PROCESSO DE DECRIPTAÇÃO DO ARQUIVO “arquivo-weak-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 "SecurityAES".
- O texto decifrado é legível em ASCII.
- O algoritmo usado é o AES no modo ECB.

2ª Estratégia de força bruta otimizada:

- Em vez de testar todas as combinações possíveis de 16 bytes, o código:
  - Usa o prefixo conhecido "SecurityAES".
  - Precisa descobrir apenas os caracteres restantes.
  - Divide o espaço de busca em grupos menores de caracteres:

```
caracteres1 = "abcdefghi"
caracteres11 = "jklmnopqr"
caracteres12 = "stuvwxyz"
caracteres2 = "ABCDEFGHI"
caracteres21 = "JKLMNOPQR"
caracteres22 = "STUVWXYZ"
caracteres3 = "01234"
caracteres31 = "56789"
```

3ª Paralelização do Processamento:

- Usa a biblioteca "multiprocessing" com POOL.
- Cada processo trabalha com um subconjunto diferente do espaço de busca.
- O código testa diferentes combinações simultaneamente através de "pool.apply_async".

4º Validação Eficiente:

```
def is_texto_legivel(texto):
    try:
        v1 = all(ord(c) < 128 for c in texto.decode('ascii'))
        return v1
    except:
        return False
```

- Verifica rapidamente se o texto decifrado é ASCII legível.
- Funciona como critério de parada quando encontra a chave correta.

5º Processo de Decifração:

```
def tenta_decifrar(texto_cifrado, chave):
    cifra = AES.new(chave, AES.MODE_ECB)
    try:
        decifrado = cifra.decrypt(bytes.fromhex(texto_cifrado))
        return decifrado
    except:
        return None
```

- Usa a biblioteca "Cryptodome" para implementação do AES.
- Modo ECB permite testar cada chave independentemente.

6º Monitoramento e Logging:

- Registra progresso a cada 60 segundos.
- Mantém estatísticas de performance:
    - Número de chaves testas.
    - Tempo decorrido.
    - Taxa de tentativas por segundo.

Está implementação é particularmente eficiente porque:

1º - Reduz drasticamente o espaço de busca usando o prefixo conhecido.

2º - Paraleliza o processamento para aumentar a velocidade.

3º - Tem um critério claro de sucesso (texto ASCII legível).

4º - Divide o problema em partes menores e gerenciáveis.

Resultado do arquivo decriptado:

- Chave = SecurityAES3LUbU.
- Código Secreto do arquivo = nXzDkn.
- Tempo máximo de decriptação = 510.86 segundos (8 minutos e 514 segundos)
- Quantidade de chaves testadas até encontrar a chave correta: 53.324.077.
- Média de chaves testadas por segundo: 835.049,68 (8 threads) chaves/seg.