

ADDETC – Área Departamental de Engenharia Eletrónica e Telecomunicações e de Computadores

MEET – Mestrado Engenharia Eletrónica e Telecomunicações

MEIM - Mestrado Engenharia informática e multimédia

**Cibersegurança**

**Capture the Flag Challenge**

**Turma:**

MEIM-11D, MEET-31D

**Trabalho realizado por:**

Carlos Costa Nº45231

Tiago Martins Nº45240

Miguel Távora Nº45102

Duarte Domingues Nº45140

**Docente:**

Nuno Cruz

**Data:** 22/1/2022

**Índice**

[1. Introdução 1](#_Toc93568983)

[2. Desenvolvimento 2](#_Toc93568984)

[Desafio #1 – Wireshark doo dooo do doo... 2](#_Toc93568985)

[Desafio #2 – Wireshark twoo twooo two twoo... 3](#_Toc93568986)

[Desafio #3 – shark on wire 1 5](#_Toc93568987)

[Desafio #4 – picobrowser 6](#_Toc93568988)

[Desafio #5 – plumbing 7](#_Toc93568989)

[Desafio #6 – Where are the robots 8](#_Toc93568990)

[Desafio #7 - WebNet0 9](#_Toc93568991)

[Desafio #8 - WebNet1 10](#_Toc93568992)

[Desafio #9 - Extensions 12](#_Toc93568993)

[Desafio #10 – So Meta 13](#_Toc93568994)

[3. Conclusões 14](#_Toc93568995)

**Índice ilustrações**

[Figura 1 - Filtragem por respostas do tipo HTTP 200 OK 2](https://iselpt-my.sharepoint.com/personal/a45240_alunos_isel_pt/Documents/Documents/UNIVERSIDADE/MESTRADO/3º%20Semestre/CS/Módulo%204/Trabalho/CS-mod4-trabalho-Grupo-04.docx#_Toc93568996)

[Figura 2 - Desencriptação da flag no CyberChef 3](https://iselpt-my.sharepoint.com/personal/a45240_alunos_isel_pt/Documents/Documents/UNIVERSIDADE/MESTRADO/3º%20Semestre/CS/Módulo%204/Trabalho/CS-mod4-trabalho-Grupo-04.docx#_Toc93568997)

[Figura 3 - Filtro com os pedidos para os subdomínios de reddshrimpandherring 3](https://iselpt-my.sharepoint.com/personal/a45240_alunos_isel_pt/Documents/Documents/UNIVERSIDADE/MESTRADO/3º%20Semestre/CS/Módulo%204/Trabalho/CS-mod4-trabalho-Grupo-04.docx#_Toc93568998)

[Figura 4 - Descodificação através de base 64 no CyberChef 4](https://iselpt-my.sharepoint.com/personal/a45240_alunos_isel_pt/Documents/Documents/UNIVERSIDADE/MESTRADO/3º%20Semestre/CS/Módulo%204/Trabalho/CS-mod4-trabalho-Grupo-04.docx#_Toc93568999)

[Figura 5 - Quantidade de pacotes UDP na captura “capture.pcap” 5](https://iselpt-my.sharepoint.com/personal/a45240_alunos_isel_pt/Documents/Documents/UNIVERSIDADE/MESTRADO/3º%20Semestre/CS/Módulo%204/Trabalho/CS-mod4-trabalho-Grupo-04.docx#_Toc93569000)

[Figura 6 - Conteúdo de um dos pacotes UDP 5](https://iselpt-my.sharepoint.com/personal/a45240_alunos_isel_pt/Documents/Documents/UNIVERSIDADE/MESTRADO/3º%20Semestre/CS/Módulo%204/Trabalho/CS-mod4-trabalho-Grupo-04.docx#_Toc93569001)

[Figura 7 - Flag numa stream UDP 5](https://iselpt-my.sharepoint.com/personal/a45240_alunos_isel_pt/Documents/Documents/UNIVERSIDADE/MESTRADO/3º%20Semestre/CS/Módulo%204/Trabalho/CS-mod4-trabalho-Grupo-04.docx#_Toc93569002)

[Figura 8 - Acesso ao website com o browser errado 6](https://iselpt-my.sharepoint.com/personal/a45240_alunos_isel_pt/Documents/Documents/UNIVERSIDADE/MESTRADO/3º%20Semestre/CS/Módulo%204/Trabalho/CS-mod4-trabalho-Grupo-04.docx#_Toc93569003)

[Figura 9 - Flag escondida no resultado do comando “curl” 6](#_Toc93569004)

[Figura 10 - Output inicial do programa 7](https://iselpt-my.sharepoint.com/personal/a45240_alunos_isel_pt/Documents/Documents/UNIVERSIDADE/MESTRADO/3º%20Semestre/CS/Módulo%204/Trabalho/CS-mod4-trabalho-Grupo-04.docx#_Toc93569005)

[Figura 11 - Output do programa após filtragem 7](https://iselpt-my.sharepoint.com/personal/a45240_alunos_isel_pt/Documents/Documents/UNIVERSIDADE/MESTRADO/3º%20Semestre/CS/Módulo%204/Trabalho/CS-mod4-trabalho-Grupo-04.docx#_Toc93569006)

[Figura 12 – Disallow obtido a partir de robots.txt 8](#_Toc93569007)

[Figura 13 - Flag picoCTF obtida 8](#_Toc93569008)

[Figura 14 - Adição da RSA key ao wireshark 9](https://iselpt-my.sharepoint.com/personal/a45240_alunos_isel_pt/Documents/Documents/UNIVERSIDADE/MESTRADO/3º%20Semestre/CS/Módulo%204/Trabalho/CS-mod4-trabalho-Grupo-04.docx#_Toc93569009)

[Figura 15 - Obtenção do conteúdo desencriptado 9](https://iselpt-my.sharepoint.com/personal/a45240_alunos_isel_pt/Documents/Documents/UNIVERSIDADE/MESTRADO/3º%20Semestre/CS/Módulo%204/Trabalho/CS-mod4-trabalho-Grupo-04.docx#_Toc93569010)

[Figura 16 - Obtenção da flag 9](https://iselpt-my.sharepoint.com/personal/a45240_alunos_isel_pt/Documents/Documents/UNIVERSIDADE/MESTRADO/3º%20Semestre/CS/Módulo%204/Trabalho/CS-mod4-trabalho-Grupo-04.docx#_Toc93569011)

[Figura 17 - Adição da key RSA ao Wireshark 10](#_Toc93569012)

[Figura 18 - Obtenção do conteúdo desencriptado 10](#_Toc93569013)

[Figura 19 - Flag movida 11](#_Toc93569014)

[Figura 20 - Obtenção da flag 11](#_Toc93569015)

[Figura 21 - Inicio do ficheiro de texto 12](https://iselpt-my.sharepoint.com/personal/a45240_alunos_isel_pt/Documents/Documents/UNIVERSIDADE/MESTRADO/3º%20Semestre/CS/Módulo%204/Trabalho/CS-mod4-trabalho-Grupo-04.docx#_Toc93569016)

[Figura 22 - Flag da resolução do desafio 12](https://iselpt-my.sharepoint.com/personal/a45240_alunos_isel_pt/Documents/Documents/UNIVERSIDADE/MESTRADO/3º%20Semestre/CS/Módulo%204/Trabalho/CS-mod4-trabalho-Grupo-04.docx#_Toc93569017)

[Figura 23 - Imagem pico\_img.png 13](#_Toc93569018)

[Figura 24 - Flag encontrada nas propriedades da imagem 13](#_Toc93569019)

# 1. Introdução

Neste trabalho foram realizados 10 desafios do website *picoCTF* com o intuito de exercitar os conhecimentos adquiridos nas aulas de Ciber Segurança. Destes desafios, 5 estão enquadrados na temática das redes, e os restantes são de outros diversos temas.

No total foram obtidos 1900 pontos na plataforma pela resolução dos 10 desafios.

# 2. Desenvolvimento

Neste capítulo são apresentadas as resoluções dos diferentes desafios, tendo sido resolvidos 5 desafios da área de redes, e outros 5 desafios de outras diversas áreas.

## Desafio #1 – Wireshark doo dooo do doo...

Depois de descarregado o ficheiro *shark1.pcapng* com os pacotes provenientes de uma captura de rede utilizou-se a ferramenta Wireshark para a análise do tráfego forma a se encontrar o código da *flag* correto.

Verificou-se que na sua maioria os pacotes existentes eram de HTTP e TCP.

Começou-se por analisar o *flow* dos pacotes HTTP do tipo *response* com o estado 200 OK. Para isso utilizou-se o seguinte filtro *http.responde.code==200* de acordo com a Figura 1:

Graphical user interface, application, table

Description automatically generated with medium confidence

Figura 1 - Filtragem por respostas do tipo HTTP 200 OK

Numa destas *stream* verificou-se um formato de *flag* muito semelhante ao pretendido mas encriptado:

*Gur synt vf cvpbPGS{c33xno00\_1\_f33\_h\_qrnqorrs}\n*

Utilizou-se a ferramenta CyberChef na tentativa de decifrar a informação. A imagem da Figura 2 mostra o sucesso no processo de desencriptação da *flag* através de força bruta da cifra de César que é uma cifra substituição.

Graphical user interface, text, application

Description automatically generated

Figura 2 - Desencriptação da flag no CyberChef

## Desafio #2 – Wireshark twoo twooo two twoo...

Começou-se por descarregar o ficheiro *shark2.pcapng* com o conteúdo relativo a uma captura de tráfego, onde se observou que a maioria dos pacotes registados estão associados aos protocolos HTTP e DNS.

Através da análise por *scan* da *stream* TCP, verificou-se que nos vários pedidos GET existiam diversas *flags* com o formato correto de acordo com o que se procurava. Não se tentou submeter essas flags porque se considerou que eram apenas uma distração.

De seguida, verificou-se que existiam diversos pedidos DNS para vários subdomínios de reddshrimpandherring.com com o seguinte formato:

*XXXXXXXX.reddshrimpandherring.com*

Text

Description automatically generatedAnalisou-se melhor este tráfego por se considerar suspeito visto que a dica dada era procurar por tráfego deste tipo.

Figura 3 - Filtro com os pedidos para os subdomínios de reddshrimpandherring

Para isso, de forma a filtrar o tráfego utilizou-se o seguinte comando apresentado na Figura 3 através do Wireshark.

Como existiam muitos pedidos para o servidor de DNS da google (8.8.8.8), decidiu-se ocultar esses mesmos pacotes, e após a filtragem obteve-se apenas aqueles para o endereço f18.217.1.57.

O próximo passo passou pela concatenação dos subdomínios tendo-se obtido a seguinte codificação:

*cGljb0NURntkbnNfM3hmMWxfZnR3X2RlYWRiZWVmfQ==*

Finalmente obteve-se a *flag* no formato correto através da ferramenta CyberChef pela descodificação de *string* obtida por base 64 como se pode observar na Figura 4.

Graphical user interface, application

Description automatically generated

Figura 4 - Descodificação através de base 64 no CyberChef

## Desafio #3 – shark on wire 1

Este desafio consiste em encontrar a *flag* numa captura de rede através do Wireshark. Analisando o ficheiro “capture.pcap” observou-se a existência de muitos pacotes UDP, como se pode observar na Figura 5, e quando se analisou o conteúdo de um destes pacotes (no campo “*Data*”) estes apenas continham o que aparentavam ser carateres aleatórios, Figura 6.

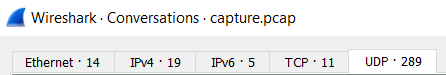


Figura 5 - Quantidade de pacotes UDP na captura “capture.pcap”

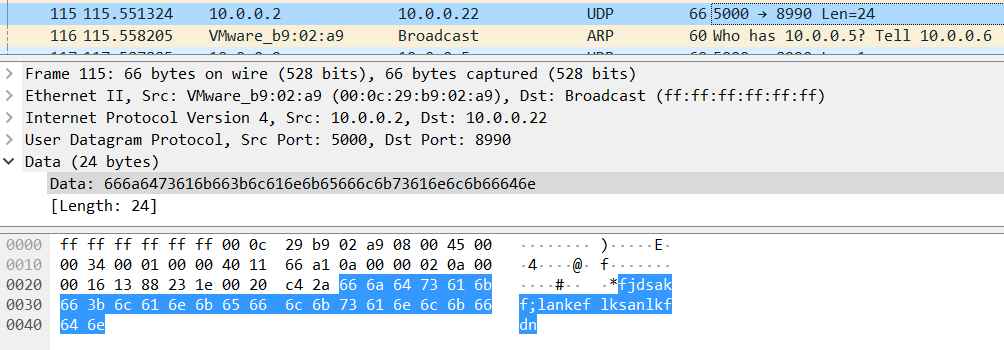


Figura 6 - Conteúdo de um dos pacotes UDP

Através da pista deste desafio, “*What are streams*?”, pensou-se que a *flag* estaria repartida entre diferentes pacotes UDP de uma certa conversa, ou seja, esta poderia ser encontrada analisando várias *streams* UDP. Desta forma foi usada a ferramenta “*Follow UDP Stream*”, com a qual foi possível encontrar a *flag* na *stream* número 6.

Uma imagem com texto

Descrição gerada automaticamente

Figura 7 - Flag numa stream UDP

## Desafio #4 – picobrowser

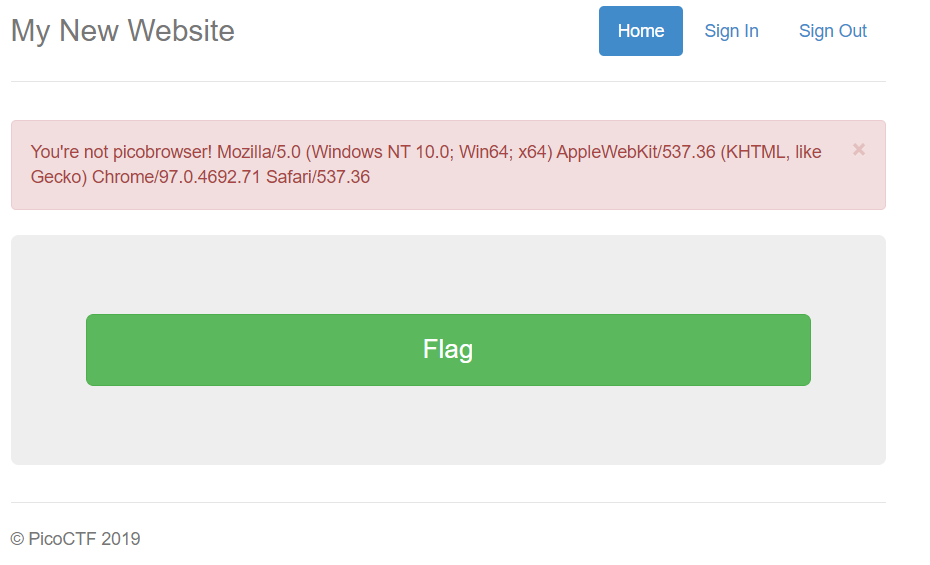
Este desafio consiste em encontrar a *flag* num *website*, no entanto, esta apenas pode ser acedida se o acesso ao mesmo for realizado através do “picobrowser”. Inicialmente, quando se acedeu mao site, apareceu um erro a indicar que não acedemos com o *browser* correto.

Figura 8 - Acesso ao website com o browser errado

Desta forma seria necessário aceder aos conteúdos deste *website* de uma forma alternativa que permitisse “enganá-lo” de forma que este pensasse que estávamos a realizá-lo através do picobrowser. Foi usada a ferramenta “curl” da linha de comandos com recurso à extensão “-A” que é responsável por alterar o *User-Agent* do pedido HTTP.

Assim, foi executado o seguinte comando, “curl -A "picobrowser" <https://jupiter.challenges.picoctf.org/problem/28921/flag>”, e na resposta do mesmo foi possível encontrar a *flag*.

Uma imagem com texto

Descrição gerada automaticamente

Figura 9 - Flag escondida no resultado do comando “curl”

## Desafio #5 – plumbing

Este desafio consiste em encontrar a *flag* no *output* de um do site “jupiter.challenges.picoctf.org”, no porto 4427. Inicialmente foi realizado o acesso através da ferramenta “curl”, no entanto o output rapidamente ficou cheio de mensagens, onde aparentemente não estava presenta a *flag*.

Uma imagem com texto

Descrição gerada automaticamente

Figura 10 - Output inicial do programa

Foi entanto decidido filtrar o resultado deste output para apenas incluir linhas que possuíssem a palavra “picoCTF”. Para tal foi usado o comando “grep “picoCTF””, e este teve de ser concatenado ao comando “curl”, tendo sido usado para tal o comando “|”, também conhecido como *pipe* (referência ao título deste desafio).

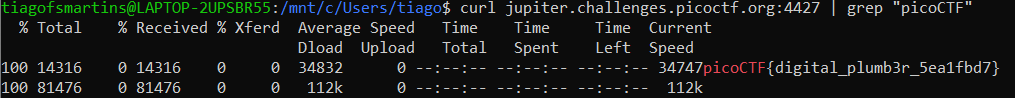
Desta forma, foi possível obter a *flag* deste desafio, uma vez que foi o único resultado presente no *output* do programa.

Figura 11 - Output do programa após filtragem

## Desafio #6 – Where are the robots

Este desafio consiste em encontrar a *flag* no site <https://jupiter.challenges.picoctf.org/problem/60915/>.

Essencialmente para conseguir acesso a informação acedemos ao ficheiro robots.txt, isto é feito por inserir no fim do site robots.txt, como se segue: <https://jupiter.challenges.picoctf.org/problem/60915/robots.txt>

Este ficheiro serve essencialmente para controlar os ficheiros que os rastreadores podem ter acesso. Possui uma ou mais restrições, cada restrição bloqueia ou permite um rastreador de ter acesso ou não a ficheiros de um determinado caminho no website. Caso não seja especificado todos os ficheiros são permitidos para os rastreadores.

Uma imagem com texto

Descrição gerada automaticamente

Figura 12 – Disallow obtido a partir de robots.txt

A partir deste ficheiro foi possível obter o ficheiro html e assim obter informação privilegiada.

<https://jupiter.challenges.picoctf.org/problem/60915/8028f.html>

Uma imagem com texto

Descrição gerada automaticamente

Figura 13 - Flag picoCTF obtida

## Desafio #7 - WebNet0

Este desafio consiste em decifrar a *stream* TLS, nessa mesma *stream* encontra-se a *flag*.

Para isto foi utilizado o Wireshark. Para obter as mensagens decifradas foi adicionado a *key* RSA no Wireshark, como se pode observar na imagem:

Uma imagem com texto

Descrição gerada automaticamente

Figura 14 - Adição da RSA key ao wireshark

Após adicionar a *key*, foi reiniciado o Wireshark e obteve-se mensagens decifradas, representadas a amarelo na figura que se segue:

Uma imagem com texto

Descrição gerada automaticamente

Figura 15 - Obtenção do conteúdo desencriptado

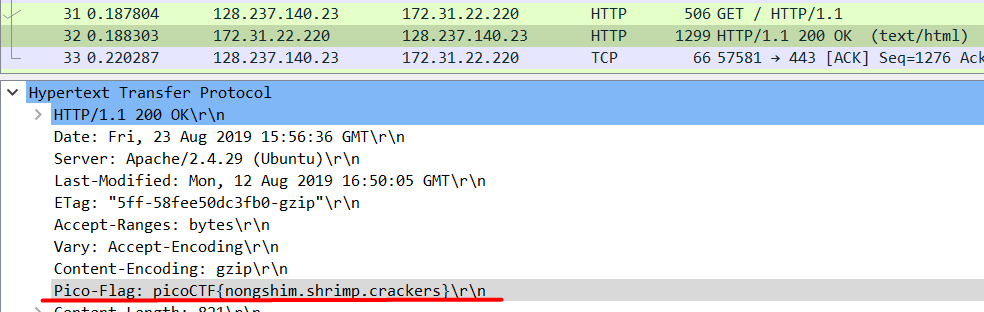
Com a obtenção do conteúdo do HTML foi pesquisado no *header* do HTML e encontrou-se a chave.

Figura 16 - Obtenção da flag

## Desafio #8 - WebNet1

Este desafio é semelhante ao WebNet0, mas desta vez a *flag* não se encontra na *header*.

Da mesma forma que anteriormente, foi utilizado o Wireshark. Começou-se também por obter as mensagens decifradas adicionando a RSA *key* no Wireshark, a partir do ficheiro “picopico.key”.

Uma imagem com texto

Descrição gerada automaticamente

Figura 17 - Adição da key RSA ao Wireshark

Após adicionar a *key*, foi reiniciado o Wireshark e obteve-se mensagens HTTP decifradas, representadas a amarelo na figura que se segue:

Uma imagem com mesa

Descrição gerada automaticamente

Figura 18 - Obtenção do conteúdo desencriptado

Com a obtenção do conteúdo do HTML foi pesquisado no *header* do HTML, como no desafio anterior, mas desta vez encontrou-se que a chave tinha sido movida.

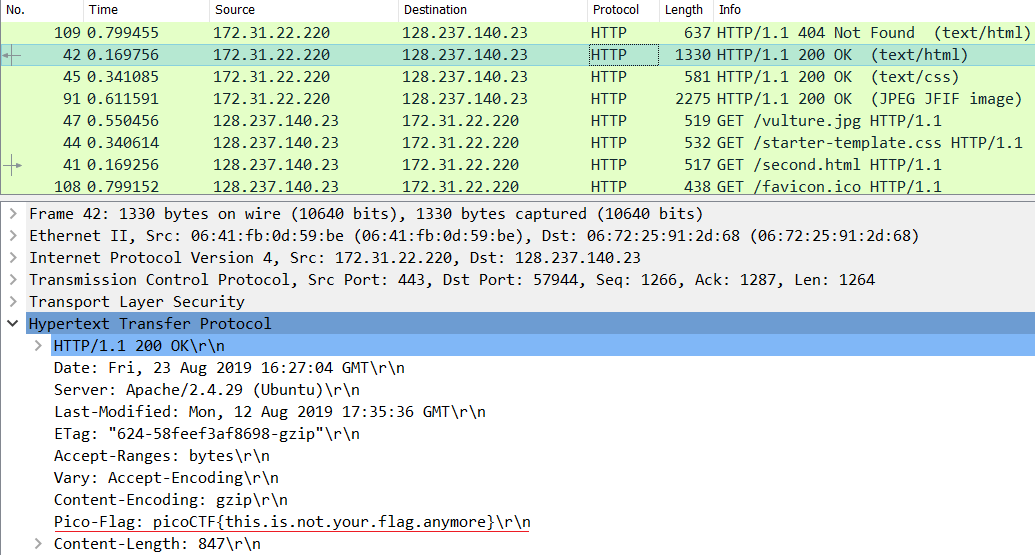


Figura 19 - Flag movida

Decidiu-se então seguir a *stream* HTTP (*Follow* HTTP *Stream*) desta forma é possível observar os dados a circularem entre cliente e servidor. Seguir uma HTTP *stream* permite descodificar informação HTTP, desta forma é possível observar o conteúdo das mensagens HTTP.

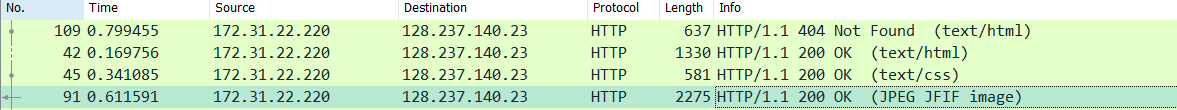




Figura 20 - Obtenção da flag

Realizou-se *Follow HTTP Stream* em um pacote de uma imagem (JPEG JFIF image) e procurou-se por pico, na barra de pesquisa. Encontrou-se finalmente a *flag* (picoCTF{honey.roasted.peanuts}) na secção dos dados da imagem.

## Desafio #9 - Extensions

O objetivo é descobrir a *flag* que se encontra num ficheiro txt. O ficheiro começava por %PNG, como se observa na imagem que se segue.



Figura 21 - Inicio do ficheiro de texto

Uma imagem com texto

Descrição gerada automaticamenteDesta forma surgiu a questão de poder ser um ficheiro PNG, outra pista é o facto de o conteúdo estar escrito em bytes. Desta forma o objetivo era converter o ficheiro para PNG. Foi utilizado o sistema operativo Windows, mas não era possível obter o conteúdo da imagem. Por isso foi utilizado um website de conversão de ficheiro texto para PNG, nomeadamente <https://products.groupdocs.app/conversion/txt-to-png>. O ficheiro obtido foi:

Figura 22 - Flag da resolução do desafio

## Desafio #10 – So Meta

Este desafio consiste em encontrar a *flag* numa imagem png (pico\_img.png), o desafio pode ser encontrado em

<https://play.picoctf.org/practice/challenge/19?page=1&search=so%20meta>.



Figura 23 - Imagem pico\_img.png

Este exercício baseia-se em encontrar uma *flag* em uma imagem png, portanto teve-se a ideia de visualizar a *metadata* da imagem.

Utilizou-se o *magick identify*, que faz parte do *ImageMagick*, um programa que permite obter o formato e características de uma imagem.

Usou-se o comando **identify -verbose**, este comando permite imprimir informação detalhada da imagem.

Uma imagem com texto

Descrição gerada automaticamente

Figura 24 - Flag encontrada nas propriedades da imagem

Nas propriedades da imagem foi possível encontrar a *flag* picoCTF{s0\_m3ta\_eb36bf44} e concluir o desafio.

# 3. Conclusões

Em suma, com este trabalho foi possível consolidar os conhecimentos adquiridos na unidade curricular e aplicá-los na resolução de desafios de diversas temas, maioritariamente na área de redes.