Uma imagem contendo Logotipo

Descrição gerada automaticamente

Relatório de CTF

U.A. High School – TryHackMe

|  |  |
| --- | --- |
| **Informações do documento** | |
| **Referência** | U.A. High School – Alexandre Gualiume Coruquieri |
| **N° Revisão** | 1 |
| **Data de publicação** | 02/10/2024 |
| **Link** | https://tryhackme.com/r/room/yueiua |

|  |  |  |
| --- | --- | --- |
| **Redação** | Alexandre Gualiume Coruquieri | Estudante |
| **Revisão** | João Branco | Presidente |
| **Aprovação** | João Branco | Diretor |

|  |  |  |
| --- | --- | --- |
| **Histórico de revisões** | | |
| **N°** | **Entregas** | **Descrição** |
| **0** | 16/09/2024 | Produção |
| **1** | 02/10/2024 | Revisão |
| **2** | 02/10/2024 | Aprovação |

**Sumário**

[Contextualização 2](#_Toc177418575)

[Desenvolvimento 2](#_Toc177418576)

[USER.txt: 2](#_Toc177418577)

[ROOT.txt: 9](#_Toc177418578)

[Conclusão 11](#_Toc177418579)

[Referências 11](#_Toc177418580)

Contextualização

Esse desafio apresenta conceitos importantes como *Port Scanning*, *Commmand Injection*, *Reverse Shell,* e *Privilege Escalation.* Temas de suma importância para o pentesting e soluções de CTFs voltado para Web.

Desenvolvimento

USER.txt:

Primeiro, usaremos o **Nmap** com o parâmetro “-A” para enumerar as possíveis portas abertas da máquina alvo, e descobrir o sistema operacional usado e os serviços que estão rodando na porta:

Texto

Descrição gerada automaticamente

Vemos a porta 80 aberta rodando uma aplicação Web e a porta 22 rodando o protocolo SSH (conexão remota). Abrindo o site, nos deparamos com uma página simples, onde não aparenta ter nada de interessante:

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

Portanto, vamos usar a ferramenta **Gobuster** para descobrir diretórios nesse site: Texto

Descrição gerada automaticamente

Visto que “index.html” é a página principal, precisamos explorar o diretório /assets. Então, vamos usar o **Gobuster** novamente, mas dessa vez começando por /assets:

Texto

Descrição gerada automaticamente

Descobrimos que há uma página php rodando no site. Porém, ao entrar nessa página, não há nenhum conteúdo. Dado isso, podemos suspeitar de um *command injection* na URL, então vamos usar a ferramenta **Ffuf** para fazer um *fuzzing*, que consiste em gerar tarefas de entradas para descobrir se a resposta da máquina é positiva para a entrada esperada:

Texto

Descrição gerada automaticamente

Nesse caso, o **Ffuf** testou diversos comandos para o comando “whoami” e o único que retornou o protocolo HTTP 200 (solicitação bem-sucedida) foi o “cmd”. Visto isso, podemos fazer uma *Reverse Shell* para conectar à porta SSH na URL da página web e usaremos o **Netcat** para escutar a porta da *Reverse Shell*:

Interface gráfica do usuário, Site

Descrição gerada automaticamente

Irei rodar o **Netcat** no terminal da minha máquina com meu IP no comando Python na URL do site.

Ex. : http://{IP\_THM} /assets/index.php?cmd={*comando reverse shell python*}

Texto

Descrição gerada automaticamente

Dessa forma, conseguimos uma senha que poderá ser usada posteriormente. Enquanto isso, vamos continuar explorando os diretórios a partir do terminal SSH.

Interface gráfica do usuário

Descrição gerada automaticamente

Fazendo o download das imagens usando o **Wget**, obtemos uma imagem que não pode ser aberta, pois os números hexadecimais iniciais não indicam que a imagem é de extensão .jpg:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Então, devemos alterar o hexadecimal da imagem usando o **Hexeditor** para que se adeque ao padrão .jpeg:

Uma imagem contendo Gráfico de mapa de árvore

Descrição gerada automaticamente

Trocaremos os primeiros números por: FF D8 FF E0 00 10 4A 46.

Dessa forma, conseguimos abrir a imagem:

Tela de computador com imagem de personagem de desenho animado

Descrição gerada automaticamente com confiança média

Usaremos o **Esteghide** para extrair possíveis mensagens com o parâmetro “-sf” para especificar o arquivo:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Descobrimos uma credencial usuário:senha para conectar ao protocolo SSH ou trocar de usuário a partir da *Reverse Shell*:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Explorando os diretórios que o usuário deku tem acesso, encontramos a primeira flag user.txt:

Texto

Descrição gerada automaticamente com confiança baixa

ROOT.txt:

Primeiramente, vamos verificar quais as permissões do usuário “deku”:

Texto

Descrição gerada automaticamente

Podemos executar o script “feedback.sh” como root, o que pode ser uma porta de entrada para um escalonamento de privilégios:

Imagem em preto e branco

Descrição gerada automaticamente

O script em questão, tem uma entrada para o usuário digitar um feedback e em seguida o programa executará o que foi escrito com algumas validações de caracteres. Então, podemos dizer para o programa não perguntar nenhuma senha ao entrar no usuário root:

Texto

Descrição gerada automaticamente

Assim, podemos acessar o usuário root e capturar a flag final:

Texto

Descrição gerada automaticamente

Conclusão

Esse CTF, apesar de considerado fácil na plataforma do THM, pode ser desafiador caso o usuário não identifique certos detalhes cruciais para a resolução do CTF, como o *command injection* na URL. Ademais, a dinâmica para resolver o desafio é bem divertida e tranquila.

Referências

* <https://book.hacktricks.xyz/crypto-and-stego/stego-tricks>
* <https://www.revshells.com/>
* <https://en.wikipedia.org/wiki/List_of_file_signatures>