Logotipo

Descrição gerada automaticamente com confiança baixa

Relatório de CTF

Bugged – TryHackMe

|  |  |
| --- | --- |
| **Informações do documento** | |
| **Referência** | Bugged – Pedro Leal Ariosa |
| **N° Revisão** | 1 |
| **Data de publicação** | 02/10/2025 |
| **Link** | https://tryhackme.com/room/bugged |

|  |  |  |
| --- | --- | --- |
| **Redação** | Pedro Leal Ariosa | Estudante |
| **Revisão** | Sérgio Guidi Trovo | Orientador |
| **Aprovação** | Sérgio Guidi Trovo | Diretor |

|  |  |  |
| --- | --- | --- |
| **Histórico de revisões** | | |
| **N°** | **Entregas** | **Descrição** |
| **0** | 30/09/2025 | Produção |
| **1** | 02/10/2025 | Revisão |
| **2** | 02/10/2025 | Aprovação |

|  |  |
| --- | --- |
| **Informações do CTF** | |
| **Nível de Dificuldade** | Fácil |
| **Tipo de acesso** | Gratuito |
| **Conceitos envolvidos** | Mosquitto, Base64 |
| **Plataforma** | Tryhackme |
| **Área** | Red |

**Sumário**

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

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

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

[Conclusão 4](#_heading=h.1t3h5sf)

[Referências 5](#_heading=h.4d34og8)

**Contextualização**

O relatório descreve as atividades realizadas no CTF Bugged (TryHackMe). O estudo buscou praticar reconhecimento, enumeração, exploração e escalonamento de privilégios em um ambiente seguro e intencionalmente vulnerável. A sala foi escolhida por ser criativa e exótica, dificultando seguir o fluxo típico de um pentest — mapeamento de portas, descoberta de diretórios, testes em aplicações web e verificação de privesc — com registro das evidências e dos comandos utilizados.

**Desenvolvimento**

**What is the flag?**

Utilizando o comando “nmap” foi possível fazer o reconhecimento sobre as portas da aplicação, descobrindo a porta aberta 1883 rodando MQTT. MQTT é um protocolo de comunicação, que funciona a partir de dispositivos publicando mensagens em tópicos.

Texto

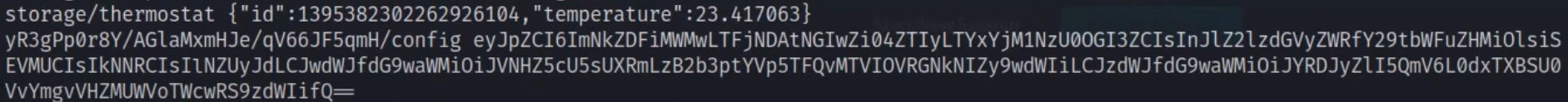
O conteúdo gerado por IA pode estar incorreto.

Para realizar uma conexão, foi utilizado a ferramenta Mosquitto. Desse modo, o comando para realizar a conexão fica: “mosquito\_sub -h IP -t “#” -v”.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

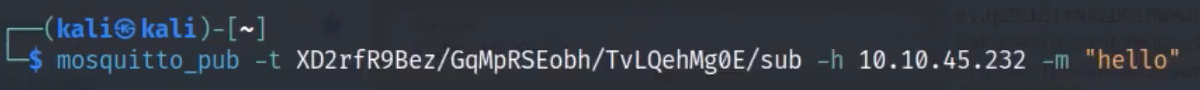
O comando retornou uma resposta criptografada em base64.

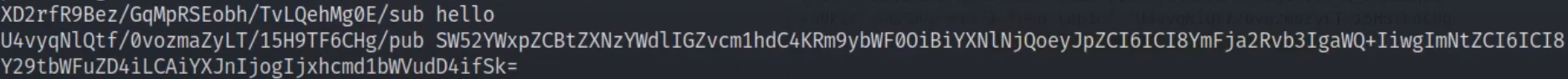


Utilizando um software para descriptografar a mensagem obtém-se a seguinte resposta:Texto

O conteúdo gerado por IA pode estar incorreto.

A partir dessa resposta, é possível observar 2 tópicos abordados, que podem ser úteis. Utilizando o segundo tópico com uma mensagem genérica apenas para ver o retorno do programa, obtemos o formato de código do programa.





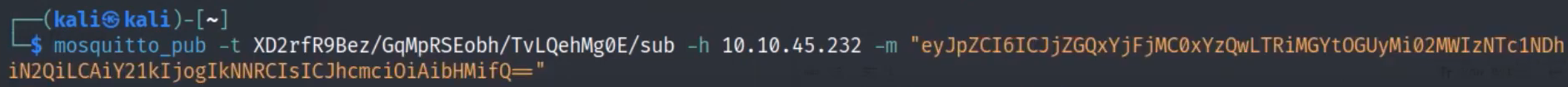
Convertendo a base64 para identificar o formato de código do programa resulta em:



Agora basta substituir o ID encontrado anteriormente, utilizar o comando ls para lista todos os arquivos do programa e transformar para base64.



Após a conversão para base64:

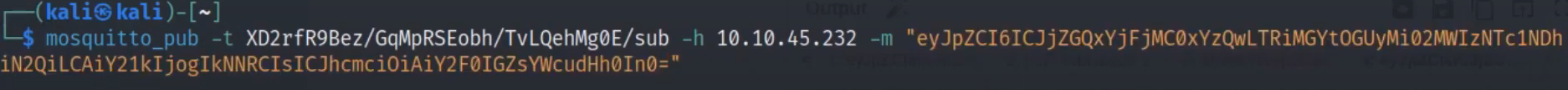


O programa retorna outra mensagem criptografada em base64, que confirma a existência de um arquivo flag.txt.



Após essa descoberta, basta utilizar o comando cat flag.txt que será retornada a flag necessária para resolver o CTF.





Por fim, basta decodificar a resposta obtida que é a flag necessária para a resolução do CTF.

**Conclusão**

Conclui-se que o CTF Bugged proporcionou uma experiência prática e objetiva na aplicação de técnicas de reconhecimento, exploração e escalonamento de privilégios. A sala cumpriu seu papel didático ao reforçar o fluxo básico de um pentest em ambiente controlado, permitindo consolidar conhecimentos e evidenciar vulnerabilidades comuns que podem ser prevenidas com medidas simples de segurança.

**Referências**

HACKTRICKS. *Pentesting Checklist*. Disponível em: https://book.hacktricks.xyz/.