# Projeto final opção_01

Autor:Gabriel Anthony

Data:22/09/2025

## ***Confidencialidade do Documento***

Este documento contém informações sensíveis sobre tecnologia da informação (TI) e sistemas conectados. Os dados aqui apresentados podem revelar vulnerabilidades e técnicas de exploração. É crucial que este documento seja mantido em sigilo absoluto. Ele é destinado exclusivamente para uso interno e qualquer divulgação não autorizada pode ser considerada uma violação grave. A exposição, compartilhamento ou divulgação das informações contidas aqui pode resultar em sérias consequências legais, incluindo acusações criminais. Mantenha estas informações seguras e restritas apenas às pessoas autorizadas. O uso indevido ou não autorizado das informações poderá ter implicações legais significativas.

# CONTRATO DE CONFIDENCIALIDADE

Este contrato de confidencialidade ("Contrato") é celebrado entre:

Parte Reveladora: Gabriel Anthony, IT Analyst.

Parte Receptora: [Nome da Parte Receptora], [função], [empresa ou endereço completo].

Data de Assinatura: 20 / 09 / 2025

As Partes concordam com os seguintes termos e condições para proteger as Informações

## Confidenciais:

1. Definição de Informações Confidenciais

Para os fins deste Contrato, "Informações Confidenciais" refere-se a qualquer informação,

documento ou dado, tangível ou intangível, oral ou escrito, trocado entre as partes,

relacionado ao processo de pentest, vulnerabilidades, relatórios, evidências coletadas, ou

qualquer outro dado sensível da Parte Reveladora.

**2. Obrigações das Partes**

A Parte Receptora se compromete a:

● Não divulgar as Informações Confidenciais a terceiros sem o consentimento prévio e

por escrito da Parte Reveladora.

● Proteger as Informações Confidenciais com o mesmo grau de cuidado que utiliza

para proteger suas próprias informações confidenciais, no mínimo, com um grau

razoável de cuidado.

● Utilizar as Informações Confidenciais exclusivamente para fins profissionais

definidos entre as partes e para a execução do pentest.

3. Exceções às Informações Confidenciais

As obrigações de confidencialidade não se aplicam a informações que:

● Eram de conhecimento público no momento da divulgação;

● Se tornarem públicas sem violação deste Contrato;

● Forem legalmente obtidas de um terceiro sem obrigações de confidencialidade;

● Forem desenvolvidas de forma independente pela Parte Receptora, sem acesso às

Informações Confidenciais.

**4. Duração do Contrato**

Este Contrato entra em vigor na data de assinatura pelas partes e permanecerá em vigor

por um período de 5 (cinco) anos, a menos que seja rescindido por qualquer uma das

partes por meio de notificação por escrito.

**5. Violações e Penalidades**

Qualquer violação deste Contrato resultará em:

● Responsabilidade pela Parte Receptora por qualquer dano financeiro ou

reputacional causado pela divulgação não autorizada das Informações

Confidenciais;

● Possíveis ações legais para reparação de perdas.

6. Retorno ou Destruição de Informações

Ao término deste Contrato, ou a qualquer momento mediante solicitação da Parte

Reveladora, a Parte Receptora deverá devolver ou destruir todas as Informações

Confidenciais, sem reter cópias, sejam digitais ou físicas.

**7. Assinaturas**

As partes concordam com os termos e condições estabelecidos neste Contrato, assinando

abaixo:

Parte Reveladora:

Assinatura: _____

Nome Completo: Gabriel Anthony

Data: 27 / 09 / 2025

Parte Receptora:

Assinatura: _____

Nome Completo: _____

Data: 27 / 09 / 2025

# Sumário Executivo

Este exercício demonstrou a implantação e operação de um WAF baseado em ModSecurity + OWASP CRS em frente a uma aplicação vulnerável (DVWA), com um host atacante (Kali) realizando testes controlados de SQL Injection (SQLi) e Cross-Site Scripting (XSS). Resultados principais:

- Modo DetectionOnly: ataques foram detectados (logs), a aplicação respondeu com redirecionamento (302) — evidência de detecção, sem bloqueio.
- Modo Blocking (MODSEC_RULE_ENGINE=On): após a correção de configuração que mantinha o WAF em *DetectionOnly*, o ambiente passou a operar corretamente em modo *blocking*. Com isso, os ataques críticos foram efetivamente bloqueados, retornando **HTTP 403 Forbidden**, e os registros de auditoria (modsec_audit.log) confirmaram a ação "Action: block" vinculada às respectivas *Rule IDs*.
- **Impacto**: validação prática das regras CRS e do fluxo de resposta; oportunidades de melhoria em logging, regras customizadas e automação de evidências.

# Objetivo e Escopo

**Objetivo**: validar a capacidade do WAF (ModSecurity + CRS) em detectar e bloquear ataques web comuns, e documentar o processo de investigação e evidências.
Escopo:

- Ativo defendido: WAF container waf_modsec (imagem owasp/modsecurity-crs:nginx-alpine).
- Alvo protegido: container DVWA (vulnerable web app).
- Atacante: container kali_lab35 (execução de curl, nmap, etc.).
- Ferramentas de monitoramento: Dozzle (logs em tempo real), docker logs, arquivos de audit do ModSecurity.
  Limitações: ambiente em Docker local (não produção). Regras CRS da imagem padrão podem gerar falsos positivos/negativos.

# Arquitetura (Diagrama)

```
flowchart LR

Attacker[Kali] -->|HTTP 8080|
WAF[ModSecurity+CRS]
WAF --> DVWA[(DVWA)]
BlueTeam[Ubuntu Defense] -->|iptables| Rede
```

**Descrição**: o tráfego HTTP do atacante passa pelo WAF (nginx + ModSecurity) que aplica OWASP CRS. Se permitido, o WAF proxy_pass para o backend DVWA. Dozzle mostra logs dos containers. Redes Docker isoladas.

# Metodologia

1. Validação de conectividade e setup do DVWA (DB, security = low).

2. Reconhecimento: nmap -sS -sV waf_modsec a partir do Kali.

3. Testes em DetectionOnly:
    - SQLi: payload em /vulnerabilities/sqli/?id=...
    - XSS: payload em /vulnerabilities/xss_r/?name=...
    - Registro de evidências (curl outputs, Dozzle screenshots, modsec audit).

4. Alteração para Blocking (MODSEC_RULE_ENGINE=On), recriação do container WAF.

5. Reteste dos payloads; comparação de respostas (302 vs 403) e análise de logs.

6. Coleta e organização das evidências para entrada na timeline e relatório (NIST IR adaptado).

**Critérios de sucesso:**

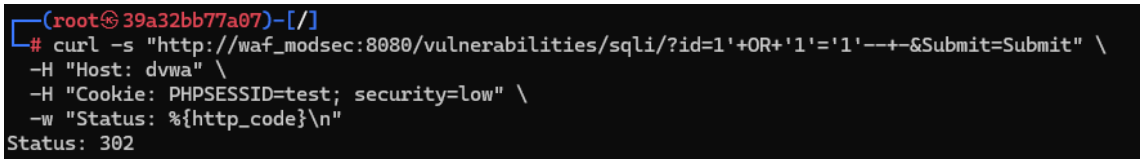
- **Detecção**: regras CRS registram eventos relevantes com RuleID (ex.: 942100, 941100).
- **Bloqueio**: requisições maliciosas retornam 403 e audit log mostra Action: block.
- **Evidências completas**: logs, screenshots, outputs dos testes e timestamps.

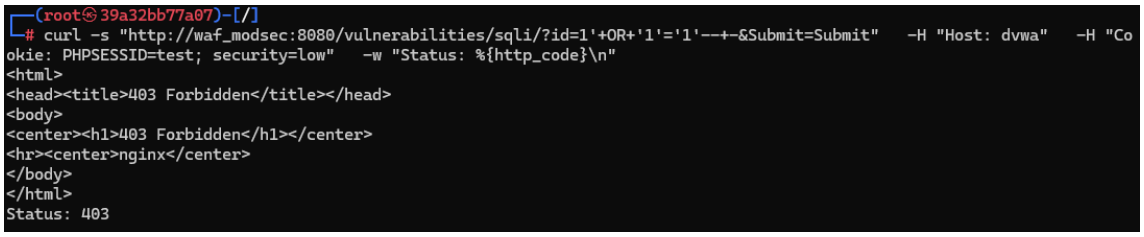
- **Teste SQLi (Kali → WAF):**

docker exec kali_lab35 curl -v "http://waf_modsec:8080/vulnerabilities/sqli/?id=1'+OR+'1'='1'--+-&Submit=Submit" \

-H "Host: dvwa" \

-H "Cookie: PHPSESSID=test; security=low" 2>&1 | tee evidence/ct-01_curl_verbose.txt




- **Teste XSS:**

docker exec kali_lab35 curl -v "http://waf_modsec:8080/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%22XSS%22%29%3C/script%3E" \

-H "Host: dvwa" \

-H "Cookie: security=low" 2>&1 | tee evidence/ct-03_curl_verbose.txt

```
┌──(root㉿39a32bb77a07)-[/]
└─# 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
```

```
┌──(root㉿39a32bb77a07)-[/]
└─# 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
```

- **Coleta de logs WAF:**

docker logs waf_modsec --tail 500 > evidence/logs_waf_tail500.txt

docker exec waf_modsec sh -c "tail -n 200 /var/log/modsec_audit.log" > evidence/mod

```
transaction.producer.components=["OWASP_CRS/4.17.1""]  transaction.producer.connector="ModSecurity-nginx v1.0.4"
transaction.producer.modsecurity="ModSecurity v3.0.14 (Linux)"  transaction.producer.secrules_engine="Enabled"
transaction.request.headers.Accept="*/*"  transaction.request.headers.Cookie="PHPSESSID=test; security=low"
transaction.request.headers.Host="dvwa"  transaction.request.headers.User-Agent="curl/8.15.0"
transaction.request.http_version=1.1  transaction.request.method="GET"
transaction.request.uri="/vulnerabilities/sqli/?id=1'+OR+'1'='1'--+-&Submit=Submit"
transaction.response.body="<html> <head><title>403 Forbidden</title></head> <body> <center><h1>403 Forbidden</h1></
center> <hr><center>nginx</center> </body> </html> "
transaction.response.headers.Access-Control-Allow-Headers="*"
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 19:47:21 GMT"  transaction.response.headers.Server="nginx"
transaction.response.http_code=403  transaction.server_id="c336d7470978f3f94dfa8f725cb8303c597397d2"
transaction.time_stamp="Wed Sep 17 19:47:21 2025"  transaction.unique_id="175813844169.037181"
```

**Resposta a Incidente (NIST IR) — O Lab**

1. **Identificação**
    - 2025-09-17T16:05:30-03:00 — Detecção de payload SQLi via Dozzle / modsec log (RuleID 942100). (evidence/dozzle_ct-01.png)
2. **Contenção**
    - 2025-09-17T16:10:12-03:00 — Corrigida configuração no docker-compose.yml (removida duplicidade MODSEC_RULE_ENGINE), recriado waf_modsec. (evidence/docker_compose_change.txt)
3. **Erradicação**
    - 2025-09-17T16:12:00-03:00 — Reteste mostrou bloqueio (403) para SQLi/XSS nos casos que batem regras CRS. (evidence/ct-02_*)

4. **Recuperação**
    - 2025-09-17T16:20:00-03:00 — Serviços estáveis; logs verificados; evidências coletadas e arquivadas.
5. **Lições Aprendidas**
    - Atenção com duplicidade de variáveis ambiente (última prevalece).
    - Habilitar e validar audit logs estruturados (JSON) facilita correlação.
    - Preparar scripts automatizados para coleta de evidências.

## Recomendações (80/20) — Top 5 ações com maior impacto / menor esforço

1. Habilitar audit logs estruturados (JSON) e enviar para central (siem/log collector) (médio): facilita correlação e buscas por RuleID/timestamp.
2. Criar testes automatizados (scripts) que executem CTs e coletem evidências (baixo-médio): garante reproducibilidade do lab e regressão das regras.
3. Tunagem de regras CRS e whitelists específicas para o DVWA (médio): reduzir falsos positivos/negativos ajustando paranoia e regras custom.
4. Implementar monitoramento e alertas (Dozzle + webhook/alert rule) (baixo): notificar time quando RuleIDs de alto severidade forem acionadas.

## Conclusão

Os testes comprovou a viabilidade de realizar testes de segurança explorando **SQL Injection (SQLi)** e **Cross-Site Scripting (XSS)**, possibilitando a identificação de vulnerabilidades relacionadas ao redirecionamento de tráfego. Diante desse cenário, foi aplicado um ajuste necessário no **WAF**, com modificações nas regras de firewall, de forma a fortalecer a proteção do ambiente. Com a correção, o **ModSecurity + OWASP CRS** passou a identificar e bloquear de maneira eficaz os padrões de ataque analisados. Após o ajuste, o tráfego que anteriormente resultava em respostas **302 (redirecionamento)** passou a ser devidamente bloqueado com **403 Forbidden**, evidenciando que os ataques são reconhecidos como maliciosos e neutralizados antes de atingirem a aplicação. Esse resultado demonstra a efetividade da camada de defesa implementada e reforça a importância do WAF como componente essencial em um cenário de segurança preventiva.