado que o WAF é eficaz contra ataques comuns (SQLi e XSS), que o monitoramento em tempo real é essencial e que a metodologia NIST IR ajuda a organizar a resposta a incidentes.

Anexos:

Configurações Docker/WAF:

Figura 1 - Iniciando containers, verificação de status de containers e teste de conectividade

```
monitoramento/projeto-Final/opcaol-hands-on/labs (main)
$ docker compose up -d --build
time="2025-09-17T14:14:13-03:00" level=warning msg="C:\\Users\\dalis\\formacao-cybersec\\modulo2-defesa-monitoramento\\projeto-final\\opcaol-hands-on\\labs\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
unable to get image 'labs-kali_lab35': error during connect: Get "http://%2F%2F.%2Fpipe%2FdockerDesktopLinuxEngine/v1.50/images/labs-kali_lab35/json": open //pipe/dockerDesktopLinuxEngine: O sistema não pode encontrar o arquivo especificado.

dalis@dalilassr MINGW64 ~/formacao-cybersec/modulo2-defesa-monitoramento/projeto-Final/opcaol-hands-on/labs (main)
$ docker compose up -d --build
time="2025-09-17T14:18:02-03:00" level=warning msg="C:\\Users\\dalis\\formacao-cybersec\\modulo2-defesa-monitoramento\\projeto-final\\opcaol-hands-on\\labs\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Building 2.9s (8/8) FINISHED
=> [internal] load local bake definitions 0.0s
=> => reading from stdin 524B 0.0s
=> [internal] load build definition from Dockerfile.kali 0.0s
=> => transferring dockerfile: 262B 0.0s
=> [internal] load metadata for docker.io/kalilinux/kali-rolling:latest 2.3s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [1/2] FROM docker.io/kalilinux/kali-rolling:latest@sha256:cb291c3a47 0.0s
=> => resolve docker.io/kalilinux/kali-rolling:latest@sha256:cb291c3a47 0.0s
=> CACHED [2/2] RUN apt-get update && DEBIAN_FRONTEND=noninteractive ap 0.0s
=> exporting to image 0.1s
=> => exporting layers 0.0s
=> => exporting manifest sha256:1d0f98dfb982e365bbd5a90ba34ed45d1b26cf4 0.0s
=> => exporting config sha256:da20235c96732abd81ec8da546026ae7872f03538 0.0s
=> => exporting attestation manifest sha256:dd70023f0357c9dc5c04bd0f8b9 0.0s
=> => exporting manifest list sha256:2c73c824d0462522a58b5007d148f12753 0.0s
=> => naming to docker.io/library/labs-kali_lab35:latest 0.0s
=> => unpacking to docker.io/library/labs-kali_lab35:latest 0.0s
=> resolving provenance for metadata file 0.0s
[+] Running 5/5
✔ kali_lab35 Built 0.0s
✔ Container kali_lab35 Started 0.9s
✔ Container dvwa Started 0.5s
✔ Container dozzle Started 0.7s
✔ Container waf_modsec Started 0.4s

dalis@dalilassr MINGW64 ~/formacao-cybersec/modulo2-defesa-monitoramento/projeto-Final/opcaol-hands-on/labs (main)
$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS                               COMMAND                  CR
NAMES
69b950fe2544   labs-kali_lab35                     "/bin/bash"             41 seconds ago  Up 41 seconds  -                                     -                         43
kali_lab35
ef5d6377d018   owasp/modsecurity-crs:nginx-alpine "/docker-entrypoint..." 20 hours ago    Up 41 seconds (healthy)  0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp  waf_modsec
ab7da90980b7   amir20/dozzle:latest                "/dozzle"               20 hours ago    Up 41 seconds          0.0.0.0:9999->8080/tcp, [::]:9999->8080/tcp  dozzle
da6bc0717448   vulnerables/web-dvwa                "/main.sh"              20 hours ago    Up 41 seconds          80/tcp                  dvwa

dalis@dalilassr MINGW64 ~/formacao-cybersec/modulo2-defesa-monitoramento/projeto-Final/opcaol-hands-on/labs (main)
$ curl -s http://localhost:8080 | head -5
```

Figura 2 - Dentro do Container Kali, executar Scan Nmap (identificação do WAF rodando nginx nas portas 8080(HTTP) e 8443(HTTPS)).

```
PS C:\Users\dalis\formacao-cybersec\modulo2-defesa-monitoramento\projeto-final\opcao1-hands-on\labs> docker exec -it kali_lab35 /bin/bash
(root@ 69b950fe2544)-[/]
# nmap -sS -sV waf_modsec
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-17 17:49 UTC
Nmap scan report for waf_modsec (192.168.35.30)
Host is up (0.0000090s 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: C6:67:1B:53:8C:C7 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.66 seconds

(root@ 69b950fe2544)-[/]
# exit
exit
PS C:\Users\dalis\formacao-cybersec\modulo2-defesa-monitoramento\projeto-final\opcao1-hands-on\labs>
```


Configuração WAF para Modo Detecção:

Figura 3 - No arquivo docker-compose.yml, Modo Detecção está assim: **-MODSEC_RULE_ENGINE=DetectionOnly**
modo detecção apenas

```
1 version: "3.9"
2
3 services:
4   kali_lab35:
5     build:
6       context: .
7       dockerfile: Dockerfile.kali
8     container_name: kali_lab35
9     tty: true
10    volumes:
11      - ./scripts:/scripts
12    networks:
13      labnet35:
14        ipv4_address: 192.168.35.11
15
16    waf_modsec:
17      image: owasp/modsecurity-crs:nginx-alpine
18      container_name: waf_modsec
19      environment:
20        - BACKEND=http://dvwa:80 # para onde o WAF faz proxy_pass
21        - SERVER_NAME=localhost # hostname do WAF
22        - MODSEC_RULE_ENGINE=DetectionOnly # modo detecção apenas
23        - MODSEC_RULE_ENGINE-On # modo blocking (bloqueia ataques)
24        - MODSEC_RULE_ENGINE=DetectionOnly
25        - MODSEC_RULE_ENGINE=DetectionOnly # Modos de paranoia (afinam detecção/bloqueio):
26        - BLOCKING_PARANOIA=1
27        - DETECTION_PARANOIA=1
28      depends_on:
29        - dvwa
30      ports:
31        - "8080:8080" # atente: 8080:8080 (default da imagem OWASP)
32      networks:
33        labnet35:
34          ipv4_address: 192.168.35.30
35
36    dvwa:
37      image: vulnerabilities/web-dvwa
38      container_name: dvwa
39      networks:
40        labnet35:
41          ipv4_address: 192.168.35.40
42
43    dozzle:
44      image: amir20/dozzle:latest
45      container_name: dozzle
46      ports:
47        - "9999:8080"
48      environment:
49        - DOZZLE_USERNAME=admin
50        - DOZZLE_PASSWORD=admin
51      volumes:
52        - /var/run/docker.sock:/var/run/docker.sock:ro
53      networks:
54        labnet35:
55          ipv4_address: 192.168.35.50
56
57    networks:
58      labnet35:
```

Figura 4 - Container WAF

```

dalis@dalilassr MINGW64 ~/formacao-cybersec/modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs (main)
$ docker compose up -d --force-recreate waf_modsec
time="2025-09-17T15:23:51-03:00" level=warning msg="C:\\Users\\dalis\\formacao-cybersec\\modulo2-defesa-monitoramento\\proj
eto-final\\opcao1-hands-on\\labs\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remov
e it to avoid potential confusion"
[+] Running 2/2
 ✓ Container dvwa           Running    0.0s
1'+OR+'1'='1'--+&Submit=Submit" \
-H "Host: dvwa" \
-H "Cookie: PHPSESSID=test; security=low" \
-w "Status: %{http_code}\n"
Status: 302

```

Figura 5 - Ataque SQLi detectado, mas não bloqueado. Status 302

```

dalis@dalilassr MINGW64 ~/formacao-cybersec/modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs (main)
$ docker exec kali_lab35 bash -lc 'while true; do 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 "\nStatus: %{http_code}\n";
sleep 2; done'

Status: 302

```

Figura 6 - Ataque XSS detectado, mas não bloqueado. Status 302

```

dalis@dalilassr MINGW64 ~/formacao-cybersec/modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs (main)
$ docker exec kali_lab35 curl -s "http://waf_modsec:8080/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%22XSS%22%29%3C/sc
ript%3E" \
-H "Host: dvwa" \
-H "Cookie: security=low" \
-w "Status: %{http_code}\n"
Status: 302

dalis@dalilassr MINGW64 ~/formacao-cybersec/modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs (main)
$ █

```

Configuração WAF para Modo Blocking:

Figura 7 – No arquivo docker-compose.yml, alteração para

Modo Blocking:- **MODSEC_RULE_ENGINE = On**

#modo blocking (bloqueia ataques)

```
1 version: '3.9'
2
3 services:
4   kali_lab35:
5     build:
6       context: .
7       dockerfile: Dockerfile.kali
8     container_name: kali_lab35
9     tty: true
10    volumes:
11      - ./scripts:/scripts
12    networks:
13      labnet35:
14        ipv4_address: 192.168.35.11
15
16  waf_modsec:
17    image: owasp/modsecurity-crs:nginx-alpine
18    container_name: waf_modsec
19    environment:
20      - BACKEND=http://dvwa:80 # para onde o WAF faz proxy_pass
21      - SERVER_NAME=localhost # hostname do WAF
22      - MODSEC_RULE_ENGINE=On # modo blocking (bloqueia ataques)
23      # - MODSEC_RULE_ENGINE=DetectionOnly
24      # Níveis de paranoia (afinam detecção/bloqueio):
25      - BLOCKING_PARANOIA=1
26      - DETECTION_PARANOIA=1
27    depends_on:
28      - dvwa
29    ports:
30      - "8888:8888" # atenção: 8888:8888 (default da imagem OWASP)
31    networks:
32      labnet35:
33        ipv4_address: 192.168.35.38
34
35  dvwa:
36    image: vulnerables/web-dvwa
37    container_name: dvwa
38    networks:
39      labnet35:
40        ipv4_address: 192.168.35.40
41
42  dozzle:
43    image: amir20/dozzle:latest
44    container_name: dozzle
45    ports:
46      - "9999:8888"
47    environment:
48      - DOZZLE_USERNAME=admin
49      - DOZZLE_PASSWORD=admin
50    volumes:
51      - /var/run/docker.sock:/var/run/docker.sock:ro
```

Figura 8 - Recriando container WAF

```
PS C:\Users\dalis\formacao-cybersec\modulo2-defesa-monitoramento\projeto-final> docker compose up -d --force-recreate waf_modsec
time="2025-09-17T15:06:00-03:00" level=warning msg="C:\\Users\\dalis\\formacao-cybersec\\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    1.5s
PS C:\Users\dalis\formacao-cybersec\modulo2-defesa-monitoramento\projeto-final\opcao1-hands-on\labs> _
```

Figura 9 - Ataque SQLi Bloqueado. Status 403 + página "403 Forbidden"

```
dalis@dalilassr MINGW64 ~/formacao-cybersec/modulo2-defesa-monitoramento/projeto-final/opcao1-hands-on/labs (main)
$ 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
```

Figura 10 - Ataque XSS Bloqueado. Status 403 + página "403 Forbidden"

```
$ 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
```

Interface Dozzle

Logs do WAF, container “waf-modsec”:

Figura 11 - Nginx e ModSecurity foram iniciados corretamente.

```
8/09/2025 16:26:07 /docker-entrypoint.sh: Configuration complete; ready for start up
18/09/2025 16:26:07 2025/09/18 19:26:07 [warn] 1#1: "ssl_stapling" ignored, issuer certificate not found for certificate "/etc/nginx/conf/server.crt"
18/09/2025 16:26:07 nginx: [warn] "ssl_stapling" ignored, issuer certificate not found for certificate "/etc/nginx/conf/server.crt"
18/09/2025 16:26:07 2025/09/18 19:26:07 [notice] 1#1: ModSecurity-nginx v1.0.4 (rules loaded inline/local/remote: 0/836/0)
```

Figura 12 - Acesso normal - Detecção sem bloqueio

```
2025 16:27:35 192.168.35.1 - - [18/Sep/2025:19:27:35 +0000] "GET /login.php HTTP/1.1" 200 699 "-" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
```

Figura 13 - SQL Injection (SQLi) - Bloqueado

```
18/09/2025 16:36:38 2025/09/18 19:36:38 [error] 583#583: *25 [client 192.168.35.11] ModSecurity: Access denied with code 403 (phase 2). Matched "Operator 'Ge' with parameter '5' against variable 'TX:BLOCKING_INBOUND_ANOMALY_SCORE' (Value: '5' ) [file "/etc/modsecurity.d/owasp-crs/rules/REQUEST-949-BLOCKING-EVALUATION.conf"] [line "222"] [id "949110"] [rev "" ] [msg "Inbound Anomaly Score Exceeded (Total Score: 5)"] [data "" ] [severity "0"] [ver "OWASP_CRS/4.17.1"] [maturity "0"] [accuracy "0"] [tag "modsecurity"] [tag "anomaly-evaluation"] [tag "OWASP_CRS"] [hostname "waf_modsec"] [uri "/vulnerabilities/sqli/"] [unique_id "175822419870.716990"] [ref ""], client: 192.168.35.11, server: localhost, request: "GET /vulnerabilities/sqli/?id=1'+OR+'1'='1'--+&-&Submit=Submit HTTP/1.1", host: "waf_modsec:8080"
```

Figura 14 - XSS (Cross-Site Scripting)- Bloqueado

```
18/09/2025 16:50:03 2025/09/18 19:50:03 [error] 584#584: *53 [client 192.168.35.11] ModSecurity: Access denied with code 403 (phase 2). Matched "Operator 'Ge' with parameter '5' against variable 'TX:BLOCKING_INBOUND_ANOMALY_SCORE' (Value: '20' ) [file "/etc/modsecurity.d/owasp-crs/rules/REQUEST-949-BLOCKING-EVALUATION.conf"] [line "222"] [id "949110"] [rev "" ] [msg "Inbound Anomaly Score Exceeded (Total Score: 20)"] [data "" ] [severity "0"] [ver "OWASP_CRS/4.17.1"] [maturity "0"] [accuracy "0"] [tag "modsecurity"] [tag "anomaly-evaluation"] [tag "OWASP_CRS"] [hostname "dwaa"] [uri "/vulnerabilities/xss_r/"] [unique_id "175822500380.300904"] [ref ""], client: 192.168.35.11, server: localhost, request: "GET /vulnerabilities/xss_r/?name=X3CscriptX3EalertX38X22X55X22X329X3C/scriptX3E HTTP/1.1", host: "dwaa"
```

Figura 15 - Nmap: nmap_waf_evidencias.txt

```
(root@69b950fe2544)-[/]
# cat nmap_waf_evidencias.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-17 19:55 UTC
Nmap scan report for waf_modsec (192.168.35.30)
Host is up (0.000017s 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: EE:C5:B6:6A:94:1D (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

