**Projeto de programa de Fuzzing para Identificação de Subdomínios e Caminhos**

**DISCENTES:**

**ANDRÉ PHELIPE BARBOSA GOULART**

**THEO MENEZES DE SOUSA GUERRA**

**PROFESSOR:**

**PROF.DR. RONALDO SANTOS**

**UNIVERSIDADE ESTÁCIO DE SÁ**

**CAMPUS NOVA AMÉRICA**

**2025.01**

**Rio de Janeiro - RJ**

Sumário

[**1.** DIAGNÓSTICO E TEORIZAÇÃO 3](#_Toc195541471)

[1.1. Idealização do projeto : 3](#_Toc195541472)

[1.2. Problemática e/ou problemas identificados: 3](#_Toc195541473)

[1.3. Justificativa : 3](#_Toc195541474)

[1.4. Objetivos a serem alcançados : 3](#_Toc195541475)

[1.5. Referencial teórico (subsídio teórico para propositura de ações da extensão): 3](#_Toc195541476)

[2. PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO 4](#_Toc195541477)

[2.1. Grupo de trabalho (descrição da responsabilidade de cada membro): 4](#_Toc195541478)

[2.2. Divisão da estrutura do código: 5](#_Toc195541479)

[2.3. Funções do código: 5](#_Toc195541480)

[2.4. Detalhamento técnico do projeto: 6](#_Toc195541481)

[2.5. Evidência técnica: 6](#_Toc195541482)

[3.USO DA MODELAGEM RAD 7](#_Toc195541483)

[3.1. Como o modelo RAD foi eficaz e útil no projeto: 7](#_Toc195541484)

[3.2. Etapas do RAD no Projeto 7](#_Toc195541485)

[3.3. Resultados obtidos 7](#_Toc195541486)

[4.ENCERRAMENTO DO PROJETO 8](#_Toc195541487)

[4.1 Relato Coletivo: 8](#_Toc195541488)

[4.1.1 CONTEXTUALIZAÇÃO: 8](#_Toc195541489)

# DIAGNÓSTICO E TEORIZAÇÃO

## Idealização do projeto :

Esse projeto visa beneficiar diversas partes interessadas, sendo elas, os alunos da Universidade Estácio de Sá, André Phelipe Barbosa Goulart e Theo Menezes de Souza Guerra, assim como também os atuantes da área de segurança, fazendo com que a verificação de veracidade de subdomínios e fuzzings seja mais pratica e segura para os atuantes da área.

## Problemática e/ou problemas identificados:

Com o número crescente de acessos digitais, a segurança cibernética virou uma questão essencial para o mundo inteiro. Porém, muitas pequenas empresas ainda enfrentam desafios para adotar medidas eficientes, principalmente pela falta de recursos. Este projeto quer simplificar isso, trazendo soluções práticas e acessíveis para proteger dados e garantir que essas empresas continuem funcionando sem preocupações.

## Justificativa :

Este projeto é fundamental para trazer soluções simples e acessíveis que ajudam pequenas empresas a proteger seus dados sem estourar seus recursos limitados. Assim, elas conseguem reduzir riscos e continuar operando com segurança, mesmo diante dos desafios do mundo digital.

## Objetivos a serem alcançados :

Identificação de subdomínios e caminhos vulneráveis, permitindo ações preventivas.

Implementação de soluções práticas que minimizem riscos de ataques.

Maior conscientização dos públicos envolvidos sobre a importância da segurança cibernética

## Referencial teórico (subsídio teórico para propositura de ações da extensão):

Para referencial teórico pegamos alguns autores como referência:

-David Kim e Michael G. Solomon em ”Fundamentos de Segurança de Sistemas de Informação"

-Richard A. Clarke em “Guerra cibernética"

-Justin Seitz e Tim Arnold em “Black Hat Python”

# PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO

Um dos maiores apoios que tivemos para o nosso plano de estudos foi a plataforma "Sava" da Estácio, nossa sala de aula virtual. Nela, conseguimos acompanhar as etapas do projeto, além de poder contar com as aulas do professor Ronaldo Santos para esclarecer dúvidas e questionamentos. Também tivemos à disposição os computadores pessoais e os laboratórios da Estácio, o que facilitou muito o desenvolvimento do trabalho.

Com esse suporte, conseguimos criar o nosso plano de projeto, que seguiu as seguintes etapas:

**Planejamento Inicial**: Identificamos as necessidades do projeto e definimos seus objetivos, além de estabelecer o foco das soluções a serem desenvolvidas.

**Pesquisa de Ferramentas**: Pesquisamos as tecnologias disponíveis e avaliamos quais ofereciam o melhor equilíbrio entre eficiência e custo. A partir disso, escolhemos os recursos mais adequados para a execução.

**Desenvolvimento**: Criamos o sistema com base nas ferramentas escolhidas, priorizando funcionalidade e simplicidade. Realizamos testes para garantir que ele atendesse às necessidades do projeto.

**Validação e Ajustes**: Aplicamos testes em cenários reais e ajustamos o sistema conforme os resultados obtidos. Melhoramos a solução com base nas análises e no feedback que recebemos.

**Preparação para o Uso**: Documentamos o sistema e criamos materiais explicativos para facilitar seu uso. Também preparamos orientações e guias para os públicos envolvidos.

**Entrega e Acompanhamento**: Disponibilizamos a solução para os interessados e observamos o impacto. Continuamos a buscar oportunidades de aprimoramento para futuras melhorias.

## Grupo de trabalho (descrição da responsabilidade de cada membro):

Theo Menezes de Sousa Guerra e André Phelipe Barbosa Goulart dividiram igualmente as responsabilidades no projeto:

**Pesquisa e Planejamento**: Juntos, fizeram uma pesquisa sobre segurança cibernética e definiram o escopo do projeto.

**Desenvolvimento do Sistema**: Ambos se encarregaram de implementar as funcionalidades do sistema de fuzzing para subdomínios e caminhos vulneráveis.

**Testes e Validações**: Realizaram testes práticos e ajustaram o sistema com base nos resultados obtidos.

**Documentação**: Trabalharam na organização e elaboração da documentação, explicando o processo e os resultados.

Ambos colaboraram em todas as etapas, garantindo que o projeto fosse bem-sucedido e alinhado.

## Divisão da estrutura do código:

**Enumeração de Subdomínios** No começo, o script tenta descobrir subdomínios que estão ligados ao domínio principal. Ele usa uma lista de palavras (wordlist), que é basicamente um arquivo de texto com várias opções de subdomínios para testar. O processo funciona assim: o código pega cada palavra da lista, combina com o domínio (tipo "teste.dominio.com") e faz uma tentativa de acessar esse endereço com uma requisição HTTP. Se o subdomínio funcionar e responder com sucesso (status 200), ele entra na lista de subdomínios encontrados e aparece na tela. Essa parte é muito útil para descobrir áreas ativas no site que não são tão óbvias.

**Fuzzing de diretórios** Depois disso, vem a vez de explorar os caminhos internos do domínio. Aqui o código também usa uma wordlist, mas com possíveis rotas internas (como "admin" ou "login"). Para cada caminho, ele monta uma URL e tenta acessar (tipo dominio.com/admin). Quando o caminho está ativo e retorna sucesso, ele é registrado e mostrado. Isso ajuda a mapear partes escondidas do site, que podem incluir páginas administrativas ou de login.

**Armazenamento dos Resultados** Depois de achar subdomínios e caminhos válidos, o script salva tudo num arquivo de texto, usando a função de armazenamento. Isso garante que os dados fiquem organizados e disponíveis para consulta mais tarde, sem risco de perda.

**Coordenação do Processo** Por último, a função principal (main) organiza tudo. Ela recebe as informações do usuário, como o domínio e as listas de palavras, e coordena as etapas do processo. E tem mais: se o usuário decidir parar o script de repente (com Ctrl+C, por exemplo), ele salva os resultados parciais antes de ser finalizado, evitando que informações importantes sejam perdidas.

## Funções do código:

O código foi dividido em partes para que cada função tivesse um papel específico dentro do projeto. As principais funções são:

**fuzz\_subdomains**: Essa função cuida de testar subdomínios associados ao domínio principal. Ela usa uma lista de palavras para criar combinações e tenta acessar cada uma delas. Se o subdomínio estiver ativo, ele é identificado e salvo.

**fuzz\_paths**: Parecida com a anterior, mas voltada para caminhos dentro do site. Aqui, o código testa várias possibilidades de URLs para encontrar quais estão funcionando.

**save\_results**: Depois de descobrir subdomínios e caminhos válidos, essa função organiza e salva os dados em um arquivo para que fiquem disponíveis para consulta.

**main**: Essa função é a responsável por coordenar tudo. É ela que define a ordem de execução e lida com os parâmetros fornecidos pelo usuário. Também garante que, se o processo for interrompido, os resultados sejam salvos.

Os nomes das funções refletem bem o que cada uma faz, deixando claro para quem for ler ou mexer no código. Tudo foi dividido de forma lógica para que cada parte fosse independente e fácil de entender.

## Detalhamento técnico do projeto:

O projeto foi desenvolvido utilizando ferramentas e tecnologias que garantiram eficiência e resultados práticos. Abaixo, estão as principais etapas e decisões técnicas:

**Linguagem de Programação**: Optamos por **Python** pela sua simplicidade e eficiência, o que facilitou tanto o desenvolvimento quanto os testes das funcionalidades.

**Bibliotecas e Ferramentas**: A biblioteca **requests** foi escolhida para fazer as requisições HTTP, permitindo a comunicação com os hosts de forma eficiente. Já o **argparse** proporcionou uma interação mais flexível com o sistema, com argumentos sendo passados diretamente via CLI(Linha de comando)

**Estrutura do Código**: O código foi organizado em módulos, com funções específicas para:

-Testar subdomínios e caminhos.

-Salvar os resultados encontrados em arquivos de texto.

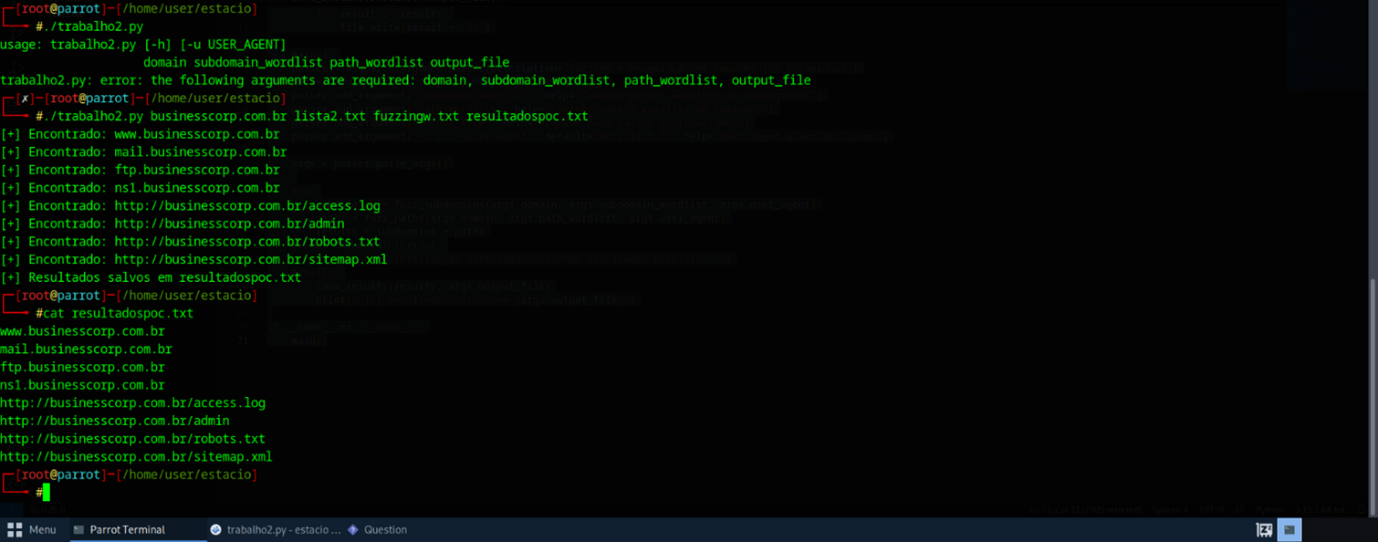
-Realizar testes durante o desenvolvimento, ajustando as funcionalidades conforme necessário.

-Mudar o user-agent se necessário.

**Ambiente de Trabalho**: O desenvolvimento e os testes ocorreram tanto em computadores pessoais quanto nos laboratórios da Estácio, garantindo que o sistema funcionasse bem em diferentes ambientes.

**Validação e Melhorias**: Realizamos testes com várias listas de palavras para garantir que o sistema fosse capaz de identificar subdomínios e caminhos válidos. Além disso, fizemos melhorias contínuas com base nos resultados obtidos.

## Evidência técnica:



# 3.USO DA MODELAGEM RAD

## Como o modelo RAD foi eficaz e útil no projeto:

O modelo RAD foi muito útil para o projeto, principalmente porque prioriza um desenvolvimento rápido e flexível. Ele nos permitiu criar e testar protótipos em ciclos curtos, o que facilitou ajustes conforme surgiam novas necessidades durante o processo.

**Adaptação Rápida**: Conseguimos implementar mudanças contínuas, garantindo que o programa permanecesse alinhado com as expectativas ao longo do desenvolvimento.

**Interação Constante**: As revisões frequentes ajudaram a manter o alinhamento constante com os objetivos do projeto e as necessidades reais do contexto.

**Velocidade no Desenvolvimento**: O modelo acelerou o tempo entre a concepção inicial da ideia e a entrega de funcionalidades práticas e funcionais.

**Eficiência de Recursos**: Identificar rapidamente os ajustes necessários nos ajudou a evitar desperdícios, tornando o processo mais eficiente e focado no que realmente importava.

## Etapas do RAD no Projeto

**Definição de Requisitos**: Identificamos os objetivos principais, como a identificação de subdomínios e caminhos vulneráveis e o salvamento dos resultados.

**Prototipagem Rápida**: Criamos versões iniciais funcionais para testar as principais funções, como fuzzing de subdomínios e caminhos.

**Feedback e Ajustes**: Testamos os protótipos e fizemos melhorias constantes com base nos resultados obtidos.

**Desenvolvimento Modular**: Dividimos o código em funções específicas, como teste de subdomínios, caminhos e salvamento de resultados.

**Validação Final**: Realizamos testes exaustivos para garantir a eficácia do sistema.

**Entrega**: Documentamos tudo e entregamos uma solução funcional e prática.

## Resultados obtidos

Nos testes realizados, o sistema identificou diversos subdomínios e caminhos válidos que poderiam ser explorados por invasores. Por exemplo:

**Subdomínios encontrados**: admin.domain.com, mail.domain.com.

**Caminhos identificados**: /login, /dashboard. Esses resultados demonstram a eficácia do sistema em mapear áreas vulneráveis com precisão.

# 4.ENCERRAMENTO DO PROJETO

## Relato Coletivo:

O encerramento do projeto reflete o esforço coletivo e o compromisso de todos. Enfrentamos desafios, mas, com trabalho em equipe e ajustes rápidos, conseguimos superá-los. O modelo RAD foi fundamental para entregas ágeis e adaptações conforme as necessidades surgiam.

O projeto não só atendeu às expectativas, mas também trouxe um grande aprendizado na área de segurança cibernética. O resultado final é fruto do empenho de todos, cumprindo os objetivos estabelecidos.

### CONTEXTUALIZAÇÃO:

O projeto teve como objetivo criar um programa capaz de identificar vulnerabilidades em domínios, utilizando enumeração de subdomínios e fuzzing respectivamente para subdomínios e caminhos. Desde o início, enfrentamos desafios que exigiram planejamento, pesquisa e ajustes constantes ao longo do processo.

Foi uma experiência que proporcionou muito aprendizado, tanto na parte técnica, quanto no trabalho em equipe. Conseguimos avançar juntos, superando dificuldades e alcançando os objetivos propostos de forma eficiente e alinhada.