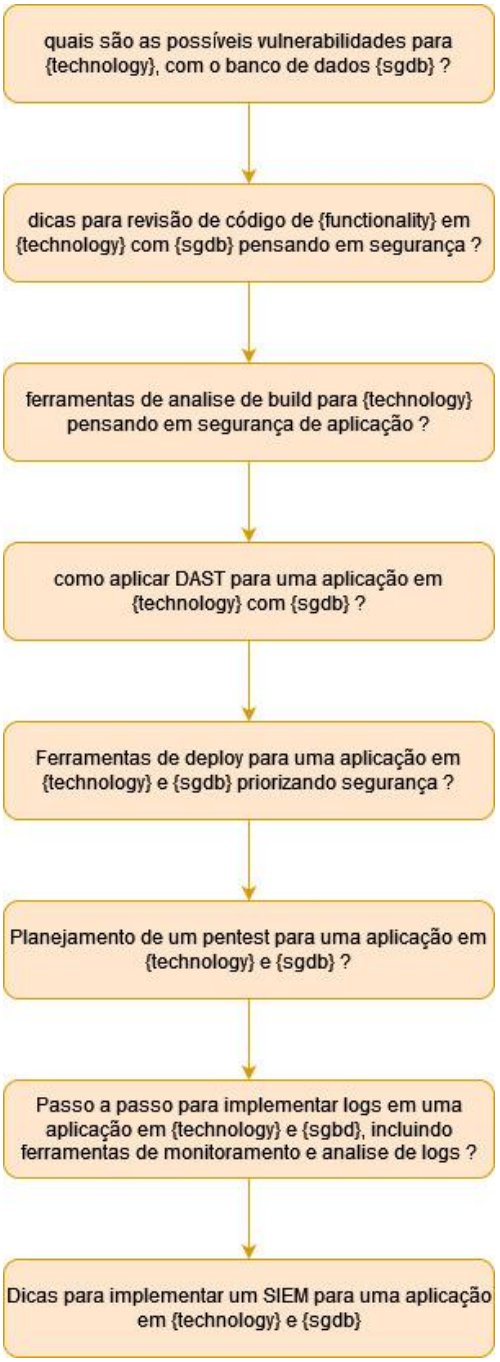
Safee Madarah

Especificação Adicional

Introdução

Esse software foi desenvolvido para fazer perguntas específicas definidas pelos desenvolvedores. Ele utiliza os parâmetros informados pelos usuários para ajudar a criar um sistema seguro, passando por todas as etapas necessárias para um desenvolvimento seguro de software, seguindo a metodologia conhecida como "DevSecOps".



Especificação

---

1. **DevSecOps - Planejamento**

Durante a fase de planejamento, é importante considerar como a inteligência artificial pode ajudar a mitigar possíveis vulnerabilidades futuras.Uma pergunta chave que pode ser feita é:

"quais são as possíveis vulnerabilidades para {technology}, com o banco de dados {sgdb}"

Observação: **technology** e **sgdb** são parâmetros que o usuário nos fornecerá para formarmos a pergunta completa.

```python
@service.route('/api/v1/service/plan', methods=['GET'])
def plan():
    try:
        query_params = request.args

        technology = query_params["technology"]
        sgdb = query_params["sgdb"]

        prompt = f"quais são as possíveis vulnerabilidades para {technology}, com o banco de dados {sgdb}"
        response = return_json(prompt)

        return jsonify(response)
    except Exception as e:
        return {"content": e.message}
```

2. **DevSecOps - Code**

Durante a fase de code, é importante considerar como a inteligência artificial pode ajudar a mitigar possíveis vulnerabilidades futuras.Uma pergunta chave que pode ser feita é:

"dicas para revisão de código de {functionality} em {technology} com {sgdb} pensando em segurança"

Observação: **technology** e **sgdb** são parâmetros que o usuário nos fornecerá para formarmos a pergunta completa.

```python
@service.route('/api/v1/service/code', methods=['GET'])
def code():
    try:
        query_params = request.args

        functionality = query_params["functionality"]
        technology = query_params["technology"]
        sgdb = query_params["sgdb"]

        prompt = f"dicas para revisão de código de {functionality} em {technology} com {sgdb} pensando em segurança"

        response = return_json(prompt)

        return jsonify(response)
    except Exception as e:
        return {"content": e.message}
```

3. **DevSecOps - Build**

Durante a fase de build, é importante considerar como a inteligência artificial pode ajudar a mitigar possíveis vulnerabilidades futuras.Uma pergunta chave que pode ser feita é:

"ferramentas de análise de build para {technology} pensando em segurança de aplicação"

Observação: **technology** e **sgdb** são parâmetros que o usuário nos fornecerá para formarmos a pergunta completa.

```python
@service.route('/api/v1/service/build', methods=['GET'])
def build():
    try:
        query_params = request.args
        technology = query_params["technology"]

        prompt = f"ferramentas de analise de build para {technology} pensando em segurança de aplicação"
        response = return_json(prompt)

        return jsonify(response)
    except Exception as e:
        return {"content": e.message}
```

4. **DevSecOps - Test**

Durante a fase de test, é importante considerar como a inteligência artificial pode ajudar a mitigar possíveis vulnerabilidades futuras.Uma pergunta chave que pode ser feita é:

"como aplicar DAST para uma aplicação em {technology} com {sgdb}"

Observação: **technology** e **sgdb** são parâmetros que o usuário nos fornecerá para formarmos a pergunta completa.

```python
@service.route('/api/v1/service/test', methods=['GET'])
def teste():
    try:
        query_params = request.args

        technology = query_params["technology"]
        sgdb = query_params["sgdb"]

        prompt = f"como aplicar DAST para uma aplicação em {technology} com {sgdb}"
        response = return_json(prompt)

        return jsonify(response)
    except Exception as e:
        return {"content": e.message}
```

5. **DevSecOps - Deploy**

Durante a fase de deploy, é importante considerar como a inteligência artificial pode ajudar a mitigar possíveis vulnerabilidades futuras.Uma pergunta chave que pode ser feita é:

"Ferramentas de deploy para uma aplicação em {technology} e {sgdb} priorizando segurança"

Observação: **technology** e **sgdb** não parametros que o usuário nos fornecerá para formarmos a pergunta completa.

```python
@service.route('/api/v1/service/deploy', methods=['GET'])
def deploy():
    try:
        query_params = request.args

        technology = query_params["technology"]
        sgdb = query_params["sgdb"]

        prompt = f"Ferramentas de deploy para uma aplicação em {technology} e {sgdb} priorizando segurança"
        response = return_json(prompt)

        return jsonify(response)
    except Exception as e:
            return {"content": e.message}
```

6. **DevSecOps - Release**

Durante a fase de release, é importante considerar como a inteligência artificial pode ajudar a mitigar possíveis vulnerabilidades futuras.Uma pergunta chave que pode ser feita é:

"Planejamento de um pentest para uma aplicação em {technology} e {sgdb}"

Observação: **technology** e **sgdb** não parametros que o usuário nos fornecerá para formarmos a pergunta completa.

```python
@service.route('/api/v1/service/release', methods=['GET'])
def release():
    try:
        query_params = request.args

        technology = query_params["technology"]
        sgdb = query_params["sgdb"]

        prompt = f"Planejamento de um pentest para uma aplicação em {technology} e {sgdb}"
        response = return_json(prompt)

        return jsonify(response)
    except Exception as e:
        return {"content": e.message}
```

7. **DevSecOps - Operate**

Durante a fase de operate, é importante considerar como a inteligência artificial pode ajudar a mitigar possíveis vulnerabilidades futuras.Uma pergunta chave que pode ser feita é:

"Passo a passo para implementar logs em uma aplicação em {technology} e {sgbd}, incluindo ferramentas de monitoramento e análise de logs"

Observação: **technology** e **sgdb** não parametros que o usuário nos fornecerá para formarmos a pergunta completa.

```python
@service.route('/api/v1/service/operate', methods=['GET'])
def operate():
    try:
        query_params = request.args

        technology = query_params["technology"]
        sgbd = query_params["sgbd"]

        prompt = f"Passo a passo para implementar logs em uma aplicação em {technology} e {sgbd}, incluindo ferramentas de monitoramento e analise de logs"
        response = return_json(prompt)

        return jsonify(response)
    except Exception as e:
        return {"content": e.message}
```

8. **DevSecOps - Monitor**

Durante a fase de operate, é importante considerar como a inteligência artificial pode ajudar a mitigar possíveis vulnerabilidades futuras.Uma pergunta chave que pode ser feita é:

"Dicas para implementar um SIEM para uma aplicação em {technology} e {sgdb}"

Observação: **technology** e **sgdb** não parametros que o usuário nos fornecerá para formarmos a pergunta completa.

```python
@service.route('/api/v1/service/monitor', methods=['GET'])
def monitor():
    try:
        query_params = request.args

        technology = query_params["technology"]
        sgdb = query_params["sgdb"]

        prompt = f"Dicas para implementar um SIEM para uma aplicação em {technology} e {sgdb}"
        response = return_json(prompt)

        return jsonify(response)
    except Exception as e:
        return {"content": e.message}
```