ScanSqlBETA1: Guia Completo do Scanner de Segurança Web

Um Manual Detalhado para Usuários e Desenvolvedores

Autor: [Seu Nome]

Data: 14 de Junho de 2025

Disponível no GitHub: https://github.com/seu-usuario/ScanSqlBETA1

1 Introdução: O que é o ScanSqlBETA1.py? [Rocket]

O `ScanSqlBETA1.py` é uma ferramenta de segurança web escrita em Python, projetada para identificar vulnerabilidades em sites de forma automatizada. Suas principais funcionalidades incluem testes para SQL Injection (SQLi), Cross-Site Scripting (XSS), força bruta em formulários de login, descoberta de diretórios ocultos, busca por informações sensíveis e integração com a ferramenta SQLMap. **[Shield]**

A ferramenta possui uma interface gráfica (GUI) construída com `tkinter`, tornando-a acessível até para usuários menos técnicos. Este documento explica como o código funciona, como usá-lo, e como contribuir para seu desenvolvimento. Ele é ideal para pentesters, desenvolvedores e entusiastas de segurança que desejam testar sistemas **com permissão explícita**. **[Warning]**

- **Nota Ética**: Scans não autorizados são ilegais. Use apenas em sistemas onde você tem permissão. **[Warning]**
- **Público-Alvo**: Usuários do GitHub, pentesters iniciantes e avançados, desenvolvedores web.

2 Estrutura do Código [Gear]

O código é organizado em duas classes principais:

2.1 Classe `AdvancedSQLiScanner`

- **Função**: Contém a lógica dos scans.
- **Responsabilidades**:
  - Gerencia requisições HTTP (`requests`).
  - Carrega payloads e wordlists.
  - Executa scans de SQLi, XSS, força bruta, etc.
  - Integra com SQLMap via `subprocess`.
  - Armazena resultados em `final_report`.
- **Principais Métodos**:
  - `scan_form_sqli`: Testa SQL Injection em formulários.
  - `scan_form_xss`: Testa XSS em formulários.
  - `brute_force_login`: Realiza força bruta em logins.
  - `scan_directories`: Busca diretórios ocultos.
  - `scan_page_info`: Coleta informações sensíveis.
  - `deep_scan`: Rastreia links em profundidade.
  - `run_sqlmap_scan`: Executa scans com SQLMap.

2.2 Classe `ScannerGUI`

- **Função**: Fornece uma interface gráfica amigável. **[Computer]**

- **Responsabilidades**:
  - Exibe logs coloridos em tempo real.
  - Gerencia botões, entradas e barra de progresso.
  - Permite salvar relatórios em JSON.
- **Estrutura**:
  - Abas para SQLi/XSS, Força Bruta, Descoberta e SQLMap.
  - Logs com cores: azul (info), verde (sucesso), amarelo (aviso), laranja (erro), vermelho (crítico), ciano (SQLMap).

3 Funcionalidades Detalhadas

3.1 Scan de SQL Injection (SQLi) e XSS [Shield]

- **Descrição**: Testa formulários web para vulnerabilidades de SQLi e XSS injetando payloads.
- **Como Funciona**:
  - Detecta formulários HTML com `BeautifulSoup`.
  - Injeta payloads SQLi (ex.: `' OR '1'='1`) e XSS (ex.: `<script>alert('XSS')</script>`).
  - Analisa respostas para erros SQL, delays ou scripts refletidos.
- **Métodos**: `scan_form_sqli`, `scan_form_xss`.

3.2 Força Bruta em Formulários de Login [Key]

- **Descrição**: Tenta combinações de usuário/senha em formulários.
- **Como Funciona**:
  - Identifica campos de usuário/senha.
  - Usa wordlists para testar credenciais.
  - Verifica respostas para sinais de sucesso (ex.: "Bem-vindo").
- **Método**: `brute_force_login`.

3.3 Descoberta de Diretórios e Arquivos Ocultos

- **Descrição**: Busca caminhos sensíveis (ex.: `/admin`, `.env`).
- **Como Funciona**:
  - Testa URLs com uma wordlist.
  - Verifica códigos HTTP (200, 403).
- **Método**: `scan_directories`.

3.4 Busca por Informações Sensíveis

- **Descrição**: Identifica dados expostos, como emails, senhas ou chaves API.

- **Como Funciona**:
  - Usa expressões regulares para encontrar padrões.
  - Analisa HTML e arquivos JavaScript.
- **Método**: `scan_page_info`.

3.5 Deep Scan (Análise Profunda) ⋆

- **Descrição**: Rastreia links até uma profundidade definida.
- **Como Funciona**:
  - Usa BFS (Breadth-First Search) para visitar URLs.
  - Aplica scans de SQLi, XSS ou busca sensível.
- **Método**: `deep_scan`.

3.6 Integração com SQLMap [Gear]

- **Descrição**: Usa SQLMap para scans avançados de SQLi.
- **Como Funciona**:
  - Constrói comandos SQLMap com opções configuráveis.
  - Analisa saída com expressões regulares.
- **Método**: `run_sqlmap_scan`.

3.7 Interface Gráfica (GUI) [Computer]

- **Descrição**: Interface amigável com abas e logs coloridos.
- **Como Funciona**:
  - Usa `tkinter` com `ttk` para visual moderno.
  - Atualiza progresso via `queue.Queue`.

4 Como Usar o ScanSqlBETA1.py?

4.1 Pré-requisitos [Gear]

1. Instale Python 3.8+.
2. Instale dependências:

```
pip install requests beautifulsoup4 colorama
```

3. (Opcional) Instale SQLMap:
   - Baixe em https://github.com/sqlmapproject/sqlmap.
   - Adicione ao PATH ou especifique na GUI.
4. Arquivos padrão (`sql_payloads.json`, `common.txt`, etc.) são criados automaticamente se não existirem.

4.2 Passos para Executar [Rocket]

1. Clone o repositório:

```
git clone https://github.com/seu-usuario/ScanSqlBETA1.git
cd ScanSqlBETA1
```

2. Execute o script:

```
python ScanSqlBETA1.py
```

3. Use a GUI:
   - **SQLi/XSS**: Insira URL, marque/desmarque XSS, clique em "Iniciar".
   - **Força Bruta**: Insira URL, detecte campos ou insira manualmente, escolha wordlist, clique em "Iniciar".
   - **Descoberta**: Insira URL, escolha opções (ex.: Deep Scan), clique em "Iniciar".
   - **SQLMap**: Insira URL, configure opções, clique em "Iniciar".
4. Visualize logs coloridos e salve o relatório em JSON.

4.3 Dicas Importantes [Warning]

- **Permissões**: Use apenas em sistemas autorizados. **[Warning]**
- **Wordlists**: Use `rockyou.txt` ou `dirb` para melhores resultados.
- **JavaScript**: Formulários dinâmicos requerem entrada manual de campos.

5 Como o Código Funciona Internamente? [Gear]

- **Payloads/Wordlists**: Carregados de arquivos JSON/txt com padrões se ausentes.
- **Detecção de Formulários**: Usa `BeautifulSoup` para parsear HTML.
- **Testes**: Injeta payloads e analisa respostas (erros, delays, reflexão).
- **Multithreading**: Usa `ThreadPoolExecutor` para scans rápidos.
- **Relatórios**: Armazena em `final_report` e exibe logs coloridos.
- **SQLMap**: Executa via `subprocess` e parseia saída.
- **GUI**: Atualiza via filas e `tkinter`.

6 Limitações e Avisos [Warning]

- **Formulários JavaScript**: Não detecta formulários dinâmicos automaticamente.
- **Falsos Positivos/Negativos**: Valide resultados manualmente.
- **Impacto no Servidor**: Evite sobrecarga com scans intensos.
- **Uso Ético**: Scans não autorizados são crimes. **[Warning]**

7 Como Contribuir? [Handshake]

- **Ideias**:
  - Suporte a formulários JavaScript com `selenium`.
  - Mais payloads para XSS ou outras vulnerabilidades.
  - Otimização com proxies ou delays adaptativos.
  - Adicionar scans para CSRF, SSRF, etc.
  - Melhorar documentação com capturas de tela. **[Camera]**
- **Passos**:
  - Faça um fork do repositório.
  - Crie uma branch (`git checkout -b feature/nova-funcionalidade`).
  - Envie um pull request.

8 Exemplo de README para GitHub

```
# ScanSqlBETA1 - Ferramenta de Segurança Web \shield

Uma ferramenta Python para escanear vulnerabilidades web,
   incluindo SQL Injection, XSS, força bruta, descoberta de
   diretórios, e busca por informações sensíveis.

\warning **AVISO**: Use apenas em sistemas autorizados!

## Funcionalidades \star
- SQLi/XSS: Testa formulários para injeções.
- Força Bruta: Tenta credenciais em logins.
- Descoberta: Busca diretórios ocultos.
- Info Sensível: Identifica dados expostos.
- Deep Scan: Rastreia links.
- SQLMap: Scans avançados de SQLi.

## Instalação \gear
1. Clone o repositório:
   ```bash
   git clone https://github.com/seu-usuario/ScanSqlBETA1.git
   cd ScanSqlBETA1
   ```
2. Instale dependências:
   ```bash
   pip install requests beautifulsoup4 colorama
   ```
3. (Opcional) Instale SQLMap.

Como Usar \rocket
1. Execute:
   ```bash
```
```

```
   python ScanSqlBETA1.py
   ```
2. Na GUI, insira URLs e configure scans.
3. Salve o relatório em JSON.

Contribuindo \handshake
- Faça um fork e envie pull requests.
- Sugira melhorias.

Licença \document
MIT (ou sua escolha)

\warning **Uso Ético**: Scans não autorizados são crimes!
```

## 9 Conclusão ⋆

O `ScanSqlBETA1.py` é uma ferramenta poderosa para testes de segurança web, ideal para sistemas autorizados. Com GUI intuitiva, integração com SQLMap e suporte a múltiplos scans, é perfeito para pentesters e desenvolvedores. Adicione este documento e capturas de tela ao seu GitHub para compartilhar com a comunidade! **[Rocket]**
```