Logotipo

Descrição gerada automaticamente com confiança baixa

Relatório de CTF

PassCode – TryHackMe

|  |  |
| --- | --- |
| **Informações do documento** | |
| **Referência** | PassCode – Matheus da Cunha Castilho |
| **N° Revisão** | 1 |
| **Data de publicação** | 29/03/2025 |
| **Link** | https://tryhackme.com/room/HackfinityBattleEncore |

|  |  |  |
| --- | --- | --- |
| **Redação** | Matheus da Cunha Castilho | Estudante |
| **Revisão** | Nome do revisor | Orientador |
| **Aprovação** | Nome do aprovador | Diretor |

|  |  |  |
| --- | --- | --- |
| **Histórico de revisões** | | |
| **N°** | **Entregas** | **Descrição** |
| **0** | 29/03/2025 | Produção |
| **1** | DD/MM/AAAA | Revisão |
| **2** | DD/MM/AAAA | Aprovação |

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

**Sumário**

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

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

[Obtenção](#_heading=h.3znysh7) do contrato

[Buscando](#_heading=h.2et92p0) a flag

[Conclusão 3](#_heading=h.1t3h5sf)

[Referências 3](#_heading=h.4d34og8)

**Contextualização**

O CTF em questão trata de conceitos simples de blockchain, porém não é um assunto muito explorado em CTFs comuns. Pode ser um pouco desafiador em alguns aspectos, como a decompilação do bytecode em um contrato em Solidity.

**Desenvolvimento**

**Obtenção do contrato**

Em primeiro lugar, precisamos obter o código do contrato, para assim podermos prosseguir com a busca pela flag. Utilizando o comando “cast” do Foundry, é possível obter o bytecode do contrato com base no endereço do contrato.

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

Copiando esse bytecode, é possível obter o código do contrato e suas funções, por meio de um decompiler.

A screenshot of a computer

AI-generated content may be incorrect.

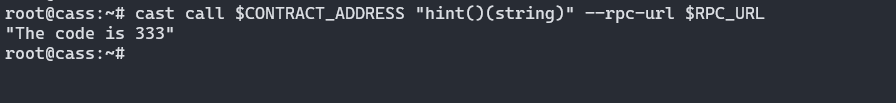
A screenshot of a computer program

AI-generated content may be incorrect.

**Buscando a flag**

Vimos que o contrato possui 3 métodos na qual podemos visualizar, porém em um comentário no código, há uma referência à um método unlock(), que recebe um valor inteiro.

Em um dos métodos, vemos que ele chama uma dica, possivelmente uma string.



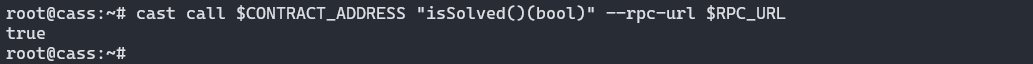
Então, temos que o código é 333. Como o método unlock() necessita de um código, podemos tentar passa-lo aqui.

A screenshot of a computer

AI-generated content may be incorrect.

Nesse caso, também foi preciso passar a private key para conseguir realizar esse envio, junto com a flag –legacy.

Com isso, se executarmos o método isSolved() de novo, vemos que ele retorna que foi resolvido.



Então, ao executar o método getFlag() agora, é possível obter a flag. Anteriormente, o contrato não fornecia a flag até que o método isSolved() retornasse true.



**Conclusão**

Este desafio reforçou a importância de compreender a estrutura de contratos inteligentes e a lógica por trás de suas funções. A experiência adquirida neste CTF pode ser aplicada em cenários mais complexos, contribuindo para o desenvolvimento de habilidades em segurança e engenharia reversa em blockchain.

**Referências**

<https://ethervm.io/decompile>

<https://book.getfoundry.sh>