RELATÓRIO DO LAB DE SEGURANÇA WAF + DVWA

Título: Simulação de Ataques Controlados contra DVWA com Proteção WAF

Data: 24/09/2025

Versão: 1.0

Responsável: Gonçalo Quissola Dala

1. Sumário Executivo

Este relatorio avaliou a resiliência de uma aplicação vulnerável (DVWA) frente a ataques controlados, utilizando NGINX + ModSecurity + OWASP CRS como defesa.
- Ambiente: DVWA exposta a ataques OWASP Top 10.
- Ataques simulados: nmap, SQLi e XSS.
- Nível de proteção: Modo detecção (302) logou ataques; modo bloqueio (403) os impediu.
- Estado atual: Ambiente protegido com logs estruturados.

Conclusão: O WAF foi eficaz ao detectar e bloquear ataques críticos.

2. Objetivo e Escopo

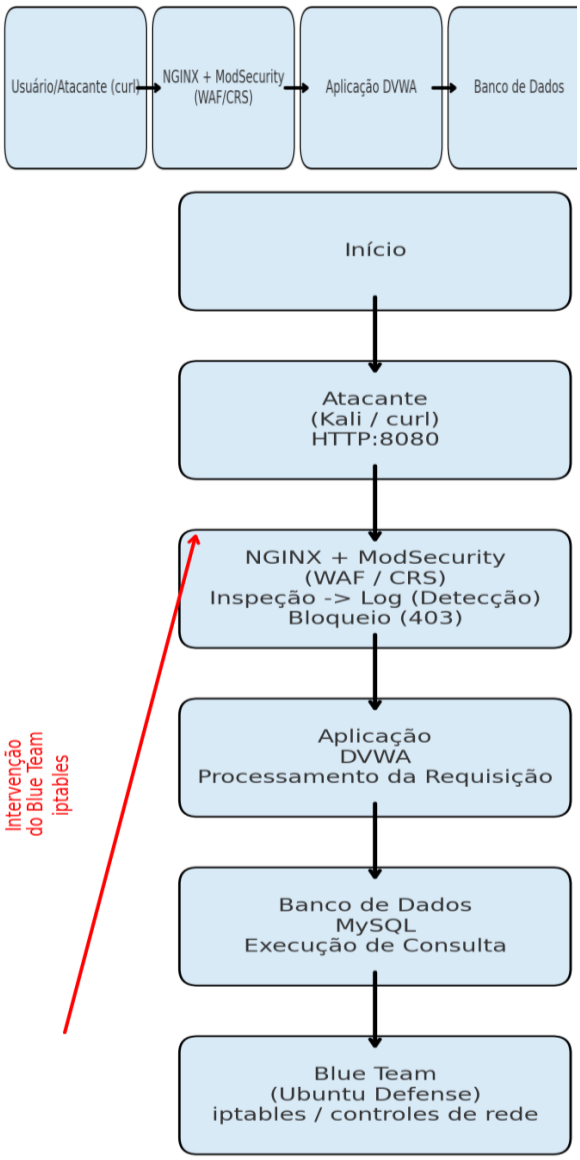
Objetivo:

- Validar a eficácia do WAF na detecção e bloqueio de ataques comuns (SQLi e XSS).
- Demonstrar o ciclo completo de segurança: detecção, análise, contenção e resposta.

Escopo:

- Aplicação alvo: DVWA (http://dvwa).
- Atacado: Endpoints SQLi (/sqli/) e XSS (/xss_r refletido.
- Defendido: DVWA em Linux protegida por WAF.
- Quando: 17/09/2025. Onde: Laboratório controlado.
- Impacto: Risco de vazamento de dados e execução de código
- Ferramentas: ModSecurity v3.0.14, OWASP CRS 4.17.1, Nginx, curl, nmap.
- Limite: Testes realizados apenas em ambiente controlado, com configuração de paranoia nível 1, sem impacto produtivo.

3. Arquitetura (Diagrama)



Descrição:

Camadas:

- **Camada 1**: Cliente (IP: 192.168.35.11) envia requisições maliciosas.
- **Camada 2**: Nginx com ModSecurity atua como WAF, analisando e bloqueando ataques.
- **Camada 3**: Aplicação DVWA (IP: 192.168.35.30:8080) hospeda a aplicação vulnerável.

Fluxo:

1. Requisição HTTP → WAF (análise de regras CRS).
2. Se detectada ameaça → Bloqueio (403) e log estruturado.
3. Se for segura → Encaminhamento para DVWA, seguindo para o banco de dados.

Ativos:
- Servidor Web (NGINX+WAF).
- Aplicação DVWA (PHP/MySQL).
- Banco MySQL.
- Logs estruturados.

4. Metodologia

Passos executados:

1. **Reconhecimento:** Varredura com **nmap** → descoberta de portas abertas e serviços.
2. **Ataques controlados:** Execução de payloads SQLi e XSS.
3. **Detecção:** WAF em modo *Detection Only*, registrando ataques (302).
4. **Bloqueio:** Ativação de modo *Blocking*, interrompendo ataques (403).
5. **Resposta:** Análise, contenção, erradicação e recuperação segundo NIST IR.

Critérios de sucesso:

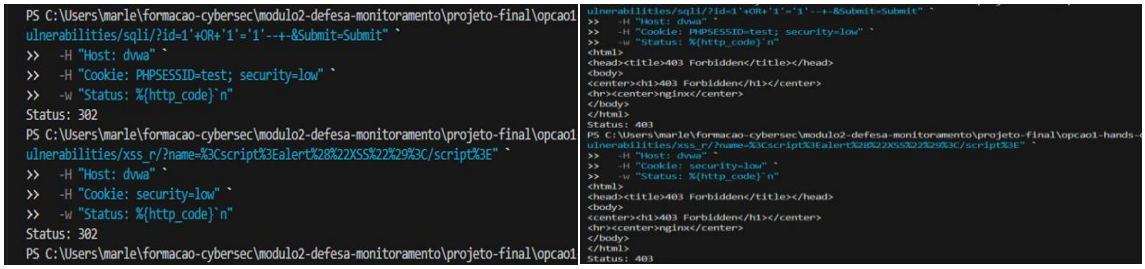
- WAF deve detectar SQLi e XSS via libinjection e regras CRS
- Bloqueio deve retornar HTTP 403 com log detalhado
- Score de anomalia deve exceder o threshold configurado (5 para inbound)

Causa raiz: vulnerabilidades DVWA.
Ferramentas: nmap, curl, ModSecurity, iptables/tcpdump.

5. Execução e Evidências

Reconhecimento (nmap): Serviços identificados.

SQL Injection:
- Modo Detecção (302): Requisição logada.
- Modo Bloqueio (403): Negada, log SQL Injection Detected.
XSS:
- Modo Detecção (302): Requisição logada.
- Modo Bloqueio (403): Negada, regras CRS acionadas.



Notas operacionais e implicações (detecção 302 vs bloqueio 403)

- **Detecção (302)** Vantagem: visibilidade e coleta de inteligência sobre técnicas do atacante sem impactar usuários. Risco: ataques são observados em produção potencialmente permitindo exploração de janelas pequenas; ideal apenas em ambientes de teste ou durante tuning.

- **Bloqueio (403)**

Vantagem: previne exploração em tempo real; reduz risco de comprometimento.

Risco: risco de falsos positivos que podem bloquear usuários legítimos — mitiga-se com tuning (paranoia level, whitelists, rule exclusions).

6. Resposta a Incidente (NIST IR)

- Detecção: SQLi e XSS em logs.
- Análise: Ataques críticos OWASP Top 10.
- Contenção: Modo blocking (403).
- Erradicação: Regras CRS ativas.
- Recuperação: Serviço restaurado.

Timeline NIST IR

Fase	Timestamp	Ação
Detecção	17/Sep/2025 19:53:54	WAF detecta SQLi (regra 942100)
Análise	19:53:54	Identificado SQLi via libinjection
Contenção	19:53:54	Bloqueio imediato (HTTP 403)
Erradicação	19:53:54	Requisição maliciosa descartada
Recuperação	-	Serviço permaneceu estável
Lições	-	Monitorar antes de bloquear reduz falsos positivos, necessário tuning contínuo.

Tabela – Classificação dos Incidentes Ocorridos

Incidente	Data/Hora	Categoria	Severidade	Impacto CIA	Urgência	Priorização
SQLi	17/Sep/2025 19:53:54	CAT-01 Ataques Web application	SEV-3: Médio Ataque bloqueado	Confidencialidade (Médio) Integridade (Médio) Disponibilidade (Baixo)	Baixa Ataque contido Imediatamente	P4
XSS	17/Sep/2025 19:55:38	CAT-01 Ataques Web application	SEV-3: Médio Ataque bloqueado	Integridade (Médio) Confidencialidade (Baixo) Disponibilidade (Baixo)	Baixa Ataque contido Imediatamente	P4

7. Recomendações (80/20)

Ação	Esforço	Impacto	Tipo
Ativar WAF permanentemente em blocking	Médio	Alto	Permanente
Integrar SIEM para correlação de logs (ex.: Splunk, ELK)	Médio	Alto	Permanente
Criar playbooks de resposta rápida	Baixo	Médio	Permanente
Atualizar CRS e ModSecurity	Baixo	Alto	Permanente
Atualizar regras CRS regularmente **Treinar equipe contra falsos positios**	Médio	Médio	Permanente

Próximos passos: dashboards, KPIs de segurança, ampliar para OWASP Top 10

- Testar ataques mais complexos (ex.: blind SQLi, XSS persistente).
- Ajustar regras para reduzir falsos positivos em produção.
- Automatizar resposta a incidentes com integração a ferramentas de orquestração.
- Implementar dashboard de monitoramento para visualização em tempo real.

8. Conclusão

O ambiente demonstrou **maturidade operacional básica** na defesa contra ataques web comuns. O WAF foi eficaz na detecção e bloqueio de SQLi e XSS, com logs detalhados para análise forense. O plano de resposta a incidentes baseado no NIST permitiu classificar adequadamente os eventos ocorridos.

- Pontos fortes: Detecção, bloqueio, logs, logging estruturado.

- A melhorar: tuning contínuo, integração com SIEM.

Anexos

Configs, logs, scripts, prints, políticas.

Configurações (Configs)

WAF (NGINX + ModSecurity + OWASP CRS)

- **Configuração do ModSecurity:** /etc/nginx/modsec/main.conf
- **Regras do OWASP CRS:** /usr/share/modsecurity-crs/rules/*.conf
- **Parâmetros relevantes:**

  SecRuleEngine On

  SecDefaultAction "phase:1,log,auditlog,deny,status:403"

Regras de Firewall (iptables)

Bloquear tráfego suspeito identificado:

iptables -A INPUT -s 192.168.35.11 -j DROP

Permitir apenas portas essenciais:

iptables -A INPUT -p tcp --dport 22 -j ACCEPT

iptables -A INPUT -p tcp --dport 8080 -j ACCEPT

Logs

Logs de Detecção (Detection Only Mode)

- **Data/Hora:** 17/Sep/2025 19:53:54
- **Evento:** SQLi detectado
- **Resposta HTTP:** 302
- **Log:** Message: SQL Injection Attack Detected via libinjection

  Action: Detected (no block)

  Inbound Anomaly Score: 5

Logs de Bloqueio (Blocking Mode)

- **Data/Hora:** 17/Sep/2025 19:55:38
- **Evento:** XSS detectado e bloqueado
- **Resposta HTTP:** 403 Forbidden
- **Log:** Message: XSS Attack Detected via libinjection

  Matched Data: <script>alert("XSS")</script>

  Inbound Anomaly Score: 20

  Action: Deny, status: 403

Scripts

Reconhecimento com nmap

nmap -Ss -sV waf_modsec

SQL Injection (teste)

192.168.35.11 - - [17/Sep/2025:19:53:54 +0000] "GET /vulnerabilities/sqli/?id=1'+OR+'1'='1'--+-&Submit=Submit HTTP/1.1" 403 146 "-" "curl/8.15.0" "-"

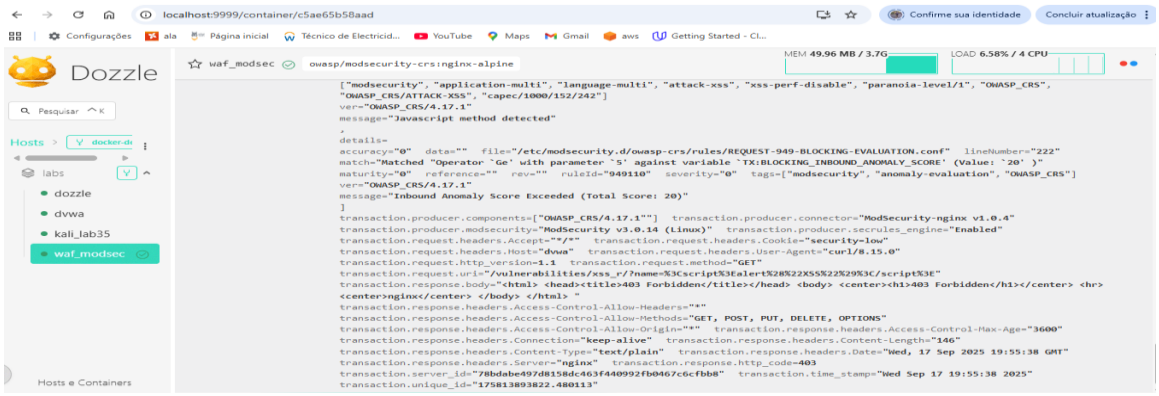
XSS (teste)

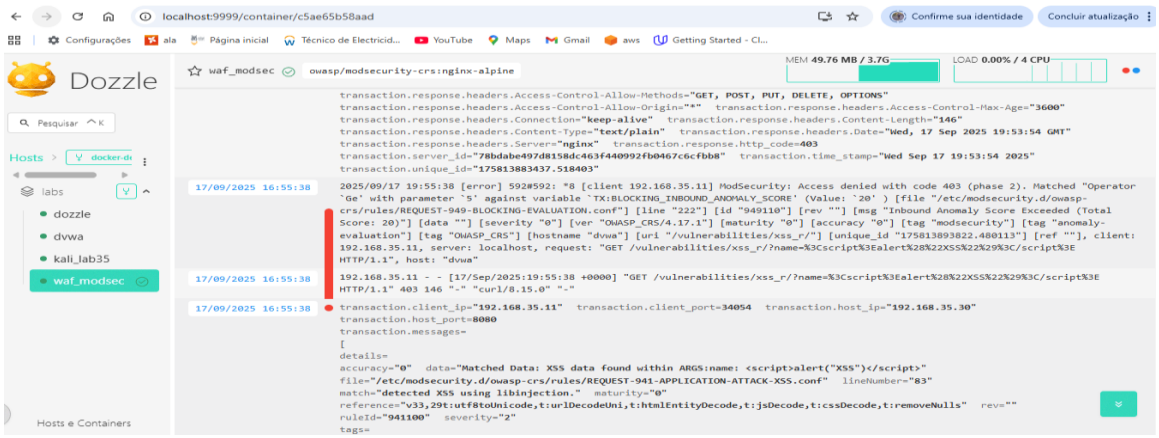
192.168.35.11 - - [17/Sep/2025:19:55:38 +0000] "GET /vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%22XSS%22%29%3C/script%3E HTTP/1.1" 403 146 "-" "curl/8.15.0" "-"

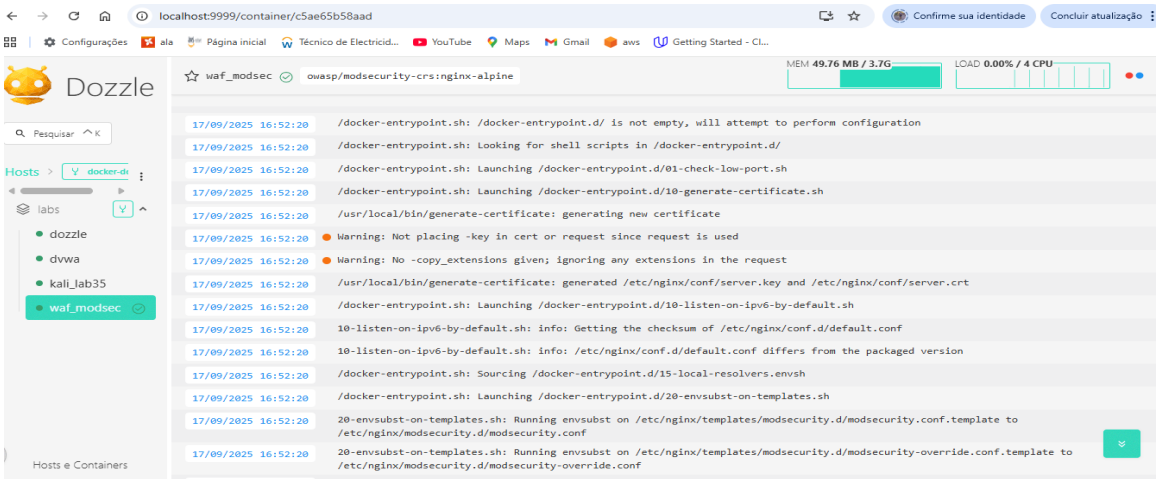
Captura de Pacotes (tcpdump)

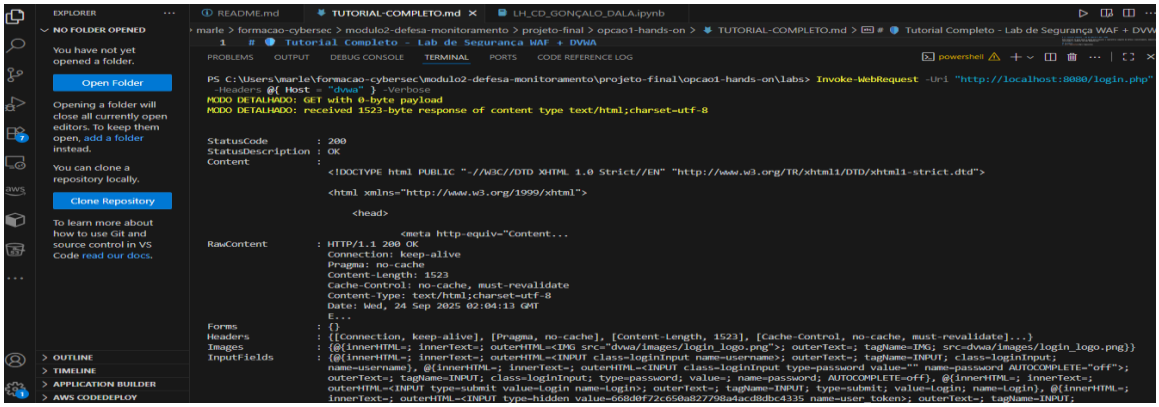
tcpdump -i eth0 port 8080 -w ataques.pcap | tcpdump -i eth0 port 8443 -w ataques.pcap

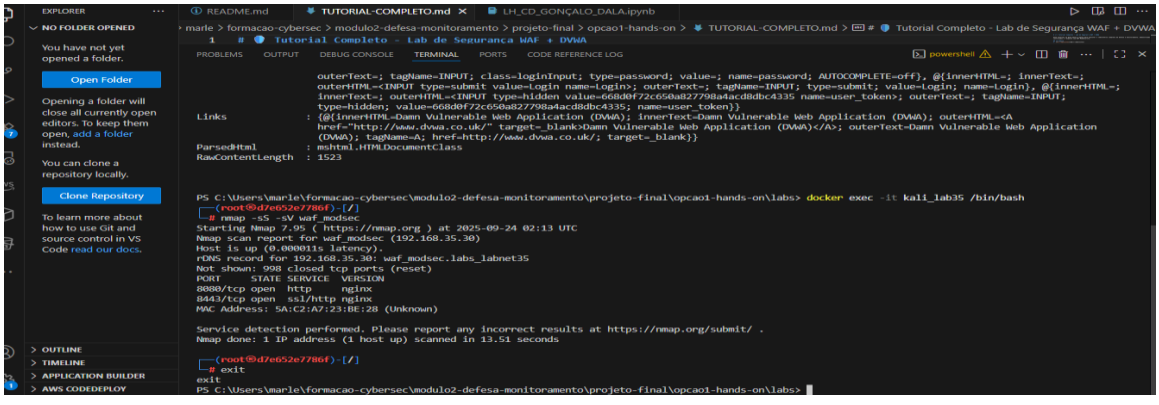
Prints:

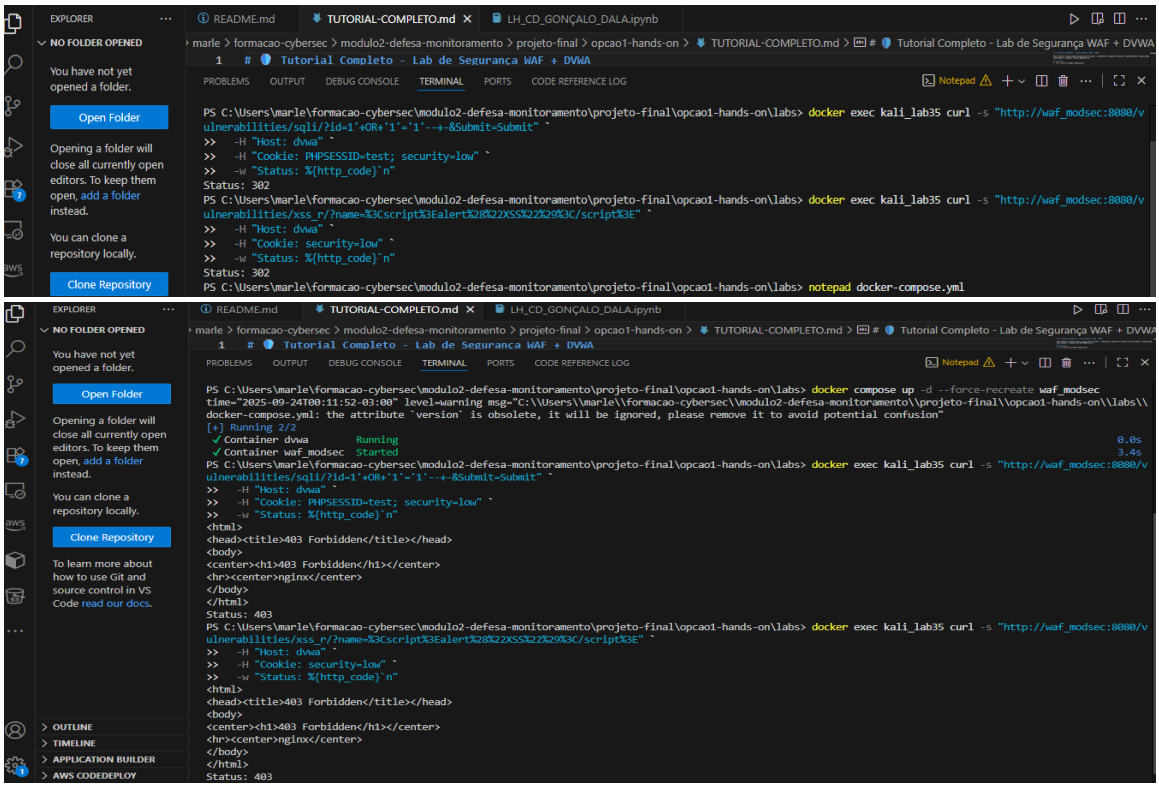



localhost:9999/container/c5ae65b58aad
Dozzle
waf_modsec
owasp/modsecurity-crs:nginx-alpine
MEM 49.76 MB / 3.7G
LOAD 0.00% / 4 CPU
Hosts
labs
dozzle
dvwa
kali_lab35
waf_modsec
Hosts e Containers
transaction.response.headers.Access-Control-Allow-Methods="GET, POST, PUT, DELETE, OPTIONS"
transaction.response.headers.Access-Control-Allow-Origin="*"
transaction.response.headers.Access-Control-Max-Age="3600"
transaction.response.headers.Connection="keep-alive"
transaction.response.headers.Content-Length="146"
transaction.response.headers.Content-Type="text/plain"
transaction.response.headers.Date="Wed, 17 Sep 2025 19:53:54 GMT"
transaction.response.headers.Server="nginx"
transaction.response.http_code=403
transaction.server_id="78bdabe497d8158dc463f440992fb0467c6cfbb8"
transaction.time_stamp="Wed Sep 17 19:53:54 2025"
transaction.unique_id="175813883437.518403"
17/09/2025 16:55:38
2025/09/17 19:55:38 [error] 592#592: *8 [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 "dvwa"] [uri "/vulnerabilities/xss_r/"] [unique_id "175813893822.480113"] [ref ""], client: 192.168.35.11, server: localhost, request: "GET /vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%22XSS%22%29%3C/script%3E HTTP/1.1", host: "dvwa"
17/09/2025 16:55:38
192.168.35.11 - - [17/Sep/2025:19:55:38 +0000] "GET /vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%22XSS%22%29%3C/script%3E HTTP/1.1" 403 146 "-" "curl/8.15.0" "-"
17/09/2025 16:55:38
transaction.client_ip="192.168.35.11" transaction.client_port=34054 transaction.host_ip="192.168.35.30"
transaction.host_port=8080
transaction.messages=
[
details=
accuracy="0" data="Matched Data: XSS data found within ARGS:name: <script>alert("XSS")</script>"
file="/etc/modsecurity.d/owasp-crs/rules/REQUEST-941-APPLICATION-ATTACK-XSS.conf" lineNumber="83"
match="detected XSS using libinjection." maturity="0"
reference="v33,29t:utf8toUnicode,t:urlDecodeUni,t:htmlEntityDecode,t:jsDecode,t:cssDecode,t:removeNulls" rev=""
ruleId="941100" severity="2"
tags=


localhost:9999/container/c5ae65b58aad
Dozzle
waf_modsec
owasp/modsecurity-crs:nginx-alpine
MEM 49.76 MB / 3.7G
LOAD 0.00% / 4 CPU
Hosts
labs
dozzle
dvwa
kali_lab35
waf_modsec
Hosts e Containers
17/09/2025 16:52:20 /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
17/09/2025 16:52:20 /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
17/09/2025 16:52:20 /docker-entrypoint.sh: Launching /docker-entrypoint.d/01-check-low-port.sh
17/09/2025 16:52:20 /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-generate-certificate.sh
17/09/2025 16:52:20 /usr/local/bin/generate-certificate: generating new certificate
17/09/2025 16:52:20 Warning: Not placing -key in cert or request since request is used
17/09/2025 16:52:20 Warning: No -copy_extensions given; ignoring any extensions in the request
17/09/2025 16:52:20 /usr/local/bin/generate-certificate: generated /etc/nginx/conf/server.key and /etc/nginx/conf/server.crt
17/09/2025 16:52:20 /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
17/09/2025 16:52:20 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
17/09/2025 16:52:20 10-listen-on-ipv6-by-default.sh: info: /etc/nginx/conf.d/default.conf differs from the packaged version
17/09/2025 16:52:20 /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
17/09/2025 16:52:20 /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
17/09/2025 16:52:20 20-envsubst-on-templates.sh: Running envsubst on /etc/nginx/templates/modsecurity.d/modsecurity.conf.template to /etc/nginx/modsecurity.d/modsecurity.conf
17/09/2025 16:52:20 20-envsubst-on-templates.sh: Running envsubst on /etc/nginx/templates/modsecurity.d/modsecurity-override.conf.template to /etc/nginx/modsecurity.d/modsecurity-override.conf


EXPLORER
NO FOLDER OPENED
Open Folder
Clone Repository
README.md
TUTORIAL-COMPLETO.md
LH_CD_GONÇALO_DALA.ipynb
marle > formacao-cybersec > modulo2-defesa-monitoramento > projeto-final > opcao1-hands-on > TUTORIAL-COMPLETO.md > # Tutorial Completo - Lab de Segurança WAF + DVWA
1 # Tutorial Completo - Lab de Segurança WAF + DVWA
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG
PS C:\Users\marle\formacao-cybersec\modulo2-defesa-monitoramento\projeto-final\opcao1-hands-on\labs> Invoke-WebRequest -Uri "http://localhost:8080/login.php" -Headers @{ Host = "dvwa" } -Verbose
MODO DETALHADO: GET with 0-byte payload
MODO DETALHADO: received 1523-byte response of content type text/html;charset-utf-8
StatusCode : 200
StatusDescription : OK
Content :
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content...
RawContent : HTTP/1.1 200 OK
Connection: keep-alive
Pragma: no-cache
Content-Length: 1523
Cache-Control: no-cache, must-revalidate
Content-Type: text/html;charset-utf-8
Date: Wed, 24 Sep 2025 02:04:13 GMT
E...
Forms : {}
Headers : {[Connection, keep-alive], [Pragma, no-cache], [Content-Length, 1523], [Cache-Control, no-cache, must-revalidate]...}
Images : {@{innerHTML=; innerText=; outerHTML=<IMG src="dvwa/images/login_logo.png">; outerText=; tagName=IMG; src=dvwa/images/login_logo.png}}
InputFields : {@{innerHTML=; innerText=; outerHTML=<INPUT class=loginInput name=username>; outerText=; tagName=INPUT; class=loginInput; name=username}, @{innerHTML=; innerText=; outerHTML=<INPUT class=loginInput type=password value="" name=password AUTOCOMPLETE="off">; outerText=; tagName=INPUT; class=loginInput; type=password; value=; name=password; AUTOCOMPLETE=off}, @{innerHTML=; innerText=; outerHTML=<INPUT type=submit value=Login name=Login>; outerText=; tagName=INPUT; type=submit; value=Login; name=Login}, @{innerHTML=; innerText=; outerHTML=<INPUT type=hidden value=668d0f72c650a827798a4acd8dbc4335 name=user_token>; outerText=; tagName=INPUT;
OUTLINE
TIMELINE
APPLICATION BUILDER
AWS CODEDEPLOY


EXPLORER
NO FOLDER OPENED
Open Folder
Clone Repository
README.md
TUTORIAL-COMPLETO.md
LH_CD_GONÇALO_DALA.ipynb
marle > formacao-cybersec > modulo2-defesa-monitoramento > projeto-final > opcao1-hands-on > TUTORIAL-COMPLETO.md > # Tutorial Completo - Lab de Segurança WAF + DVWA
1 # Tutorial Completo - Lab de Segurança WAF + DVWA
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG
outerText=; tagName=INPUT; class=loginInput; type=password; value=; name=password; AUTOCOMPLETE=off}, @{innerHTML=; innerText=; outerHTML=<INPUT type=submit value=Login name=Login>; outerText=; tagName=INPUT; type=submit; value=Login; name=Login}, @{innerHTML=; innerText=; outerHTML=<INPUT type=hidden value=668d0f72c650a827798a4acd8dbc4335 name=user_token>; outerText=; tagName=INPUT; type=hidden; value=668d0f72c650a827798a4acd8dbc4335; name=user_token}}
Links : {@{innerHTML=Damn Vulnerable Web Application (DVWA); innerText=Damn Vulnerable Web Application (DVWA); outerHTML=<A href="http://www.dvwa.co.uk/" target=_blank>Damn Vulnerable Web Application (DVWA)</A>; outerText=Damn Vulnerable Web Application (DVWA); tagName=A; href=http://www.dvwa.co.uk/; target=_blank}}
ParsedHtml : mshtml.HTMLDocumentClass
RawContentLength : 1523
PS C:\Users\marle\formacao-cybersec\modulo2-defesa-monitoramento\projeto-final\opcao1-hands-on\labs> docker exec -it kali_lab35 /bin/bash
┌──(root㉿d7e652e7786f)-[/]
└─# nmap -sS -sV waf_modsec
Starting Nmap 7.95 (https://nmap.org) at 2025-09-24 02:13 UTC
Nmap scan report for waf_modsec (192.168.35.30)
Host is up (0.000011s 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: 5A:C2:A7:23:BE:28 (Unknown)
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.51 seconds
┌──(root㉿d7e652e7786f)-[/]
└─# exit
exit
PS C:\Users\marle\formacao-cybersec\modulo2-defesa-monitoramento\projeto-final\opcao1-hands-on\labs>
OUTLINE
TIMELINE
APPLICATION BUILDER
AWS CODEDEPLOY



Glossario e Apêndice

Glossario (Termos e Definições)

Termo / Sigla	Definição
WAF	Web Application Firewall.
CRS	Core Rule Set (OWASP).
SQLi	SQL Injection.
XSS	Cross-Site Scripting.
Nmap	Scanner de portas e serviços.
302	HTTP redirect temporário – em detecção sem bloqueio.
403	HTTP Forbidden – bloqueio ativo pelo WAF.
CIA	Confidencialidade, Integridade, Disponibilidade.
NIST IR	Ciclo NIST: Detecção, Análise, Contenção, Erradicação, Recuperação.

Apêndice (Politicas NIST IR)

Categorias de Incidentes

Categoria	Descrição	Exemplo
CAT-01	Ataques Web Application	SQLi, XSS, LFI/RFI
CAT-02	Ataques de Rede	DDoS, Port Scanning
CAT-03	Acesso Não Autorizado	Bruteforce, Credential Stuffing
CAT-04	Vazamento de Dados	Exfiltração de informações sensíveis

Matriz de Urgência

Critério	Baixa	Média	Alta
Propagação	Isolado	Limitado	Em expansão
Impacto Negócio	Mínimo	Moderado	Severo
Exposição Legal	Nenhuma	Potencial	Imediata

Níveis de Severidade

Nível	Impacto	Exemplo
SEV-1 - Crítico	Comprometimento total do sistema	RCE, Data Breach em larga escala
SEV-2 – Alto	Comprometimento parcial	SQLi bem-sucedido, acesso a dados
SEV-3 – Médio	Tentativa de ataque bloqueada	SQLi/XSS bloqueados pelo WAF
SEV-4 - Baixo	Atividade suspeita	Scanning, tentativas leves

Matriz de Impacto CIA

Dimensão	Baixo	Médio	Alto
Confidencialidade	Dados públicos	Dados internos	Dados sensíveis/PII
Integridade	Alteração não crítica	Alteração parcial	Alteração crítica
Disponibilidade	Interrupção < 1h	Interrupção 1-4h	Interrupção > 4h

Matriz de Priorização

Severidade	Urgência Baixa	Urgência Média	Urgência Alta
SEV-1	P2	P1	P1 (Crítico)
SEV-2	P3	P2	P1
SEV-3	P4	P3	P2
SEV-4	P5	P4	P3