PROCESSO DE DECRIPTAÇÃO DO ARQUIVO “arquivo-strong-9.in-full.hex”

Aluno: Pedro Lucas Moraes de Sousa Rosa

Professor: Wewerton Luis da Costa Cordeiro

A implementação utilizada foi um ataque de força bruta focado e otimizado, baseado em algumas informações conhecidas:

1ª Informação conhecida critica:

- A chave começa com "Security00".
- O texto decifrado pode ou não estar legível em ASCII.
- O algoritmo usado é o AES no modo ECB.

2ª Estratégia de força bruta otimizada:

- Ao invés de efetuar os testes de chave, foi necessário apenas converter o arquivo hex para utf-8:
    - Criado um código para conversão do arquivo hex para vários tipos de enconding, fazendo com que chegasse ao resultado da conversão em UTF-8, conforme o print abaixo.

```python
def decode_hex_to_text(hex_string):
    """
    Tenta decodificar uma string hexadecimal usando diferentes codificações
    Retorna um dicionário com os resultados de cada codificação
    """
    # Remove espaços em branco e quebras de linha
    hex_string = hex_string.strip()

    # Converte hex para bytes
    try:
        byte_data = bytes.fromhex(hex_string)
    except ValueError as e:
        return f"Erro ao converter hex: {e}"

    # Lista de codificações para tentar
    encodings = ['utf-8', 'latin1', 'cp1252', 'iso-8859-1']

    results = {}

    # Tenta cada codificação
    for encoding in encodings:
        try:
            decoded_text = byte_data.decode(encoding)
            results[encoding] = decoded_text
        except UnicodeDecodeError:
            results[encoding] = f"Não foi possível decodificar usando {encoding}"

    return results

def main():
    # Lê o arquivo hex
    with open('arquivo-strong-9.in-full.hex', 'r') as f:
        hex_content = f.read()

    # Decodifica o conteúdo
    results = decode_hex_to_text(hex_content)

    # Imprime os resultados
    print("Resultados da decodificação:\n")
    for encoding, text in results.items():
        print(f"\n=== Decodificação usando {encoding} ===\n")
        print(text)
        print("\n" + "="*50 + "\n")


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

Está implementação é particularmente eficiente porque, apenas converte o arquivo hex para UTF-8, sem necessidade de testes de chave para decriptar o arquivo, uma vez que ele está apenas convertido.

Resultado do arquivo decriptado:

- Chave = Não possui chave.
- Código Secreto do arquivo = zESbhm.
- Tempo máximo de decriptação = 0.1 milissegundos.
- Quantidade de chaves testadas até encontrar a chave correta: não houve tentativas de teste de chaves, uma vez que o arquivo não estava encriptado.

- Média de chaves testadas por segundo: Não houve testes de chaves, uma vez que o arquivo não estava encriptado.