👻 SCARLET GHOST: Manual de Operações de Campo

> ⚠ **AVISO LEGAL:** Esta ferramenta e documentação foram desenvolvidas para fins educacionais, testes em ambientes autorizados (CTFs, Bug Bounty, Red Teaming contratado) e pesquisa acadêmica. O uso não autorizado contra alvos sem consentimento é ilegal.

🎯 Alvos Permitidos (Ambientes de Teste)

Utilize estes domínios para validar a instalação e funcionamento das ferramentas sem riscos legais:

Domínio	Descrição
testphp.vulnweb.com	Aplicação PHP vulnerável (Acuart)
testfire.net	Banco fictício (Altoro Mutual)
zero.webappsecurity.com	App bancário para testes de API
juice-shop.herokuapp.com	OWASP Juice Shop (Loja moderna vulnerável)
exemplo.com	Domínio reservado para documentação (RFC 2606)

🛠️ 1. ProjectDiscovery Suite (Recon & Vulnerability)

Ferramentas modernas escritas em Go, focadas em velocidade e automação.

Nuclei

> **Função:** Scanner de vulnerabilidades baseado em templates YAML. É o "canivete suíço" para encontrar falhas conhecidas.

Bash

```
# Scan básico de vulnerabilidades conhecidas
nuclei -u https://testphp.vulnweb.com

# Scan focado em tecnologias específicas (ex: sites em PHP e Apache)
nuclei -u https://testphp.vulnweb.com -tags php,apache

# Scan apenas para falhas Críticas e Altas (menos ruído)
nuclei -u https://testphp.vulnweb.com -severity critical,high

# Scan de Misconfigurations (Erros de configuração)
nuclei -u https://testphp.vulnweb.com -tags misconfig
```

Subfinder

> **Função:** Descoberta passiva de subdomínios (usa fontes públicas como Shodan, Censys, etc., sem tocar no alvo diretamente).

Bash

```
# Descobrir subdomínios (modo padrão)
subfinder -d exemplo.com

# Modo silencioso (apenas URLs limpas, ótimo para pipes)
subfinder -d exemplo.com -silent

# Salvar em arquivo para uso posterior
subfinder -d exemplo.com -o subdominios.txt
```

Httpx

> **Função:** Verifica quais domínios/subdomínios estão vivos (têm servidor web rodando) e coleta dados.

Bash

```
# Verificar quais subdomínios da lista estão ativos
cat subdominios.txt | httpx

# Coletar Título da página, Status Code e Tecnologia (Fingerprinting)
```

```
cat subdominios.txt | httpx -title -status-code -tech-detect

# Tirar prints (screenshots) das páginas ativas
cat subdominios.txt | httpx -screenshot
```

Naabu

> **Função:** Port Scanner rápido e confiável (focado em portas web).

```
Bash


# Escanear as 100 portas mais comuns (Top 100)
naabu -host testphp.vulnweb.com

# Escanear todas as portas (Full Scan)
naabu -p - -host testphp.vulnweb.com

# Encadeamento: Subdomínios -> Portas -> Http Ativo
subfinder -d exemplo.com | naabu | httpx
```

🔧 2. TomNomNom & Asset Discovery

Ferramentas focadas na filosofia "Unix": fazem uma coisa muito bem feita e permitem encadeamento (pipes).

Waybackurls

> **Função:** Busca no "passado" da internet (Wayback Machine) por URLs antigas, arquivos esquecidos e parâmetros.

```
Bash


# Buscar todo o histórico de URLs do alvo
waybackurls testphp.vulnweb.com

# Filtrar apenas arquivos PHP (interessante para SQLi/XSS)
waybackurls testphp.vulnweb.com | grep "\.php"
```

Httprobe

> **Função:** Pega uma lista de domínios e pergunta" se eles respondem por HTTP ou HTTPS.

Bash

```
# Testar lista crua de subdomínios
cat subdominios_brutos.txt | httprobe

# Preferir HTTPS
cat subdominios_brutos.txt | httprobe -p https
```

Anew

> **Função:** Gerenciador de duplicatas. Adiciona apenas linhas novas a um arquivo existente. Essencial para monitoramento contínuo.

Bash

```
# Adicionar novas descobertas sem duplicar o que você já tem
cat novas_urls.txt | anew urls_mestre.txt
```

GF (Grep Fuzzing)

> **Função:** Wrapper para o grep que usa padrões pré-definidos para encontrar vulnerabilidades (ex: padrões de URL que parecem XSS ou SQLi).

Bash

```
# Buscar URLs com potencial para XSS
cat urls.txt | gf xss

# Buscar URLs com potencial para SQL Injection
cat urls.txt | gf sqli

# Buscar URLs que podem conter redirecionamentos (SSRF/Open Redirect)
cat urls.txt | gf redirect
```

🕷️ 3. Scanning & Crawling Avançado

Nmap

> **Função:** O padrão da indústria para mapeamento de rede. Identifica portas, serviços e versões.

Bash

```
# Scan de Serviços e Versões (Detecta o que está rodando)
nmap -sV testphp.vulnweb.com

# Scan Agressivo (OS Detection + Scripts + Traceroute)
nmap -A testphp.vulnweb.com

# Scan de Vulnerabilidades (Nmap Scripting Engine)
nmap --script vuln testphp.vulnweb.com
```

Katana

> **Função:** Crawler web de nova geração. Navega pelo site (inclusive SPAs em JavaScript) para encontrar todos os links e endpoints.

Bash

```
# Crawling padrão
katana -u https://testphp.vulnweb.com

# Crawling com Headless Browser (simula um usuário real, pega links gerados via JS)
katana -u https://testphp.vulnweb.com -headless

# Salvar todos os endpoints encontrados
katana -u https://testphp.vulnweb.com -o endpoints.txt
```

🚀 4. Fuzzing & Exploração Web

FFUF (Fuzz Faster U Fool)

> **Função:** Fuzzer web ultra-rápido. Usado para descobrir diretórios ocultos, arquivos ou parâmetros testando milhares de palavras por minuto.

Bash

```
# Descoberta de Diretórios (Directory Busting)
ffuf -u http://testphp.vulnweb.com/FUZZ -w wordlist_comum.txt

# Descoberta de Arquivos com extensões específicas
ffuf -u http://testphp.vulnweb.com/FUZZ -w wordlist.txt -e .php,.bak,.zip

# Fuzzing de Parâmetros (descobrir parâmetros ocultos como 'debug', 'admin')
ffuf -u http://testphp.vulnweb.com/page.php?FUZZ=test -w wordlist_params.txt
```

SQLMap

> **Função:** Ferramenta automática para detecção e exploração de falhas de Injeção SQL.

Bash

```
# Verificar se uma URL é vulnerável
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1"

# Tentar listar os bancos de dados (se vulnerável)
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --dbs

# Tentar obter um shell interativo no sistema operacional
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --os-shell
```

Dalfox

> **Função:** Scanner especializado em XSS (Cross-Site Scripting). Analisa parâmetros e verifica reflexões.

Bash

```
# Scan de XSS em uma URL específica
dalfox url "http://testphp.vulnweb.com/search.php?test=query"

# Scan em massa via pipe (Pipeline comum)
cat urls_com_parametros.txt | dalfox pipe
```

⚡ Receitas de Automação (Pipelines)

Combine as ferramentas do **Scarlet Ghost** para criar fluxos de ataque poderosos.

Pipeline 1: Reconhecimento Total

Objetivo: Mapear toda a superfície de ataque de um domínio.

Bash

```
# 1. Encontrar subdomínios
subfinder -d testphp.vulnweb.com -silent | tee subs.txt
```

```
# 2. Verificar quais estão vivos e identificar tecnologias
cat subs.txt | httpx -tech-detect -status-code | tee alive.txt

# 3. Escanear portas nos hosts vivos
cat alive.txt | awk '{print $1}' | naabu -top-ports 1000 | tee ports.txt

# 4. Crawling para achar endpoints ocultos
cat alive.txt | katana -d 2 | tee endpoints.txt
```

Pipeline 2: Caça a Bugs (Bug Bounty Mode)

Objetivo: Encontrar falhas de aplicação rapidamente.

Bash

```
# 1. Coletar URLs históricas e filtrar por parâmetros
waybackurls testphp.vulnweb.com | grep "=" | qsreplace "FUZZ" | tee possiveis_vulns.txt

# 2. Testar XSS em massa
cat possiveis_vulns.txt | dalfox pipe > relatorio_xss.txt

# 3. Testar SQL Injection (modo batch para não perguntar sim/não)
sqlmap -m possiveis_vulns.txt --batch --dbs
```

📂 Gestão de Wordlists (SecLists)

O Scarlet Ghost organiza as wordlists em ~/.scarlet-ghost/wordlists/.

- **Discovery/Web-Content/common.txt**: Ótima para fuzzing inicial de diretórios.
- **Discovery/DNS/**: Para brute-force de subdomínios.
- **Fuzzing/XSS/**: Payloads para testar injeção de scripts.