

# Bibliotecas Essenciais de Python para Cibersegurança

## 1. Redes e Comunicação

- socket: Comunicação de baixo nível usando TCP/UDP.
- scapy: Manipulação e inspeção de pacotes de rede (sniffing, spoofing).
- requests: Requisições HTTP simples e diretas (GET, POST).
- urllib: Manipulação de URLs.
- ftplib/smtplib/telnetlib: Interação com protocolos específicos como FTP, SMTP e Telnet.

Exemplo:

```
import socket  
s = socket.socket()  
s.connect(("example.com", 80))
```

## 2. Análise de Malware e Binários

- hashlib: Geração de hashes (MD5, SHA256).
- pefile: Análise de arquivos PE (executáveis Windows).
- yara-python: Identificação de malwares com regras.
- subprocess/os/shutil: Execução de comandos e manipulação de arquivos.

Exemplo:

```
import hashlib  
print(hashlib.md5(b'teste').hexdigest())
```

## 3. Criptografia

- cryptography: Criptografia de alto nível (Fernet, AES, RSA).
- pycryptodome: Biblioteca criptográfica compatível com PyCrypto.
- base64: Codificação e decodificação.
- secrets: Geração de senhas e tokens seguros.

Exemplo:

```
from cryptography.fernet import Fernet  
key = Fernet.generate_key()  
cipher = Fernet(key)  
token = cipher.encrypt(b"mensagem")  
print(cipher.decrypt(token))
```

## 4. OSINT e Automação

- shodan: Integração com API da Shodan.
- BeautifulSoup4: Extração de dados de HTML (web scraping).
- selenium: Automação de navegador.

## Bibliotecas Essenciais de Python para Cibersegurança

- whois: Consulta dados de domínios.

Exemplo:

```
import whois
print(whois.whois("google.com"))
```

### 5. Pentest e Exploits

- nmap: Wrapper em Python para o Nmap.
- pwntools: Ferramentas para CTFs e exploração de vulnerabilidades.
- impacket: Implementação de protocolos de rede (SMB, RDP, etc).

Exemplo:

```
import nmap
scanner = nmap.PortScanner()
scanner.scan('192.168.1.1', '22-80', '-sV')
```

### 6. Suporte e Utilidades

- logging: Geração de logs estruturados.
- argparse: Criação de interfaces CLI.
- threading/asyncio: Execução concorrente.
- re: Expressões regulares.

Exemplo:

```
import re
match = re.search(r"senha: (\w+)", "senha: abc123")
print(match.group(1))
```