

**UNIVERSIDADE EDUARDO MONDLANE**

**Faculdade de Engenharia**

**Departamento de Electrotécnica**

**Licenciatura em Engenharia Informática**

**Tema: Kali Linux – Vulnerabilidades**

Docentes:

Engª. Ivone Cipriano

Eng°. Délcio Chadreca

Discentes:

Florêncio, Nuno Fonseca

Simão Júnior, Belarmino

*Maputo, Março de 2024*

**Índice**

[Índice de Figuras 4](#_Toc164594945)

[1. Resumo 5](#_Toc164594946)

[2. Conjunto de palavras-chave 6](#_Toc164594947)

[3. Introdução 7](#_Toc164594948)

[4. Objectivos 8](#_Toc164594949)

[4.1. Objectivo geral 8](#_Toc164594950)

[4.2. Objectivos específicos 8](#_Toc164594951)

[5. Metodologia 9](#_Toc164594952)

[6. Desenvolvimento 10](#_Toc164594953)

[6.1. Análise de Vulnerabilidades em Infra-estruturas 10](#_Toc164594954)

[6.1.1. Técnicas de análise de Vulnerabilidades 11](#_Toc164594955)

[6.2. OWASP - Top 10 11](#_Toc164594956)

[6.2.1. Quebra de Controle de Acesso 12](#_Toc164594957)

[6.2.2. Falhas Criptográficas 12](#_Toc164594958)

[6.2.3. Injecção 13](#_Toc164594959)

[6.2.4. Design Inseguro 13](#_Toc164594960)

[6.2.5. Configuração Insegura 14](#_Toc164594961)

[6.2.6. Componente desactualizado e vulnerável 15](#_Toc164594962)

[6.2.7. Falha de identificação e autenticação 15](#_Toc164594963)

[6.2.8. Falha na Integridade de Dados e Software 16](#_Toc164594964)

[6.2.9. Monitoramento de Falhas e Registros de Segurança 16](#_Toc164594965)

[6.2.10. Falsificação de solicitação do lado do Servidor 17](#_Toc164594966)

[6.3. Kali Linux: Uma ferramenta para análise de Vulnerabilidades 18](#_Toc164594967)

[6.3.1. Recursos de Kali Linux 18](#_Toc164594968)

[6.4. OWASP Broken Web Applications 18](#_Toc164594969)

[6.4.1. Recursos do OWASP BWA 19](#_Toc164594970)

[6.5. Ferramentas de análise de vulnerabilidades Web 21](#_Toc164594971)

[6.5.1. Nikto 21](#_Toc164594972)

[6.5.2. Skipfish 22](#_Toc164594973)

[6.5.3. Wapiti 23](#_Toc164594974)

[6.5.4. OWASP-ZAP 24](#_Toc164594975)

[6.5.5. XSSPY 25](#_Toc164594976)

[6.5.6. W3af 26](#_Toc164594977)

[6.6. Ferramentas de análise de vulnerabilidades de Infra-estruturas 27](#_Toc164594978)

[6.6.1. OpenVAS 27](#_Toc164594979)

[6.6.2. Fierce 28](#_Toc164594980)

[6.6.3. Metasploit framework 29](#_Toc164594981)

[6.6.4. Nmap 30](#_Toc164594982)

[6.6.5. Netcat 31](#_Toc164594983)

[6.6.6. Unicornscan 32](#_Toc164594984)

[7. Conclusão 34](#_Toc164594985)

[8. Referências bibliográficas 35](#_Toc164594986)

# Índice de Figuras

[Figure 1. Evolução das Vulnerabilidades OWASP Top 10: Um Comparativo entre 2017 e 2021 11](#_Toc164594937)

# Resumo

# Conjunto de palavras-chave

Análise de Vulnerabilidades, OWASP Top 10, Kali Linux, OWASP BWA, Nikto, Skipfish, Wapiti, OWASP-ZAP, XssPy, W3af, OpenVAS, Fierce, Metasploit Framework, Nmap, Netcat, Unicornscan.

# Introdução

O cenário de segurança da informação está em constante evolução, com o surgimento de novas ameaças e vulnerabilidades a cada dia. As aplicações Web e as infra-estruturas críticas se tornam cada vez mais complexas e interligadas, aumentando a superfície de ataque e os riscos de exploração por agentes mal-intencionados.

Nesse contexto, os testes de vulnerabilidade assumem um papel crucial na identificação e mitigação de riscos de segurança. Através da simulação de ataques reais, os testes de vulnerabilidade permitem detectar falhas de segurança em softwares, sistemas e redes antes que possam ser exploradas por criminosos cibernéticos.

O OWASP (*Open Web Application Security Project*) é uma organização sem fins lucrativos que publica anualmente o OWASP Top 10, uma lista das dez falhas de segurança mais críticas em aplicações Web. O OWASP Top 10 serve como um guia essencial para profissionais de segurança na identificação e priorização de seus esforços de mitigação de riscos.

Além do OWASP Top 10, o OWASP também oferece o OWASP BWA (*Broken Web Applications*), um projecto que fornece um conjunto de aplicações Web vulneráveis ​​pré-construídas para fins de treinamento e testes de penetração. O OWASP BWA é uma ferramenta valiosa para profissionais de segurança que desejam aprimorar suas habilidades em testes de vulnerabilidade e explorar técnicas de exploração de falhas de segurança.

# Objectivos

## Objectivo geral

* Avaliar as principais ferramentas de teste de vulnerabilidade para aplicações Web e infra-estruturas e analisar sua efectividade na identificação e exploração de falhas de segurança.

## Objectivos específicos

* Identificar e analisar as principais ferramentas de teste de vulnerabilidade para aplicações Web e infra-estruturas
* Analisar a efectividade das ferramentas de teste de vulnerabilidade na identificação de falhas de segurança relacionadas ao OWASP Top 10

# Metodologia

A metodologia de pesquisa a ser utilizada neste trabalho envolve as seguintes etapas:

* **Revisão da literatura**: Será realizada uma revisão abrangente da literatura sobre ferramentas de teste de vulnerabilidade, metodologias de testes de segurança e o OWASP Top 10.
* **Selecção de ferramentas**: As ferramentas de teste de vulnerabilidade a serem utilizadas serão as mencionadas no enunciado do trabalho.
* **Implementação dos testes**: Os testes de vulnerabilidade serão implementados em um ambiente controlado, utilizando as ferramentas seleccionadas.
* **Análise dos resultados**: Os resultados dos testes serão colectados e analisados para determinar a efectividade das ferramentas de teste de vulnerabilidade na identificação e exploração de falhas de segurança.

# Desenvolvimento

## Análise de Vulnerabilidades em Infra-estruturas

Qualquer infra-estrutura que deseja ter maior segurança e eficiência nos processos deve estar preparada para qualquer tipo de eventualidade. A análise de vulnerabilidade surge justamente com o objectivo de identificar brechas e falhas na infra-estrutura de TI, evitando ameaças e exposição de dados.

Segundo Whittman e Mattord (2014), uma vulnerabilidade é uma fraqueza ou falha em um sistema ou mecanismo de protecção que o torna vulnerável a ataques ou danos. Alguns exemplos de vulnerabilidades incluem falhas em um software, ou em equipamentos da infra-estrutura. Assim sendo, todo o sistema que manipula e trafega informações está sujeito a alguma vulnerabilidade.

A análise de vulnerabilidades é uma técnica usada para identificar e avaliar as vulnerabilidades de um sistema ou rede. É uma prática fundamental na segurança cibernética, que ajuda na compreensão das áreas de risco que precisam ser protegidas. É um processo de avaliação de sistemas e redes em busca de vulnerabilidades que possam ser exploradas por atacantes. Isso envolve a realização de testes para identificar falhas de segurança em software, hardware e configurações de rede que possam permitir a entrada de invasores, roubo de dados, comprometimento do sistema, entre outros.

A análise de vulnerabilidades geralmente é realizada em três etapas: **descoberta**, **avaliação** e **correcção**. A descoberta envolve a identificação dos activos da rede, como servidores, estações de trabalho, dispositivos móveis, impressoras e outros dispositivos conectados à rede, caso seja uma verificação de software, actua-se na compreensão do funcionamento e regras de negócio, tecnologias, bibliotecas e funcionalidades existentes. A avaliação envolve a identificação de vulnerabilidades e a sua classificação em termos de risco, gravidade e impacto potencial.

A identificação precoce de vulnerabilidades pode ajudar a reduzir o risco de ataques cibernéticos, roubo de dados, comprometimento do sistema e outras ameaças. Além disso, a análise de vulnerabilidades pode ajudar a atender aos requisitos regulatórios de segurança cibernética e a demonstrar o compromisso com a segurança para clientes e outras partes interessadas.

### Técnicas de análise de Vulnerabilidades

A análise de vulnerabilidades pode ser realizada por meio de várias técnicas, incluindo:

* **Varreduras de Portas**: Uma varredura de portas é uma técnica que envolve a verificação de portas abertas em um sistema ou rede. Portas abertas podem indicar serviços activos que possam ser explorados por invasores.
* **Varredura de Vulnerabilidades**: A varredura de vulnerabilidades é uma técnica que envolve a execução de testes automatizados para identificar vulnerabilidades conhecidas em sistemas ou redes.
* **Testes de Invasão**: Os testes de invasão são uma técnica que envolve a simulação de um ataque para identificar vulnerabilidades e explorá-las de maneira controlada. Esse tipo de teste é conduzido com o consentimento do proprietário do sistema ou rede e é geralmente realizado por profissionais de segurança cibernética experientes.

## OWASP - Top 10

OWASP, ou simplesmente *Open Web Application Security Project*, é uma organização sem fins lucrativos dedicada a segurança de aplicativos Web.

O “OWASP Top 10” é um documento de conscientização padrão para desenvolvedores e segurança de aplicações Web. Este documento representa um amplo consenso sobre os riscos de segurança mais críticos para aplicações Web.

É um documento reconhecido globalmente pelos desenvolvedores como o primeiro passo para a codificação mais segura.

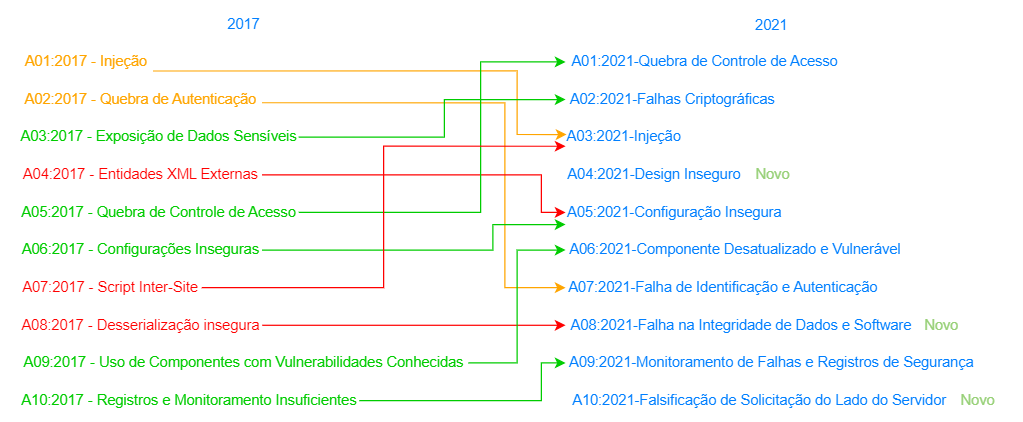


Figure 1. Evolução das Vulnerabilidades OWASP Top 10: Um Comparativo entre 2017 e 2021

### Quebra de Controle de Acesso

Quebra de Controle de Acesso, também conhecida como **A01:2021 - *Broken Access Control*** na lista OWASP Top 10, é uma falha de segurança crítica que ocorre quando um aplicativo web permite que usuários não autorizados acessem recursos ou realizem acções que não deveriam. Isso geralmente acontece devido a falhas na implementação dos mecanismos de autenticação e autorização do aplicativo.

As falhas de controle de acesso podem ter diversas consequências, tais como:

* Divulgação de dados confidenciais;
* Modificação de dados;
* Destruição de dados;
* Execução de funções não autorizadas.

### Falhas Criptográficas

As falhas criptográficas, também conhecidas como **A02:2021 - *Cryptographic Failures*** na lista OWASP Top 10, são um conjunto de vulnerabilidades que podem comprometer a confidencialidade, integridade ou autenticidade dos dados. Elas geralmente surgem devido ao uso inadequado ou implementação incorrecta de algoritmos criptográficos, chaves ou protocolos de segurança.

Os diferentes tipos de falhas criptográficas podem ser:

* **Uso de algoritmos fracos ou obsoletos**: Algoritmos criptográficos mais antigos podem ser vulneráveis a ataques modernos, como ataques de força bruta ou análise criptoanalítica.
* **Chaves criptográficas fracas ou mal gerenciadas**: Chaves criptográficas curtas ou reutilizadas podem ser facilmente quebradas, permitindo que invasores descriptografem dados confidenciais.
* **Falhas na implementação de protocolos criptográficos**: Erros na implementação de protocolos criptográficos, como TLS/SSL, podem permitir ataques como interceptação de dados ou ataques *Man-in-The-Middle*.
* **Uso inadequado de criptografia**: A criptografia pode ser usada de forma incorrecta, como criptografar apenas parte dos dados ou usar um tipo de criptografia inadequado para o tipo de dado.

### Injecção

Injecção, também conhecida como **A03:2021 *- Injection*** na lista OWASP Top 10, é uma falha de segurança crítica que ocorre quando um aplicativo web interpreta dados não confiáveis como parte de um comando ou consulta. Isso pode permitir que um invasor execute comandos indesejados, acesse dados não autorizados ou modifique o comportamento do aplicativo.

Esta vulnerabilidade tem várias consequências, e algumas delas incluem a divulgação de dados confidenciais, a modificação dos dados, a aquisição de controle do sistema, negação de serviço, entre outras consequências.

Uma das medidas de precaução é a validação de dados de entrada de um aplicativo web, uso de APIs de parâmetros preparados, manter o software sempre actualizado e também usar testes de segurança.

### Design Inseguro

Design inseguro, também conhecido como **A04:2021 – *Insecure Design*** na lista OWASP Top 10, é uma categoria ampla que abrange falhas de segurança introduzidas desde as etapas iniciais do projecto e do design de um software. Essas falhas criam vulnerabilidades que podem ser facilmente exploradas por invasores, mesmo que a implementação do código seja perfeita.

No contexto do software, o design inseguro pode se manifestar de diversas maneiras, que podem ser:

* **Falta de modelagem de ameaças**: O design não considera os riscos de segurança potenciais, deixando o software vulnerável a diversos tipos de ataques.
* **Arquitectura inadequada**: A arquitectura do software dificulta a implementação de medidas de segurança eficazes, criando pontos fracos que podem ser explorados por invasores.
* **Falta de princípios de segurança**: O design não segue princípios básicos de segurança, como o princípio do "mínimo privilégio" ou a defesa em profundidade, tornando o software mais susceptível a ataques.
* **Interfaces inseguras**: As interfaces do software não fornecem controles de segurança adequados, permitindo que usuários mal-intencionados realizem acções não autorizadas.

Esta vulnerabilidade tem várias consequências, e algumas delas incluem custos altos de correcção e até perda de reputação.

### Configuração Insegura

Configuração insegura, também conhecida como **A05:2021 - *Security Misconfiguration*** na lista OWASP Top 10, refere-se à prática de configurar softwares, sistemas ou dispositivos de forma inadequada, deixando-os vulneráveis a ataques. Isso geralmente acontece quando as configurações padrão de segurança não são alteradas ou quando configurações incorrectas são feitas manualmente.

As falhas de configuração insegura podem acontecer de diversas maneiras:

* **Uso de senhas padrão ou fracas**: Senhas padrão fáceis de adivinhar ou senhas reutilizadas em diferentes sistemas são alvos fáceis para ataques de força bruta ou credenciais roubadas.
* **Configurações de acesso desnecessárias**: Permitir acesso remoto desnecessário a sistemas ou serviços aumenta a superfície de ataque e facilita a exploração de vulnerabilidades.
* **Desactivação de recursos de segurança**: Desactivar recursos de segurança como firewalls, criptografia ou actualizações automáticas torna o sistema mais vulnerável a ataques.
* **Configurações incorrectas de software**: Configurar software de forma incorrecta, como habilitar serviços desnecessários ou conceder permissões excessivas a usuários, pode criar vulnerabilidades.

Alguns exemplos de consequências que a configuração insegura pode trazer são as seguintes:

* Violação de dados;
* Malwares;
* Ataques de negação de serviço;
* Perda do controle do sistema; entre outros.

### Componente desactualizado e vulnerável

Componentes desactualizados e vulneráveis, também conhecidos como **A06:2021 -** ***Vulnerable and Outdated Components*** na lista OWASP Top 10, referem-se ao uso de software, bibliotecas ou frameworks desactualizados em sistemas ou aplicações web.

Isso geralmente acontece quando os componentes não são actualizados regularmente para as últimas versões, deixando-os vulneráveis a falhas de segurança conhecidas que já foram corrigidas pelos fornecedores.

Ter componentes desactualizados pode levar a diversos problemas, como:

* Vulnerabilidades de segurança;
* Instabilidade e bugs;
* Incompatibilidades; e
* Dificuldades de suporte…

### Falha de identificação e autenticação

Falha de Identificação e Autenticação, também conhecida como A07:2021 - Falhas de Identificação e Autenticação na lista OWASP Top 10, refere-se a falhas nos mecanismos de autenticação e autorização em sistemas e aplicações web.

Isso permite que usuários não autorizados acessem recursos, realizem acções ou assumam identidades que não deveriam.

As falhas de identificação e autenticação podem ter diversas origens:

* **Mecanismos de autenticação fracos**: Senhas fáceis de adivinhar, autenticação de factor único baseada em senhas ou falta de mecanismos multifatores (MFA) facilitam a entrada de invasores.
* **Falhas na implementação de protocolos de autenticação**: Erros na implementação de protocolos como *SAML, OAuth* ou *OpenID Connect* podem criar vulnerabilidades que podem ser exploradas por invasores.
* **Gerenciamento inadequado de contas de usuário**: Senhas reutilizadas, falta de bloqueio de contas após tentativas falhas de login ou redefinição de senha sem medidas de segurança adequadas aumentam o risco de acesso não autorizado.
* **Sessões não seguras**: Sessões de usuário com tempo limite longo, falta de rotação de tokens de autenticação ou armazenamento inadequado de sessões podem permitir que invasores interceptem ou reutilizem sessões para se passar por usuários legítimos.

### Falha na Integridade de Dados e Software

Falha na Integridade de Dados e Software, também conhecida como **A08:2021 - *Software and Data Integrity Failures*** na lista OWASP Top 10, refere-se à falta de mecanismos para garantir a confiabilidade, precisão e consistência dos dados e do software em sistemas e aplicações web.

Isso permite que dados sejam modificados, excluídos ou corrompidos sem autorização, comprometendo a segurança e a confiabilidade do sistema.

As falhas na integridade de dados e software podem ter diversas origens:

* **Falta de mecanismos de validação de dados**: Dados não validados podem ser corrompidos ou modificados acidentalmente ou intencionalmente.
* **Falhas na implementação de controles de acesso**: Controles de acesso inadequados permitem que usuários não autorizados modifiquem dados.
* **Falta de mecanismos de backup e recuperação**: A falta de backups ou mecanismos de recuperação de dados torna o sistema vulnerável à perda de dados em caso de falhas ou ataques.
* **Uso de software desactualizado ou vulnerável**: Software desactualizado pode conter falhas que permitem a modificação indevida de dados.
* **Falhas na implementação de criptografia**: A falta de criptografia nos dados ou a implementação incorrecta de algoritmos criptográficos pode permitir que invasores interceptem e modifiquem dados.

### Monitoramento de Falhas e Registros de Segurança

Monitoramento de Falhas e Registros de Segurança, também conhecido como **A09:2021 -** ***Security Logging and Monitoring Failures*** na lista OWASP Top 10, é uma prática crucial para a segurança da informação, que envolve a colecta, análise e armazenamento de logs de eventos gerados por sistemas, aplicações e dispositivos de rede.

Esses logs contêm informações valiosas sobre as actividades realizadas, acessos, tentativas de acesso falhas, alterações na configuração, erros e outras ocorrências relevantes para a segurança do sistema.

Por exemplo, ao analisar os logs de segurança, é possível identificar o ponto de entrada do ataque, as acções realizadas pelo invasor e os dados que foram acessados ou modificados.

Essa informação crucial permite tomar medidas imediatas para conter o ataque, investigar a causa raiz do problema e implementar medidas preventivas para evitar que ataques semelhantes aconteçam no futuro.

### Falsificação de solicitação do lado do Servidor

Falsificação de Solicitação do Lado do Servidor (SSRF), também conhecida como **A10:2021** - ***Server-Side Request Forgery*** na lista OWASP Top 10, é uma vulnerabilidade de segurança que permite que um invasor mal-intencionado induza um servidor a realizar solicitações não autorizadas em outros sistemas.

Isso geralmente acontece quando um aplicativo web vulnerável permite que um usuário controle os parâmetros de uma solicitação HTTP, como o URL, cabeçalhos ou corpo da solicitação.

As falhas de SSRF podem ter diversas origens:

* **Entrada não validada**: O aplicativo web não valida os parâmetros de entrada do usuário, permitindo que este inclua URLs maliciosos ou outros comandos.
* **Uso de funções de avaliação inseguras**: O aplicativo web utiliza funções de avaliação que permitem ao usuário executar código no servidor, o que pode ser explorado para realizar solicitações não autorizadas.
* **Componentes de terceiros vulneráveis**: O aplicativo web utiliza bibliotecas ou frameworks de terceiros que possuem falhas de SSRF.

## Kali Linux: Uma ferramenta para análise de Vulnerabilidades

O Kali Linux (anteriormente conhecido como ***BackTrack Linux***) é uma distribuição Linux de código aberto baseada no Debian, voltada para Testes Avançados de Penetração e Auditoria de Segurança.

O Kali Linux contém modificações específicas da indústria, bem como várias centenas de ferramentas direccionadas para várias tarefas de Segurança da Informação, como Testes de Penetração, Pesquisa de Segurança, Forense Computacional, Engenharia Reversa, Gerenciamento de Vulnerabilidades e Testes de Red Team.

### Recursos de Kali Linux

* **Amplo conjunto de ferramentas de teste de penetração**: Mais de 600 ferramentas pré-instaladas para tarefas como avaliação de vulnerabilidade, pesquisa de segurança e teste de penetração.
* **Gratuito e código aberto**: Sem taxas de licença e o código fonte está disponível para personalização.
* **Suporte Amplo a Dispositivos**: Funciona em uma variedade de arquitecturas, incluindo ARM e x86.
* **Desenvolvimento focado em segurança**: Desenvolvido em um ambiente seguro com pacotes e repositórios assinados.
* **Personalizável**: Pode ser adaptado a necessidades específicas, incluindo a personalização do kernel.
* **Suporte Multilíngue**: Opera em vários idiomas.
* **Inicialização Live por USB**: Pode ser inicializado a partir de uma unidade USB sem afectar a máquina host.

## OWASP Broken Web Applications

O OWASP *Broken Web Applications Project* (BWA) é um projecto da Open Web Application Security Project (OWASP), uma organização sem fins lucrativos dedicada à segurança de aplicativa web.

O BWA fornece uma máquina virtual contendo vários aplicativos webs deliberadamente vulneráveis. Esses aplicativos podem ser usados para aprender sobre segurança de aplicativa web, praticar técnicas de teste de penetração e avaliar ferramentas de segurança.

### Recursos do OWASP BWA

Desenvolvido pela Open Web Application Security Project (OWASP), uma comunidade global de especialistas em segurança, o OWASP BWA é um conjunto de ferramentas e recursos abrangentes que te permitem navegar na web com mais confiança e protecção.

#### Máquina Virtual com Aplicativos Vulneráveis

A peça central do BWA é a máquina virtual que contém diversos aplicativos webs deliberadamente vulneráveis. Esses aplicativos foram criados para representar falhas de segurança comuns encontradas em aplicações web reais, fornecendo um ambiente seguro e controlado para explorar e testar diferentes tipos de ataques. A máquina virtual inclui:

* **Aplicativos Web**: Uma variedade de aplicativa web com diferentes funcionalidades e vulnerabilidades, abrangendo desde SQL Injection até cross-site scripting (XSS) e autenticação fraca.
* **Níveis de Dificuldade**: Os aplicativos possuem diferentes níveis de dificuldade, permitindo que usuários de todos os níveis de experiência pratiquem e testem suas habilidades.
* **Documentação Detalhada**: Documentação completa para cada aplicativo, descrevendo as vulnerabilidades presentes, seus impactos e como explorá-las.

#### Ferramentas de Segurança

O BWA inclui diversas ferramentas de segurança que podem ser usadas para testar os aplicativos vulneráveis e avaliar sua eficácia na detecção e prevenção de ataques. As ferramentas disponíveis incluem:

* **Scanners de Vulnerabilidades**: Ferramentas automatizadas que analisam os aplicativos em busca de vulnerabilidades conhecidas.
* **Ferramentas de Teste de Penetração**: Ferramentas manuais e automatizadas para testar a segurança dos aplicativos e explorar vulnerabilidades.
* ***Sniffers* de Rede**: Ferramentas que monitoram o tráfego de rede para identificar ataques em andamento.

#### Documentação e Suporte

O BWA fornece documentação extensa e detalhada sobre cada aplicativo vulnerável, suas vulnerabilidades e como explorá-las. Além disso, um fórum online está disponível para que os usuários façam perguntas, compartilhem informações e colaborem com outros profissionais de segurança.

## Ferramentas de análise de vulnerabilidades Web

### Nikto

O Nikto é um scanner de vulnerabilidades de servidor web abrangente e gratuito escrito em Perl. Ele é projectado para identificar e relatar uma ampla gama de problemas de segurança em servidores web comuns, incluindo:

* Servidores e programas perigosos;
* Servidores web vulneráveis;
* Configurações inseguras;
* Problemas de CGI;
* Versões de software desactualizadas.

O Nikto pode ser usado para uma variedade de propósitos, que podem ser:

* **Auditoria de segurança de sites**: O Nikto pode ser usado para realizar uma varredura completa de um site em busca de vulnerabilidades de segurança.
* **Teste de penetração**: O Nikto pode ser usado como parte de um teste de penetração para identificar vulnerabilidades que podem ser exploradas por invasores.
* **Pesquisa de vulnerabilidades**: O Nikto pode ser usado para pesquisar vulnerabilidades em um grande número de sites.
* **Manutenção de segurança**: O Nikto pode ser usado para verificar regularmente um site em busca de novas vulnerabilidades.

A ferramenta Nikto possui várias características que o tornam uma ferramenta valiosa para profissionais de segurança da informação:

* Gratuito e de código aberto;
* Fácil de usar;
* Actualizado regularmente;
* Relatórios detalhados;

### Skipfish

O Skipfish é uma ferramenta gratuita e de código aberto de reconhecimento de segurança de aplicativos web disponível no GitHub e desenvolvida pela Google, criada para pesquisadores de segurança. Ele é projectado para auxiliar na colecta de informações e no teste de segurança de sites e servidores web.

Funciona através de um rastejamento recursivo e sondagens baseadas em dicionário, mapeando interactivamente o site alvo e anotando o mapa resultante com a saída de várias ferramentas, como Nmap, Nikto.

A ferramenta é útil para identificar problemas como:

* **Recursos ocultos**: Páginas da web que não estão facilmente acessíveis por meio de navegação normal ou links conhecidos.
* **Pontos de entrada em potencial**: Directórios e arquivos que podem ser vulneráveis ​​a ataques.
* **Problemas de configuração**: Configurações incorrectas que podem expor o site a riscos.
* **Tecnologias em uso**: Identificar quais frameworks, bibliotecas e ferramentas estão sendo usados ​​no site.

O Skipfish caracterizar-se por ser:

* Código aberto;
* Modular, permite que o usuário personalize a varredura para atender as suas necessidades especificas;
* Fácil de usar;
* Relatórios detalhados.
* Abrangente, pode identificar uma ampla gama de recursos ocultos e vulnerabilidades existentes;
* Extensível, permite que o usuário adicione novos recursos e funcionalidades.

### Wapiti

O Wapiti é um scanner de vulnerabilidades web de código aberto e gratuito escrito em Python. Ele é projectado para identificar e relatar uma ampla gama de problemas de segurança em aplicativos web, que são:

* **Vulnerabilidades de injecção de código**: O Wapiti pode detectar vulnerabilidades de injecção de código, como injecção de SQL, injecção de JavaScript.
* **Vulnerabilidades de cross-site scripting (XSS)**: O Wapiti pode identificar vulnerabilidades de XSS que podem permitir que um invasor injecte scripts maliciosos em um site.
* **Vulnerabilidades de *cross-site request forgery* (CSRF):** O Wapiti pode detectar vulnerabilidades de CSRF que podem permitir que um invasor force um usuário a executar acções indesejadas.
* **Vulnerabilidades de configuração incorrecta**: O Wapiti pode identificar configurações incorrectas em aplicativos web que podem torná-los vulneráveis a ataques.
* **Vulnerabilidades de força bruta**: O Wapiti pode realizar ataques de força bruta para tentar adivinhar credenciais de login e outras informações confidenciais.

A ferramenta Wapiti também apresenta uma variedade de propósitos para quem as usas, estas que são:

* Auditoria de segurança de aplicativos web;
* Teste de penetração;
* Pesquisa de vulnerabilidades;
* Manutenção de segurança.

O Wapiti caracterizar-se por ser:

* Código aberto;
* Modular;
* Fácil de usar;
* Relatórios detalhados.

### OWASP-ZAP

O OWASP ZAP (*Zed Attack Proxy*) é um scanner de vulnerabilidades web gratuito e de código aberto projectado para auxiliar na detecção de uma ampla gama de vulnerabilidades em aplicativos web.

Ele funciona como um proxy intermediário entre o navegador do usuário e o aplicativo web, interceptando e analisando o tráfego para identificar problemas de segurança. O ZAP pode detectar diversos tipos de vulnerabilidades, incluindo:

* **Injecção de código**: *SQL Injection, XSS, XPath Injection*;
* **Cross-site request forgery (CSRF)**;
* **Vulnerabilidades de configuração incorrecta**: Configurações incorrectas em aplicativos web que podem torná-los vulneráveis ​​a ataques.
* **Problemas de autenticação**: Vulnerabilidades em mecanismos de autenticação que podem permitir que invasores acessem contas de usuário sem autorização.
* **Problemas de gerenciamento de sessões**: Vulnerabilidades que permitem que invasores roubem ou sequestrem sessões de usuário.

O OWASP ZAP pode ser usado para uma variedade de propósitos, incluindo:

* Auditoria de segurança de aplicativos web;
* Teste de penetração;
* Pesquisa de vulnerabilidades;
* Manutenção de segurança;
* Desenvolvimento seguro;

O OWASP-ZAP caracterizar-se por ser:

* Código aberto;
* Integração com outras ferramentas;
* Fácil de usar;
* Relatórios detalhados.
* Abrangente;
* Extensível.

### XSSPY

O XssPy é um scanner de vulnerabilidades de Cross-Site Scripting (XSS) gratuito e de código aberto escrito em Python. Ele é projectado para identificar e relatar vulnerabilidades XSS em websites, permitindo que os administradores de sites as corrijam antes que sejam exploradas por invasores.

O XssPy funciona de maneira diferente da maioria dos scanners de XSS. Em vez de apenas verificar uma única página, o XssPy percorre o site inteiro, encontrando todos os links e subdomínios. Em seguida, ele testa cada entrada em cada página em busca de vulnerabilidades XSS. Isso torna o XssPy mais abrangente e preciso do que outros scanners.

O XssPy pode detectar vários tipos de vulnerabilidades XSS, incluindo:

* **XSS reflectida**: Vulnerabilidades que ocorrem quando um parâmetro de URL não é escapado correctamente.
* **XSS armazenada**: Vulnerabilidades que ocorrem quando dados maliciosos são armazenados permanentemente em um servidor web.
* **XSS com base em DOM**: Vulnerabilidades que ocorrem quando dados maliciosos são injectados em um documento HTML via JavaScript.

O XssPy pode ser usado para uma variedade de propósitos, incluindo:

* Auditoria de segurança de websites;
* Teste de penetração;
* Pesquisa de vulnerabilidades;
* Manutenção de segurança.

O XssPy possui várias características que o tornam uma ferramenta valiosa para profissionais de segurança da informação, incluindo:

* Gratuito e de código aberto;
* Fácil de usar;
* Eficiente, é um scanner rápido e eficiente que pode escanear grandes websites rapidamente.
* Preciso, é um scanner preciso que gera poucos falsos positivos.
* Relatórios detalhados;

### W3af

O W3af *(Web Application Attack and Audit Framework*) é um framework de varredura e auditoria de vulnerabilidades web abrangente e gratuito escrito em Python. Ele é projectado para ajudar você a identificar e explorar vulnerabilidades em aplicativos web, permitindo que você as corrija antes que sejam exploradas por invasores.

O W3af possui uma ampla gama de recursos, incluindo:

* **Varredura de vulnerabilidades**: O W3af pode verificar uma variedade de vulnerabilidades em aplicativos web, como injecção de SQL, cross-site scripting (XSS) e vulnerabilidades de configuração incorrecta.
* **Teste de penetração**: O W3af pode ser usado como parte de um teste de penetração para identificar e explorar vulnerabilidades em um aplicativo web.
* **Análise de fuzzing**: O W3af pode ser usado para realizar fuzzing em um aplicativo web, o que envolve enviar dados inesperados para o aplicativo para ver como ele responde.
* **Análise de protocolo**: O W3af pode ser usado para analisar o tráfego entre um navegador web e um aplicativo web.
* **Relatórios**: O W3af pode gerar relatórios detalhados que podem ser usados ​​para analisar os resultados da varredura.

## Ferramentas de análise de vulnerabilidades de Infra-estruturas

### OpenVAS

O OpenVAS (*Open Vulnerability Assessment System*) é um framework de gerenciamento de vulnerabilidades gratuito e de código aberto projectado para auxiliar na identificação e gerenciamento de vulnerabilidades de segurança em sistemas de informação. Ele é composto por vários componentes, incluindo:

* **OpenVAS Scanner**: Um scanner de vulnerabilidades de rede que utiliza NVTs (Network Vulnerability Tests) para verificar a presença de vulnerabilidades em hosts e serviços de rede.
* **Greenbone Management Console**: Uma interface web para gerenciar o OpenVAS Scanner, criar tarefas de varredura, visualizar resultados e gerar relatórios.
* **OpenVAS Feed**: Um banco de dados de NVTs que é actualizado regularmente com novas vulnerabilidades.

O OpenVAS pode ser usado para detectar uma ampla gama de vulnerabilidades, incluindo:

* **Vulnerabilidades de software**: Vulnerabilidades em sistemas operacionais, aplicativos e serviços.
* **Vulnerabilidades de hardware**: Vulnerabilidades em dispositivos de rede e outros dispositivos de hardware.
* **Vulnerabilidades de configuração**: Configurações incorrectas em sistemas de software e dispositivos de hardware que podem torná-los vulneráveis ​​a ataques.

O OpenVAS pode ser usado para uma variedade de propósitos, incluindo:

* Auditoria de segurança de redes;
* Teste de penetração;
* Gerenciamento de vulnerabilidades;
* Desenvolvimento seguro;

O OpenVAS possui várias características que o tornam uma ferramenta valiosa para profissionais de segurança da informação, incluindo: Opensource, gratuito, fácil de usar, abrangente, modular, extensível e com relatórios detalhados.

### Fierce

O Fierce é uma ferramenta de reconhecimento de DNS (Domain Name System) semi-leve projectada para ajudar a localizar e colectar informações sobre subdomínios, endereços IP e servidores de email relacionados a um determinado domínio.

Diferente de outras ferramentas de DNS que se concentram em escanear a internet inteira, o Fierce foca em um domínio específico e seus subdomínios, tornando-o mais eficiente e preciso para identificar recursos ocultos e pontos de entrada em potencial.

O Fierce funciona através de uma combinação de técnicas, incluindo:

* **Transferência de zona DNS**: A ferramenta tenta realizar uma transferência de zona DNS do servidor de nomes autoritativo para o domínio alvo, obtendo assim uma lista completa de subdomínios e registros DNS relacionados.
* **Pesquisa de subdomínios**: O Fierce utiliza dicionários e APIs de pesquisa de subdomínios para identificar subdomínios adicionais que podem não estar presentes na zona DNS.
* **Pesquisa de endereços IP**: A ferramenta tenta resolver os nomes de host dos subdomínios identificados para obter seus endereços IP correspondentes.
* **Identificação de servidores de email**: O Fierce analisa os registros MX (*Mail Exchanger*) da zona DNS para identificar os servidores de email responsáveis pelo domínio alvo.

O Fierce pode ser usado para uma variedade de propósitos, incluindo:

* **Reunindo informações antes de um teste de penetração**: O Fierce pode ser usado para mapear um domínio e identificar subdomínios, endereços IP e servidores de email antes de iniciar um teste de penetração.
* **Descobrindo vulnerabilidades**: O Fierce pode ser usado para identificar subdomínios vulneráveis ​​a ataques, como subdomínios que executam softwares desactualizados ou que possuem configurações incorrectas.
* **Monitorando domínios**: O Fierce pode ser usado para monitorar um domínio em busca de novos subdomínios e alterações nos registros DNS.
* **Pesquisa de inteligência de ameaças**: O Fierce pode ser usado para pesquisar domínios em busca de informações que podem ser usadas por invasores, como endereços de IP maliciosos e servidores de email de phishing.

### Metasploit framework

O Metasploit Framework (MSF) é uma plataforma de exploração de vulnerabilidades e testes de penetração de código aberto e multiplataforma, projectada para auxiliar na identificação, exploração e pós-exploração de vulnerabilidades em sistemas de informação.

O MSF fornece uma estrutura robusta e flexível para realizar testes de penetração, automatizando tarefas complexas e fornecendo uma ampla gama de módulos pré-construídos para explorar vulnerabilidades conhecidas.

O MSF pode ser usado para uma variedade de propósitos, incluindo:

* **Testes de penetração**: O MSF é uma ferramenta essencial para realizar testes de penetração completos e abrangentes em sistemas de informação.
* **Exploração de vulnerabilidades**: O MSF fornece uma ampla gama de exploits para explorar vulnerabilidades conhecidas em sistemas operacionais, aplicativos e serviços.
* **Desenvolvimento de exploits**: O MSF pode ser usado para desenvolver exploits personalizados para explorar novas vulnerabilidades.
* **Análise de malware**: O MSF pode ser usado para analisar malware e identificar exploits que podem ser usados ​​para removê-lo.
* **Pesquisa de segurança**: O MSF pode ser usado para pesquisar novas vulnerabilidades e desenvolver técnicas de exploração.

**Principais componentes do MSF:**

* **Console MSF**: Interface de linha de comando para interacção com o framework.
* **Database MSF**: Armazena informações sobre exploits, hosts, vulnerabilidades e outras entidades.
* **Modules MSF**: Scripts de código Python que implementam exploits, scanners, payloads e outras funcionalidades.
* **Payloads MSF**: Módulos de código que são executados em um sistema vulnerável após a exploração bem-sucedida.
* **Consoles Web MSF**: Interfaces web para gerenciar o framework e visualizar resultados de testes.

### Nmap

O Nmap (*Network Mapper*) é uma ferramenta de varredura de rede gratuita e de código aberto projectada para auxiliar na descoberta de hosts e serviços em uma rede de computadores. Ele é amplamente utilizado por administradores de rede, profissionais de segurança da informação e hackers para uma variedade de propósitos.

O Nmap funciona enviando pacotes IP para hosts em uma rede e analisando as respostas para determinar quais hosts estão activos, quais serviços estão em execução e quais portas estão abertas. O Nmap pode detectar uma ampla gama de serviços, incluindo servidores web, servidores de email, servidores FTP, firewalls e roteadores.

O Nmap é uma ferramenta poderosa que pode ser usada para colectar informações valiosas sobre uma rede, incluindo:

* **Topologia da rede**: O Nmap pode identificar hosts activos, sub-redes e gateways em uma rede.
* **Serviços em execução**: O Nmap pode identificar quais serviços estão em execução em cada host, incluindo o nome do serviço, o número da porta e a versão do software.
* **Portas abertas**: O Nmap pode identificar quais portas estão abertas em cada host, o que pode indicar quais serviços estão vulneráveis ​​a ataques.
* **Sistemas operacionais**: O Nmap pode identificar o sistema operacional em execução em cada host.

O Nmap pode ser usado para uma variedade de propósitos, incluindo:

* **Auditoria de segurança de rede**: O Nmap pode ser usado para realizar uma varredura completa de uma rede em busca de hosts vulneráveis ​​e serviços mal configurados.
* **Teste de penetração**: O Nmap pode ser usado como parte de um teste de penetração para identificar e explorar vulnerabilidades em uma rede.
* **Gerenciamento de rede**: O Nmap pode ser usado para monitorar uma rede e identificar hosts que foram recentemente adicionados ou removidos.
* **Resolução de problemas de rede**: O Nmap pode ser usado para solucionar problemas de rede identificando hosts que não estão respondendo ou que estão executando serviços inesperados.
* **Pesquisa de segurança**: O Nmap pode ser usado para pesquisar novas vulnerabilidades e desenvolver técnicas de exploração.

### Netcat

O Netcat (nc) é uma ferramenta multifuncional de rede de código aberto e multiplataforma, projectada para interagir com hosts e serviços em uma rede. Ele oferece uma interface poderosa e versátil para enviar e receber dados, abrir conexões TCP ou UDP, escanear portas, transferir arquivos e realizar diversas outras tarefas de rede.

O Netcat é frequentemente utilizado por administradores de rede, profissionais de segurança da informação, hackers e entusiastas da tecnologia por sua flexibilidade e simplicidade. Ele pode ser usado em diversas situações, desde testar conexões de rede até criar backdoors e explorar vulnerabilidades.

Principais funcionalidades do Netcat:

* **Comunicação TCP e UDP**: O Netcat pode estabelecer conexões TCP ou UDP com hosts remotos, permitindo a troca de dados bidireccional.
* **Scanner de portas**: O Netcat pode ser utilizado para escanear portas em hosts remotos, identificando quais portas estão abertas e quais serviços estão em execução.
* **Transferência de arquivos**: O Netcat pode ser utilizado para transferir arquivos entre hosts remotos, similar a ferramentas como o FTP.
* ***Port forwarding***: O Netcat pode ser utilizado para encaminhar portas entre hosts, permitindo o acesso a serviços em redes privadas.
* **Criação de backdoors**: O Netcat pode ser utilizado para criar backdoors em hosts remotos, permitindo acesso remoto persistente.
* ***Tunneling* de tráfego**: O Netcat pode ser utilizado para encapsular tráfego de outras aplicações em túneis seguros, mascarando sua origem e destino.

O Netcat pode ser utilizado em diversas situações, como:

* **Teste de conectividade de rede**: Verificar se um host remoto está acessível e qual porta está em uso para um serviço específico.
* **Transferência de arquivos**: Transferir arquivos entre hosts remotos de forma segura e eficiente.
* **Depuração de problemas de rede**: Identificar problemas de conectividade e latência em redes TCP/IP.
* **Exploração de vulnerabilidades**: Testar a vulnerabilidade de hosts remotos a ataques específicos.
* **Criação de ferramentas de rede personalizadas**: Desenvolver ferramentas personalizadas para automatizar tarefas de rede complexas.
* **Teste de intrusão**: Simular ataques de rede para avaliar a segurança de sistemas e redes.

### Unicornscan

O Unicornscan é uma ferramenta de varredura de rede de código aberto e multiplataforma projectada para identificar hosts e serviços em uma rede de forma eficiente e eficaz. Ele se destaca por sua abordagem inovadora que combina técnicas tradicionais de varredura com métodos avançados de evasão de firewall, tornando-o ideal para penetrar em redes protegidas e descobrir hosts ocultos.

Principais características do Unicornscan:

* **Varredura de alta velocidade**: O Unicornscan utiliza técnicas optimizadas para realizar varreduras de rede de forma rápida e eficiente, mesmo em grandes redes.
* **Detecção precisa de hosts**: O Unicornscan emprega diversos métodos para identificar hosts activos, incluindo varredura de portas TCP e UDP, análise de respostas ICMP e técnicas de fingerprinting.
* **Evasão de firewall avançada**: O Unicornscan implementa diversas técnicas para contornar firewalls e outros dispositivos de segurança de rede, aumentando a probabilidade de detectar hosts ocultos.
* **Suporte a IPv4 e IPv6**: O Unicornscan suporta varredura de redes IPv4 e IPv6, permitindo sua utilização em diversos ambientes de rede.
* **Interface de linha de comando intuitiva**: O Unicornscan oferece uma interface de linha de comando simples e fácil de usar, facilitando seu aprendizado e operação.

O Unicornscan pode ser utilizado em diversas situações, como:

* **Teste de penetração**: O Unicornscan é uma ferramenta essencial para realizar testes de penetração em redes, identificando hosts vulneráveis ​​e pontos de entrada em potencial.
* **Auditoria de segurança de rede**: O Unicornscan pode ser utilizado para auditar a segurança de uma rede, identificando hosts e serviços não autorizados e configurações incorrectas.
* **Pesquisa de inteligência de ameaças**: O Unicornscan pode ser utilizado para pesquisar redes em busca de hosts maliciosos ou vulneráveis, auxiliando na prevenção de ataques cibernéticos.
* **Mapeamento de rede**: O Unicornscan pode ser utilizado para mapear uma rede, identificando todos os hosts e serviços activos.
* **Descoberta de hosts ocultos**: O Unicornscan é ideal para identificar hosts ocultos que estão sendo protegidos por firewalls ou outras medidas de segurança.

# Conclusão

A análise de vulnerabilidades é uma etapa crucial na segurança da informação, permitindo identificar e corrigir falhas em sistemas, softwares e redes antes que sejam exploradas por invasores. Através da implementação de medidas de segurança adequadas e da realização periódica de testes de vulnerabilidade, as organizações podem reduzir significativamente o risco de ataques cibernéticos e proteger seus dados e activos.

Este trabalho abordou os principais conceitos relacionados à análise de vulnerabilidades:

**Definição e importância da análise de vulnerabilidades**: A análise de vulnerabilidades é o processo de identificar e avaliar falhas de segurança em sistemas, softwares e redes. Essa prática é fundamental para prevenir ataques cibernéticos e proteger dados confidenciais.

**Técnicas de análise de vulnerabilidades:** Diversas técnicas podem ser utilizadas para realizar a análise de vulnerabilidades, como varreduras de portas, varreduras de vulnerabilidades e testes de invasão.

**OWASP Top 10**: O OWASP Top 10 é uma lista das dez principais vulnerabilidades de segurança em aplicações web, publicada anualmente pela *Open Web Application Security Project* (OWASP). Essa lista serve como referência para profissionais de segurança da informação na identificação e mitigação de riscos em aplicações web.

**Ferramentas de análise de vulnerabilidades**: Diversas ferramentas podem ser utilizadas para auxiliar na análise de vulnerabilidades: Nikto, Skipfish, Wapiti, OWASP ZAP, XssPy, W3af, OpenVAS e Fierce, Nmap, Netcat, Metasploit Framework.

A escolha das ferramentas e técnicas mais adequadas para a análise de vulnerabilidades dependerá das necessidades específicas de cada organização, considerando factores como o tamanho da rede, os tipos de sistemas e softwares utilizados e os recursos disponíveis.

É importante ressaltar que a análise de vulnerabilidades é um processo contínuo que deve ser realizado de forma regular para garantir a protecção da infra-estrutura de TI contra novas ameaças e vulnerabilidades. Além disso, é fundamental que as organizações implementem medidas de segurança adequadas para corrigir as vulnerabilidades identificadas e prevenir ataques cibernéticos.

# Referências bibliográficas

1. Rajan, M. (2019, Maio, 21). Mastering Web Application Security: Conducting In-Depth Vulnerability Assessments with Kali Linux. Medium. <https://medium.com/@maliahrajan/mastering-web-application-security-conducting-in-depth-vulnerability-assessments-with-kali-linux-b88bf64b0a2>, (Acesso em 17 de Abril de 2024)
2. Kali Linux Project. (s.d.). What is Kali Linux? [O que é o Kali Linux?]. kali.org. <https://www.kali.org/docs/introduction/what-is-kali-linux/>, (Acesso em 17 de Abril de 2024)
3. OWASP Broken Web Applications Project. [Projecto OWASP de Aplicações Web Quebradas]. SourceForge.net. Disponível em: <https://sourceforge.net/projects/owaspbwa/files/>, Acesso em: 17 de Abril de 2024.
4. OWASP Community. [Comunidade OWASP]. OWASP Foundation. Disponível em: <https://owasp.org/www-community/>. Acesso em: 17 de Abril de 2024.
5. OWASP Top 10. OWASP Foundation. Disponível em: <https://owasp.org/www-project-top-ten/> .
6. Acesso em: 17 de Abril de 2024.
7. OWASP Top 10 (Português). OWASP Foundation. Disponível em: <https://owasp.org/Top10/pt_BR/> . Acesso em: 17 de Abril de 2024.
8. Artigo: Santos, J. R. (2023). What is Nikto and Use Cases of Nikto. DevOps School. https://www.devopsschool.com/blog/what-is-nikto-and-use-cases-of-nikto/
9. Repositório GitHub: Spinkham, D. (2024). Skipfish. GitHub. <https://github.com/spinkham/skipfish>
10. Artigo: OWASP. (2024). Automated Audit using WAPITI. OWASP. <https://owasp.org/www-community/Automated_Audit_using_WAPITI>
11. Documentação: OWASP. (2024). ZAP Desktop User Guide. OWASP. <https://www.zaproxy.org/docs/desktop/>
12. Repositório GitHub: Faizann24. (2024). XssPy. GitHub. <https://github.com/faizann24/XssPy>
13. Documentação: W3af Team. (2024). W3af Documentation. W3af. <http://docs.w3af.org/>
14. Documentação: Greenbone. (2024). Greenbone Security Assistant Documentation. <https://docs.greenbone.net/>
15. Repositório GitHub: Meschwager, M. (2024). Fierce. GitHub. <https://github.com/mschwager/fierce>
16. Documentação: Nmap Project. (2024). Nmap Documentation. https://nmap.org/docs.html
17. Documentação: Oracle Corporation. (2024). Netcat (nc) Reference Manual. <https://docs.oracle.com/cd/E86824_01/html/E54763/netcat-1.html>
18. Repositório GitHub: Dneufeld, D. (2024). Unicornscan. GitHub. <https://github.com/dneufeld/unicornscan>