Manual do Projeto

Tatiane de Matos Silva

Henrique Christopher de Castro Leão

Manual do projeto: Testes de Vulnerabilidades Web

Este documento apresenta a visão geral da arquitetura de rede implementada no ambiente do cliente, descrevendo cada componente, suas funções e como eles se relacionam.

A plataforma permite que sua equipe visualize, acompanhe e compreenda como funcionam ataques controlados em um ambiente seguro. Lembrando que, todas as máquinas alvo estão isoladas e não afetam sua infraestrutura real.

1. Componentes da arquitetura

1. PfSense – Firewall e Roteador

- Recebe a Internet pela interface WAN (em0)
- Gerencia a rede LAN (em1)
- Gerencia a rede OPT1 (em2) – Host-Only

Atua como:

- Firewall
- Roteador
- Servidor DHCP para a rede 192.168.200.0/24
- Controlador de isolamento das redes internas

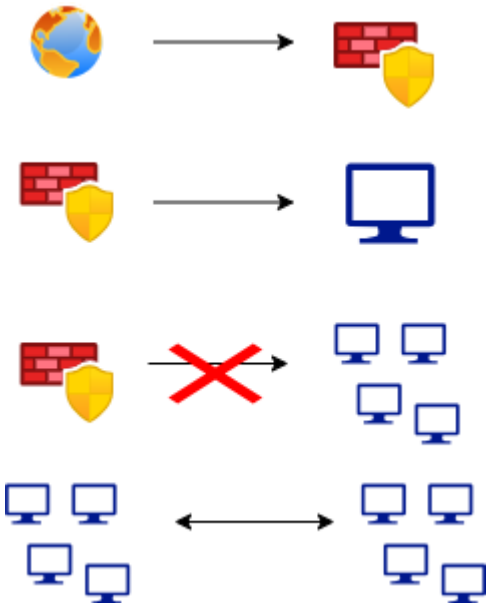
2. Host Físico (computador do cliente) – Rede Host-Only

- Conectado à interface OPT1 do pfSense.
- Recebe IP automaticamente (ex.: 192.168.200.100).
- Acesso administrativo ao pfSense.

3. Máquinas Virtuais – Rede Interna "labvirtual"

- Ambiente totalmente isolado.
- Comunicação apenas entre máquinas virtuais.
- Não possuem rota de saída para o pfSense nem para o Host.
- Usadas para testes e simulação de ataques controlados.

Fluxo de Rede Resumido



Internet → pfSense (WAN)

pfSense (OPT1) → Host físico

pfSense NÃO se comunica com as VMs e as VMs só falam entre si, pois estão em rede isolada

2. Tutorial – Entendendo a Rede Host-Only

A rede Host-Only, é uma rede isolada; pfSense e Host Físico (não tem internet direta).

Serve para:

- Administração do pfSense
- Captura de pacotes
- Ferramentas de diagnóstico

Faixa de rede

- 192.168.200.0/24
- Gateway: 192.168.200.1 (pfSense)
- DHCP: 192.168.200.10 – 192.168.200.200

Benefícios:

- Segurança total: não expõe o pfSense na LAN real
- Canal exclusivo de administração
- Evita que as VMs tenham acesso acidental ao firewall

3. Tutorial – Como Acessar o pfSense

Os pré-requisitos é estar conectado na rede Hosty-Only e ter um navegador instalado

1. Verificar o IP do Host:

No Windows, digitar no CMD o comando “ipconfig” vai aparecer “Adaptador Host-Only IPv4: 192.168.200.100

2. Acessar o pfSense:

Abra o navegador e digite: https://192.168.200.1. O primeiro login é padrão: Admin e pfsense.

O cliente pode visualizar:

- Logs de ataque
- Regras de firewall
- Relatórios de conexão
- Estatísticas de tráfego

3. Dentro do pfSense

Você verá a tela de dashboard, como: Status do sistema, tráfego, interfaces, regras de firewall, monitoramento, logs de conexões, entre outros.

4. Ambiente das máquinas virtuais

1. O cliente poderá visualizar:

- Máquina atacante
- Máquinas vulneráveis
- Processos e logs
- Resultados dos testes de segurança
- Tudo isso dentro do VirtualBox.

2. Dentro do VirtualBox

O cliente verá uma lista como:

- Firewall – pfSense
- AttackBox (máquina que fara o ataque)

- VM1 – Servidor Vulnerável
- VM2 – Aplicação Vulnerável
- VM3 – Serviço Auxiliar Vulnerável

3. Acessar as máquinas

Cada máquina tem um IP fixo na rede isolada:

Máquina	**IP**	**Função**
AttackBox	10.0.0.200	Executa os ataques
VM1	10.0.0.101	Servidor vulnerável
VM2	10.0.0.102	Servidor vulnerável
VM3	10.0.0.103	Exemplo de serviço vulnerável

O cliente pode executar ataques controlados (com o script), ver logs, monitorar respostas, repetir testes, fazer demonstrações internas

5. Como será a execução dos testes (Passo a Passo)

Passo 1 — Ligar pfSense

Isso ativa a rede host-only e distribui os IPs internos.

Passo 2 — Ligar Máquina Atacante (AttackBox)

Ela já contém o script configurado.

Passo 3 — Ligar as Máquinas Vulneráveis

Elas respondem às conexões da AttackBox.

Passo 4 — Rodar o Script de Testes

Na AttackBox, o cliente executa: ./scan_vulnerabilidades.sh

O cliente verá: Portas abertas, serviços ativos, respostas das VMs, identificação de falhas simuladas

Passo 5 — Ver resultados no Firewall

No painel do pfSense, o cliente pode ver: Conexões de entrada, logs de ataques simulados, regras que permitiram ou bloquearam tráfego, gráficos e estatísticas

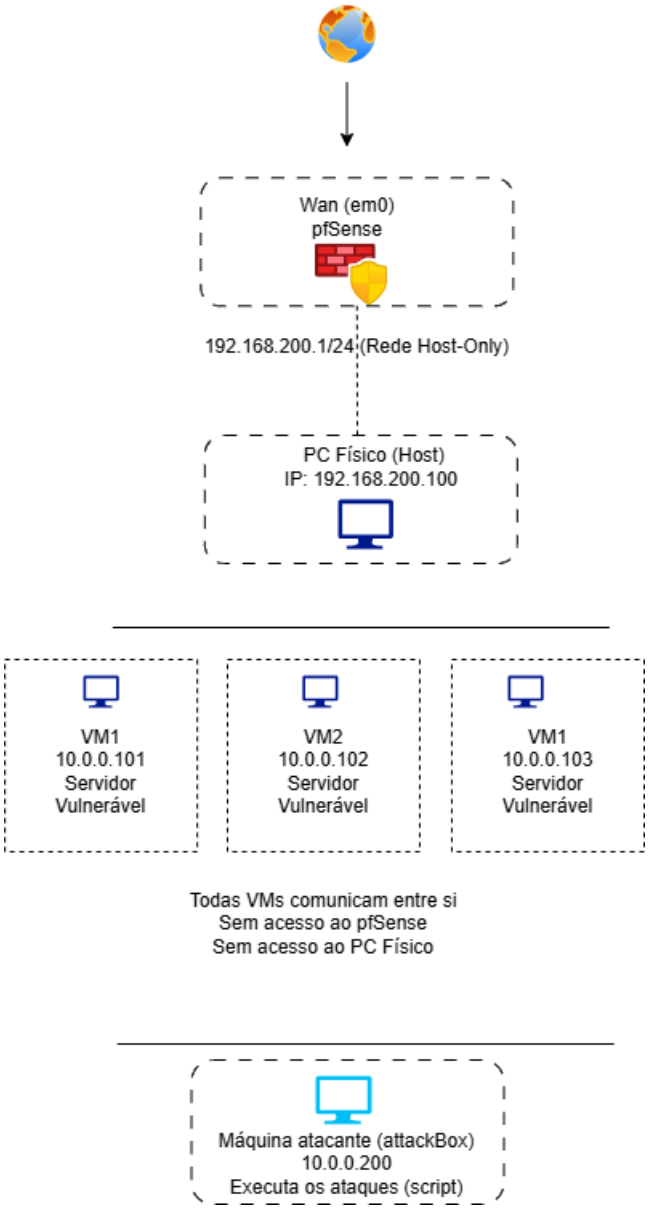
6. Segurança e entrega

O ambiente é totalmente isolado:

- Sem acesso à internet pelas VMs
- Sem risco de vazamento
- Sem impacto na rede real do cliente
- Os ataques ocorrem somente na rede LABVIRTUAL

O cliente receberá:

- O computador/servidor já configurado
- Todas as VMs funcionando
- Script de ataques pré-instalado
- Manual em PDF/Word
- Acesso ao firewall
- Suporte para dúvidas e manutenção



Funções e Relações Entre os Componentes

pfSense (Firewall)

Atua como divisor entre a rede do cliente e a rede de laboratório, permite monitorar ataques gerados pelas VMs. Garantia de que nada sai da rede interna para a internet e só o PC físico pode acessar o pfSense.

PC Físico (Host)

É a estação de controle do cliente, que conecta no pfSense para: ver logs, monitorar os ataques, acompanhar o comportamento da rede.

Rede Interna “labvirtual”

Isolada totalmente do mundo externo. As VMs se comunicam entre si e nunca acessam o firewall e éonde ocorrem os testes e ataques.

VMs Vulneráveis

Simulam serviços com falhas reais e são os alvos do script de ataque.

AttackBox

Máquina responsável por executar as varreduras, testes de intrusão, scripts de exploração, coleta de evidências.

E como isso acontece? Esta seção deve ser enviada ao cliente final. Tudo explicado sem termos técnicos complexos.

7. Segurança e recomendações

- Salvar backups do pfSense sempre antes de mudanças significativas.
- Coloque descrições nas regras do firewall.
- Conhecimento das principais vulnerabilidades web

8. Requisitos Técnicos

Requisitos de Software:

- Python ou C#
- VitualBox
- PfSense
- Suricata

Requisitos de Hardware:

- Processador compatível com virtualização
- Espaço livre em disco para Máquinas Virtuais (+/- 30 GB)

9. Sobre o Código

O script `zap_passive_report.py` é uma ferramenta segura de análise passiva de vulnerabilidades, projetada para ser usada **exclusivamente em ambiente de laboratório**. Ele utiliza o *OWASP ZAP* para realizar **coleta de informações e análise passiva**, sem executar ataques ativos.

- Conecta ao OWASP ZAP
- Faz um *Spider* para descobrir páginas do site alvo
- Aguarda a finalização do *Passive Scan*
- Coleta todos os alertas encontrados
- Gera relatórios em **JSON** e **CSV**
- Classifica alertas por nível de risco

1. Conexão com o OWASP ZAP

A função `connect_zap()` abre uma conexão entre o script e o OWASP ZAP usando proxy na porta 8080.

Isso permite que o Python controle o ZAP:

- enviar comandos
- puxar alertas
- iniciar varreduras

Se o ZAP não estiver aberto, o script encerra automaticamente.

2. Spider Scan (Mapeamento de Páginas)

A função `spider_scan()`:

- manda o ZAP visitar o site alvo (`TARGET`)
- coleta links internos
- mapeia o site sem atacar

É uma navegação automatizada que descobre páginas e parâmetros.

O script acompanha o progresso até 100% ou até atingir um *timeout* de segurança.

3. Passive Scan (Análise Passiva)

A função `passive_scan()`:

- aguarda o ZAP analisar o tráfego capturado pelo Spider
- essa análise não envia payloads maliciosos
- verifica cabeçalhos, cookies, boas práticas de segurança, etc.

Só técnicas **não invasivas** são usadas.

O script aguarda até um limite configurado (*PASSIVE_TIMEOUT*) para evitar travamentos.

4. Coleta de Alertas

A função `get_alerts()`:

Solicita todos os alertas gerados pelo ZAP, filtra apenas os da URL alvo, devolve uma lista com detalhes como:

→ nome da vulnerabilidade
→ nível de risco (Low/Medium/High)
→ URL afetada
→ parâmetros envolvidos
→ evidências encontradas

5. Exportação de Relatórios

O script gera automaticamente:

- JSON (zap_passive_alerts.json)

❖ completo
❖ estruturado
❖ útil para auditoria

- CSV (zap_passive_alerts.csv)

- ❖ compatível com Excel
- ❖ fácil para gerar gráficos ou planilhas

Este script é considerado SEGURO

- Não faz Active Scan
- Não dispara payloads
- Não explora falhas
- Apenas coleta e analisa tráfego normal
- Totalmente indicado para ambientes sensíveis