# SANTANDER BOOTCAMP CIBERSEGURANÇA

## Explicação Código - Enumeração

### 1. Criação do Dicionário port_services

```
port_services = {
        21: "FTP", # Serviço de transferência de arquivos
        22: "SSH", # Secure Shell (acesso remoto seguro)
        ...
 }
```

**O que fiz aqui:**
Criei um dicionário que mapeia **números de portas** (as chaves) aos seus respectivos serviços conhecidos (os valores). Esse dicionário facilita a consulta quando preciso saber qual serviço está associado a uma porta específica.

**Exemplo:**Se eu quiser saber o serviço associado à porta 22, só preciso consultar **port_services[22]**, e ele me retornará **"SSH"**.

### 2. Função enumerate_services

```
def enumerate_services(ports):
        services = []
        for port in ports:
                if port in port_services:
                        services.append(port_services[port])
                else:
                        services.append("Desconhecido")
        return services
```

**O que fiz aqui:**Implementei uma função que pega uma lista de portas (ex.: **[22, 80, 9999]**) e retorna uma lista com os nomes dos serviços correspondentes.

- **Detalhes do processo:**

    1. Inicializo uma lista vazia chamada **services** para armazenar os nomes dos serviços.
    2. Percorro cada porta da lista **ports**.
    3. Verifico se a porta está no dicionário **port_services**:
        - Se estiver, adiciono o serviço correspondente à lista **services**.
        - Se não estiver, adiciono **"Desconhecido"**.
    4. No final, retorno a lista **services**.

**Exemplo de uso:**
Se eu passar **[22, 80, 9999]**, o resultado será **["SSH", "HTTP", "Desconhecido"]**.

### 3. Função Principal main

```
def main():
        ports_input = input()
        ports = [int(port) for port in ports_input.strip().split(",")]
        services = enumerate_services(ports)
        print(services)
```

**O que fiz aqui:**Implementei uma função para interagir com o usuário, processar a entrada de dados e exibir o resultado final.

- **Passo a passo:**

    1. **Entrada do usuário:** Usei **input()** para capturar as portas fornecidas pelo usuário como uma string. O formato esperado é algo como **22,80,9999**.
    2. **Processamento da entrada:** Transformo a string em uma lista de números inteiros:
        - **strip()** remove espaços em branco desnecessários.
        - **split(",")** separa a string em partes usando a vírgula como delimitador.
        - **int(port)** converte cada parte para número.
    3. **Chamada da função:** Passei a lista de portas para a função **enumerate_services** para obter os serviços correspondentes.
    4. **Saída do resultado:** Usei **print(services)** para exibir a lista de serviços.

### 4. Execução do Programa

```
if __name__ == "__main__": main()
```

**O que fiz aqui:**Essa linha garante que a função **main** será executada apenas se o script for rodado diretamente (e não importado como módulo em outro programa).

### Resumo do Funcionamento Completo

1. O programa pede ao usuário uma lista de portas, separadas por vírgulas.
    - Exemplo de entrada: **22,80,9999**.
2. Ele transforma essa entrada em uma lista de números inteiros: **[22, 80, 9999]**.
3. Verifica o serviço correspondente para cada porta, consultando o dicionário **port_services**.
4. Se o serviço for encontrado, retorna o nome; caso contrário, retorna **"Desconhecido"**.
5. Mostra a lista final de serviços na tela.

### Exemplo Prático

Entrada do usuário:

```
22,80,443,9999
```

Processamento interno:

- Converte para: **[22, 80, 443, 9999]**
- Associa serviços:
    - **22** → "SSH"
    - **80** → "HTTP"
    - **443** → "HTTPS"
    - **9999** → "Desconhecido"

Saída do programa:

```
['SSH', 'HTTP', 'HTTPS', 'Desconhecido']
```

### Objetivo do Código

Este programa foi desenvolvido para ajudar na **identificação de serviços associados a números de portas**. Ele permite que usuários, especialmente profissionais ou estudantes da área de redes e segurança da informação, façam consultas rápidas para verificar serviços conhecidos e identifiquem portas desconhecidas.

### Benefícios do Código

1. **Agilidade na consulta**:Facilita a identificação de serviços sem precisar buscar manualmente em tabelas ou documentações.
2. **Praticidade**:Com uma entrada simples (lista de portas), o programa retorna resultados claros e organizados.
3. **Flexibilidade**:Pode ser facilmente adaptado ou ampliado com mais serviços, tornando-o útil em diversas situações.
4. **Educação**:Ideal para estudantes de redes e segurança da informação, pois conecta a teoria (números de portas e seus serviços) à prática.
5. **Diagnóstico Rápido**:Útil em análises iniciais de segurança, ajudando a identificar portas que podem estar em uso, mas sem um serviço conhecido mapeado.