Introducción

Este reto ha sido organizado en bonÀrea durante la HackEPS 2025 donde se planteó un reto práctico relacionado con un PLC Siemens S7-1500. El objetivo consistía en analizar el equipo dentro de la red local, descubrir los servicios disponibles y encontrar la forma correcta de acceder a la información interna configurada para la prueba.

Durante el ejercicio se trabajó con el WebServer del PLC, el servidor OPC UA, comunicación S7 y la lectura de bloques de datos, combinando técnicas de exploración de red con conocimientos básicos de automatización industrial. A partir de estas herramientas fue posible reunir las pistas necesarias, obtener credenciales válidas y acceder a las variables simbólicas que contenían la solución del reto.

Descubriendo el PLC

Para descubrir las IPs de los PLC disponibles durante el reto, una vez dentro de la red wifi proporcionada por la organización y comprobada la IP obtenida en el ordenador, se hace un escaneo para ver que IPs responden al PING dentro del rango 10.72.101.0/24.

```
~ $ for i in {1..254}; do ping -c1 -W1 10.72.101.$i >/dev/null && echo "UP: 10.72.101.$i"; done
UP: 10.72.101.1
UP: 10.72.101.15
UP: 10.72.101.18
UP: 10.72.101.21
UP: 10.72.101.22
UP: 10.72.101.68
UP: 10.72.101.72
UP: 10.72.101.86
UP: 10.72.101.87
UP: 10.72.101.90
UP: 10.72.101.226
```

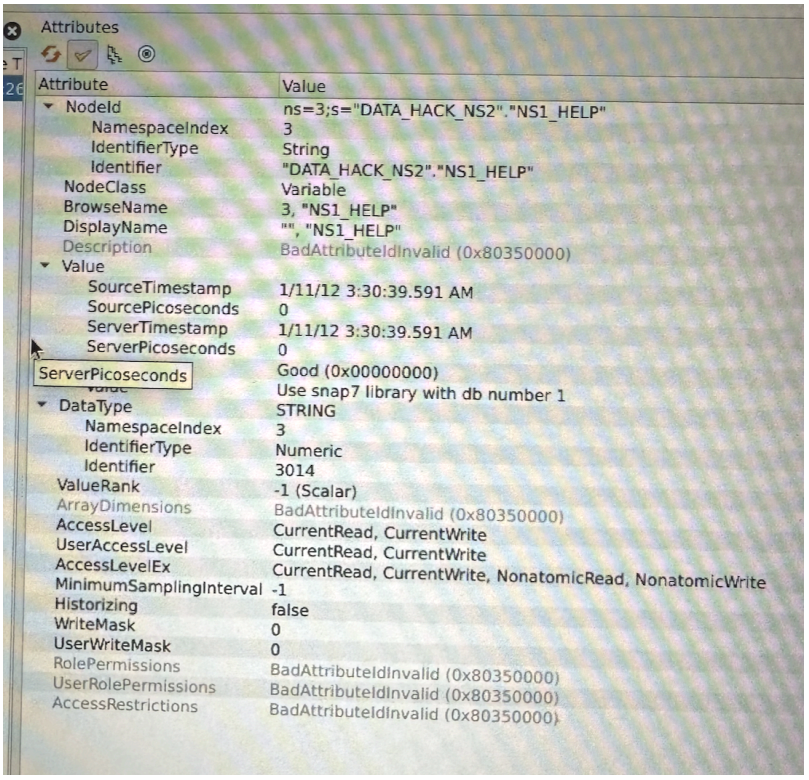
Con el listado de IPs, la organización confirma que el PLC a usar durante la prueba es el PLC_5 con IP 10.72.101.72.

Resolución NS2 (OPC UA)

A partir de la IP del PLC y utilizando el programa de UA Expert se ha podido conectar a un servidor OPC UA sin autenticación. En él se ha encontrado un bloque de datos con 4 variables con AccessLevel CurrentRead y CurrentWrite.

Las variables eran:

- ON_OFF de tipo Bool. Modificando esta variable a True se encendía el check verde de NS1.
- TEMPS_ON de tipo Int16. Modificando este valor con un valor entero se podía cambia el valor de ON
- TEMPS_OFF de tipo Int16. Modificando este valor con un valor entero se podía cambia el valor de OFF
- NS1_HELP de tipo String. En esta variable había una string que daba una pista sobre NS1 dando el nombre de una librería para conectar con protocolo S7 y el número de DB a utilizar.



Resolución NS1 (S7)

Con la pista encontrada durante la resolución de NS2, se crea un script en Python que lee los primeros 256 bytes de información de DB1. La sorpresa fue que al hacer print por pantalla del resultado se puede ver claramente un usuario y una password que se usaran durante la resolución de NS3.

```
RAW DATA: bytearray(b'\x01\x00\x00\x06\x00\t\xfe user:hack and pass:Hackathon2025\x00
LED: True
ON: 10
OFF: 3
```

Lo primero a averiguar era como activar el check para NS1. Para ello se escriben 1s en el primer byte viendo que alguno de los bits era en efecto el que controlaba el check de NS1, así que probando bit a bit se ve que el bit 0 del byte 0 es el correcto.

Para ON y OFF la estrategia ha sido similar, modificando bits poco a poco se iba viendo gracias al HMI como los números iban cambiando. La sorpresa fue que al escribir un INT se estaban modificando las dos variables, por lo que se dedujo que INT era demasiado grande y el tipo correcto sería uno menor como SINT. Así que escribiendo en el byte 3 se podía modificar ON mientra que escribiendo en el byte 5 se podía modificar OFF.

```python
set_bool(data, byte_index: 0, bool_index: 0, value: True)
set_sint(data, byte_index: 3, NS2_TEMPS_ON)  # ON
set_sint(data, byte_index: 5, NS2_TEMPS_OFF)  # OFF

# print("User/Password:", get_string(data, 6))
print("LED:", get_bool(data, byte_index: 0, bool_index: 0))
print("ON:", get_sint(data, byte_index: 3))
print("OFF:", get_sint(data, byte_index: 5))
```

Resolución NS3 (Portal web)

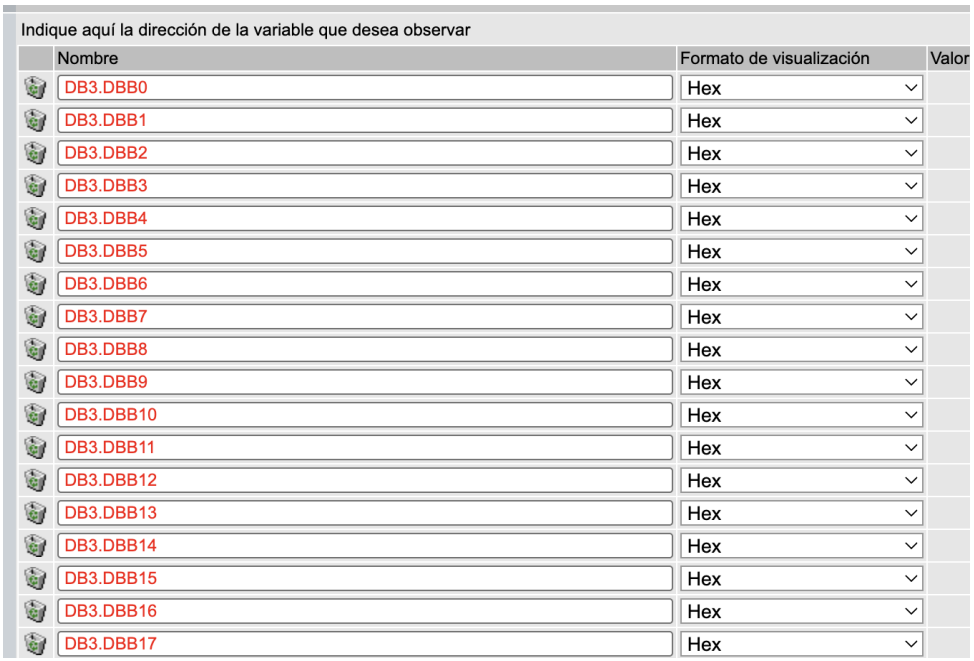
Accediendo a la IP del PLC desde un navegador web, se puede encontrar un portal para gestionar el PLC el cual muestra una información simple del PLC. En ella también hay un apartado para iniciar sesión donde se usan las credenciales descubiertas durante la resolución de NS1.
Las credenciales son:

- User: hack
- Password: Hackathon2025

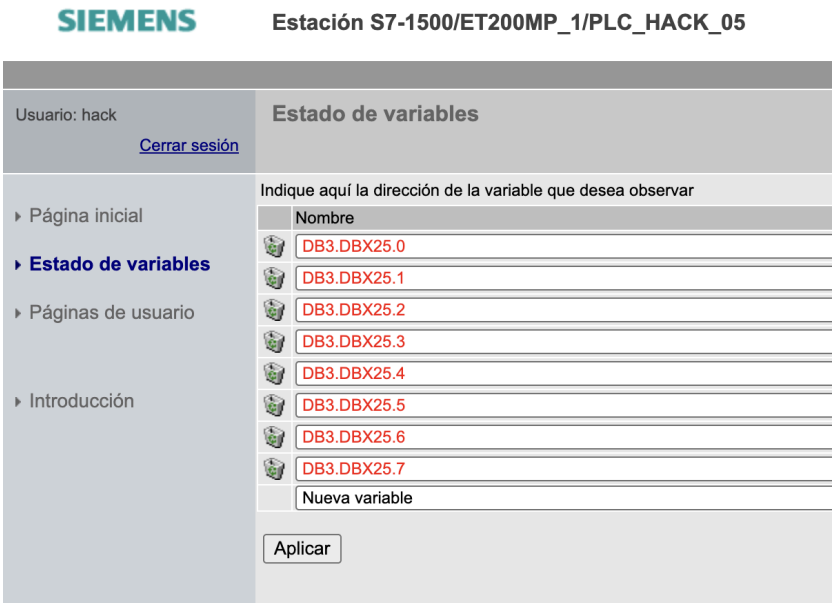
Al hacer login aparecen dos elementos nuevos en el menú, el interesante es “Estado de variables” en el cual se deduce que hay que poner algún nombre para finalmente modificarlo desde esa misma pantalla.

A parte de mirar algún video de Youtube para ver como se suele usar este portal web, se decide hacer una lectura a la documentación oficial donde se explica la sintaxis a usar para cada tipo de dato.

Ya que estábamos intentando descubrir NS3, se interpreta que el DB a utilizar es el 3, aunque no haya ninguna pista que lo indique. Se empieza por una búsqueda exhaustiva para tipo Byte. Todas las opciones acaban en color rojo, indicando que son puntos de memoria no accesibles desde el portal web.



A continuación se intenta bit a bit acabando con el mismo resultado de nombres en rojo.



Durante los diferentes intentos se intentó usar “DATA_HACK_NS3” sin indicar ON_OFF, por lo que no aparecían datos y el nombre seguía en rojo, así que se descartó ir por esa vía.

Después de más de 2 horas, la organización detecta un error en el reto y decide revelar el nombre a usar a todos los grupos que realizan el reto siendo la solución “DATA_HACK_NS3.ON_OFF”.

Resolución secuencia

En esta parte se crea un bucle para probar con fuerza bruta diferentes combinaciones de valores. Ya que no había ninguna pista de qué valores podrían ser, cualquier combinación podría ser la buena.

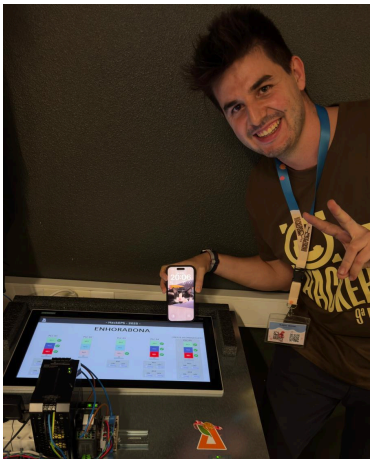
El razonamiento empieza por analizar cómo había sido el reto hasta el momento. Todos los valores eran valores bajos como NS1, NS2 o NS3. No había ningún NS8000 por ejemplo. Por otra parte, durante la resolución de NS2 se usa la DB1, así que seguimos con valores bajos. Los bytes para ON_OFF, TIEMPO_ON y TIEMPO_OFF utilizados también son valores bajos así como en namespace de NS1. Por último, hay que tener en cuenta que es una secuencia de leds que parpadean, por lo que si se quiere apreciar el parpadeo no tendría mucha lógica tener números muy altos ya que el cambio sería muy lento.

Con esa “lógica” en mente, se decide hacer un bucle que vaya modificando los valores de ON y OFF de NS1 y NS2 del 0 al 10 contando de 1 en 1 para no perder ninguna combinación por el camino. Ya que estamos usando comunicaciones, se decide añadir una pequeña pausa de 200ms entre envío y envío para no saturar al PLC.

```python
def main():  1 usage
    for NS1_TEMPS_ON in range(11):
        for NS1_TEMPS_OFF in range(11):
            for NS2_TEMPS_ON in range(11):
                for NS2_TEMPS_OFF in range(11):
                        ns1(NS1_TEMPS_ON, NS1_TEMPS_OFF)
                        ns2(NS2_TEMPS_ON, NS2_TEMPS_OFF)
                        sleep(0.2)
```

Para saber que la secuencia era la correcta era posible hacer la lectura a través de OPC UA de las variables de “Output” y ver que las 3 eran True dando el reto por finalizado.
Una vez encontrada la secuencia correcta se puede ver en la HMI el mensaje de victoria. La secuencia fue encontrada a las 20:06 siendo esta activada con los valores:

- NS1 ON: 10
- NS1 OFF: 3
- NS2 ON: 10
- NS2 OFF: 3



Conclusion

El reto propuesto por bonÀrea resultó ser una experiencia muy entretenida y didáctica. Combinar el análisis de red con tecnologías industriales como el WebServer de Siemens, OPC UA y la lectura de bloques mediante Snap7 permitió explorar de forma práctica cómo se comunican y gestionan los PLC en un entorno real.

La progresión del ejercicio, desde encontrar las primeras pistas hasta acceder al portal con credenciales ocultas y trabajar con variables simbólicas, hizo que el proceso fuera dinámico y motivador.

En resumen, un reto divertido, bien planteado y perfecto para aprender mientras se disfruta del desafío, continuar creando retos tan divertidos como este equipo de bonÀrea!