

Modbus/TCP 协议安全性评估方法的对比与分析

吴 毅, 冯 涛, 王龙鑫, 马 蓉, 石建明

兰州理工大学计算机与通信学院 兰州 中国 730050

摘要 工业控制协议的设计作为确保工业互联网平稳运行的首要条件, 逐渐成为工业控制技术的研究热点。然而工业互联网传输的数据中通常包括系统的运行情况以及设备采集信息, 一旦协议存在安全漏洞, 不但会导致工业隐私数据的泄露, 也会成为黑客入侵工业控制网络的途径, 危及整个工业互联网系统安全。研究人员使用不同方法分析工业控制协议安全性, 其中大部分方法依赖于固定攻击框架或仅基于协议数据包对协议进行安全性分析。然而使用此类方法进行工业控制协议安全性评估时, 评估结果难以全面揭示协议中潜在的攻击路径以及协议实际运行状态。本研究以 Modbus/TCP 协议为研究对象, 采用有色 Petri 网 (CPN) 形式化分析方法和 Smod 渗透测试方法对协议安全性进行系统性评估。研究结果表明, CPN 形式化分析方法能够全面解析协议的状态空间和交互过程, 而 Smod 渗透测试方法受限于其预设的攻击框架和测试用例, 难以实现对 Modbus/TCP 协议运行状态的完整覆盖。此外, 基于 Dolev-Yao 攻击者模型, 本研究利用状态空间分析工具对协议进行深入分析, 成功识别出 Modbus/TCP 协议中存在的三种新型中间人攻击漏洞。与现有评估方法相比, CPN 形式化分析方法不仅能够通过协议运行过程检测通信数据包的异常, 还能揭示多种潜在的攻击路径, 为工业控制协议的安全性评估提供了更为系统和全面的分析框架。

关键词 工业互联网; 工业控制协议; Modbus/TCP 协议; 有色 Petri 网形式化分析方法; Smod 渗透测试分析方法

中图法分类号 TP393.0 DOI 号 10.19363/J.cnki.cn10-1380/tn.2026.05.13

Comparison and Analysis of Modbus/TCP Protocol Security Evaluation Methods

WU Yi, FENG Tao, WANG Longxin, MA Rong, SHI Jianming

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China

Abstract The design of industrial control protocols, as a primary condition for ensuring the stable operation of the industrial internet, has gradually become a research hotspot in industrial control technology. However, the data transmitted in the industrial internet often includes system operation status and equipment-collected information. Once security vulnerabilities exist in the protocol, it may not only lead to the leakage of industrial private data but also serve as a way for hackers to invade industrial control networks, jeopardizing the security of the entire industrial internet system. Researchers have employed various methods to analyze the security of industrial control protocols, most of which rely on fixed attack frameworks or are solely based on protocol packet analysis. However, when using such methods to evaluate the security of industrial control protocols, the results often fail to comprehensively reveal potential attack paths or the actual operational state of the protocol. This study takes the Modbus/TCP protocol as the research object and systematically evaluates its security using the Colored Petri Net (CPN) formal analysis method and the Smod penetration testing method. The results demonstrate that the CPN formal analysis method can comprehensively analyze the state space and interaction processes of the protocol, while the Smod penetration testing method, constrained by its predefined attack framework and test cases, struggles to achieve complete coverage of the operational state of Modbus/TCP protocol. Furthermore, based on the Dolev-Yao attacker model, this study utilizes state space analysis tools to conduct an in-depth analysis of the protocol, successfully identifying three new types of man-in-the-middle attack vulnerabilities in the Modbus/TCP protocol. Compared to existing evaluation methods, the CPN formal analysis method not only detects anomalies in communication packets through the operational process of protocol but also uncovers multiple potential adversarial attack paths, providing a more systematic and comprehensive analytical framework for the security assessment of industrial control protocols.

Key words industrial internet; industrial control protocol; Modbus/TCP protocol; Colored Petri Net formal analysis method; Smod penetration testing method

通讯作者: 冯涛, 博士, 研究员, Email: fengt@lut.edu.cn.

本课题得到国家自然科学基金项目(No. 61762060, No. 62162039); 甘肃省科技厅重点研发项目(No. 23YFGA0060); 甘肃省优秀博士项目(No. 23JRR837)资助。

收稿日期: 2024-09-20; 修改日期: 2025-03-04; 定稿日期: 2026-04-09

1 引言

工业互联网作为现代工业系统的关键组成部分,正逐渐改变传统工业控制系统的生产和管理方式,为工业 4.0 的实现提供了强大的技术支持^[1-2]。通过连接各类工业设备与传感器,工业互联网实现了数据的实时采集、传输和处理,显著提升了工业生产效率和管理水平^[3-4]。然而,随着工业互联网的快速发展,其伴随的安全问题也日益成为限制工业互联网发展的主要挑战之一^[5]。工业互联网安全涉及数据的保密性、完整性和可用性,任何安全漏洞都可能导致严重的经济损失甚至安全事故^[6]。例如,乌克兰的电力系统在 2015 年遭受黑客攻击,导致数十万用户停电,这是首次确认的针对电力系统的网络攻击事件。2023 年 7 月,全球最大芯片制造商台积电遭到 LockBit 勒索软件团伙攻击,被索要 7000 万美元赎金。2023 年 11 月,俄罗斯黑客组织沙虫(Sandworm)针对乌克兰一处变电站发起了新型协同攻击。这些工控安全事件发生的原因都是攻击者通过互联网直接对工业控制协议或工控设备进行恶意代码注入,不仅会泄露工业秘密信息,还可能威胁到生产系统的稳定运行,甚至对工控环境造成严重的物理损害^[7-9]。

针对工业控制协议的安全性分析,研究人员和学者提出了多种协议安全性分析方法并搭建各类协议安全性测试平台^[10],其主要采用逆向分析、渗透测试、模糊测试和异常检测四种技术手段。上述四种方法存在的主要问题如下:

1) 逆向分析技术^[11]通常使用反汇编等方式解析协议允许程序的源代码,旨在监控和分析协议执行过程中的潜在漏洞。然而这一方法要求分析人员不仅具备专业的代码评估能力和反汇编调试技能,还需熟知工业控制系统运行过程。然而,考虑到工业控制系统的封闭性和隔离性,研究人员往往无法在工厂之外的环境中接触到系统组件,从而限制了逆向分析技术的实施。

2) 渗透测试技术^[12]通过模拟已知攻击向量对目标系统进行渗透,旨在评估系统的安全性与稳定性。该方法从敌手角度出发,识别和验证系统潜在的安全漏洞。然而,这一方法的局限性在于无法全面覆盖所有可能的安全威胁,从而无法揭示所有导致系统异常的潜在因素。

3) 模糊测试^[13]和异常检测^[14]通常应用机器学习技术,旨在识别数据中的不符合预期模式的行为。这种技术主要通过分析通用网络数据,虽然能够识

别出存在安全风险的数据包,但是无法指出数据包在真实协议运行环境过程中所遭受的具体攻击手段,更无法指出协议中存在的潜在攻击路径。

有色 Petri 网(Colored Petri Net, CPN)形式化分析方法作为一种基于模型检测^[15]的形式化验证方法,凭借其数学和逻辑技术的支持,以及较高的自动化程度,能够在系统或协议设计的早期阶段发现潜在的弱点,从而保证协议设计的正确性^[16]。具体而言,CPN 形式化分析方法通过构建协议的完整状态空间模型,覆盖所有可能执行路径,能够检测到传统方法难以发现的潜在漏洞。同时,借助状态空间分析工具,CPN 形式化分析方法不仅能识别漏洞,还可精确定位具体攻击路径,为漏洞修复提供明确指导。与现有方法相比,CPN 形式化分析方法能够有效解决:逆向分析方法受限于工业控制系统的封闭性,难以全面获取协议实现的细节;渗透测试方法依赖已知攻击向量,无法充分覆盖潜在的安全威胁;而模糊测试和异常检测方法虽然能够识别异常行为,却无法揭示具体的攻击路径或协议设计中的缺陷。相比之下,CPN 方法能够通过精确的建模和分析,全面识别协议中的潜在安全漏洞。

本研究主要工作如下:

1) 根据 Modbus/TCP 协议消息流抽象出协议参与者间的具体传输流程和消息内容,采用有色 Petri 网形式化分析方法和 Dolev-Yao 攻击者模型对 Modbus/TCP 协议建模并对模型进行安全性分析。使用状态空间分析工具计算协议模型中状态空间节点、有向弧、强联通节点、强联通弧和死节点数目,验证模型的正确性。此外,通过撰写相关查询函数,进一步分析协议中存在的潜在安全漏洞以及具体攻击路径。

2) 本研究搭建工业互联网安全监测与分析虚实实验平台,该平台中人机界面(Human Machine Interface, HMI)、可编程逻辑控制器(Programmable Logic Controller, PLC)和设备节点间通过 Modbus/TCP 协议进行通信。采用 Smod 渗透测试分析方法对 Modbus/TCP 协议进行安全性评估,通过设备及 PLC 对消息的响应,验证 Smod 库函数攻击能力的有效性。

3) 将 Smod 渗透测试分析结果和其他 Modbus/TCP 协议安全性评估方法结果,与 CPN 形式化分析方法结果进行对比。通过对比可以发现,CPN 形式化分析方法相较于其他协议安全评估分析方法可以识别更多的 Modbus/TCP 协议潜在攻击漏洞,并提供具体的攻击路径。

2 相关工作

2.1 Modbus/TCP 协议

Modbus/TCP 协议^[17]作为工业自动化领域的经典通信协议之一,以其简单、稳定、易于实现的特点,在工业互联网环境中得到了广泛应用。Modbus/TCP 协议基于 TCP/IP 协议,采用客户端/服务器架构,能够实现设备之间的快速、可靠通信,适用于各种规模的网络系统,具体 Modbus/TCP 协议数据帧如图 1 所示。其中,事务处理标识可以理解为报文的序列号,通常情况下每次通信后报文序号都要加 1;协议标识通常使用 0000 表示 Modbus/TCP 协议;长度表示当前数据包的长度;单元标识符表示设备地址;功能码和数据决定当前 Modbus/TCP 协议数据包的具体功能和具体对象。

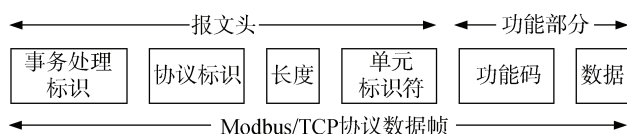


图 1 Modbus/TCP 协议数据帧格式

Figure 1 Modbus/TCP protocol data frame format

然而,在工业互联网环境中,Modbus/TCP 协议面临着安全性和扩展性的挑战。文献[18-26]指出 Modbus/TCP 协议存在以下安全漏洞:①Modbus/TCP 协议在传输过程中不提供端到端的加密机制,使得数据包容易被拦截和篡改;②协议缺乏有效的认证机制来验证通信双方的身份,导致攻击者可能冒充合法设备进行通信;③协议未实施有效措施防止重放攻击,导致攻击者可以重复发送数据包干扰系统操作;④设备节点在接收到数据包后未进行完整性校验,导致通信数据可能被敌手篡改。

2.2 基于机器学习的 Modbus/TCP 协议安全评估方法

为了有效识别和防范 Modbus/TCP 协议中存在的漏洞,近年来不少研究提出了不同的基于机器学习和智能算法的异常检测方法。文献[27]使用支持向量机、决策树、k-近邻和 k 均值聚类四种不同的基于机器学习的异常检测算法分析 Modbus/TCP 协议安全性,根据数据包中协议功能码字段分析协议是否存在命令注入错误和回复注入错误等安全漏洞。文献[28]基于模糊测试方法和遗传算法设计了一种针对工业协议的结构解析算法,通过此算法可识别出 Modbus/TCP 协议中存在的拒绝服务(Denial of Service, DoS)攻击漏洞。文献[29-30]提出一种名为

modlstm 的神经网络架构,用于识别基于 Modbus/TCP 协议中存在的 DoS 攻击。此架构通过对输入数据包进行预处理、特征重新编码和流量分类三步,通过协议数据帧中线圈和寄存器等信息判断系统是否受到 DoS 攻击。文献[31]基于卷积神经网络构建 Modbus/TCP 协议异常报文识别方法。尽管基于机器学习的异常检测方法能有效识别 Modbus/TCP 协议中的潜在安全风险,如命令注入和 DoS 攻击等漏洞,但其无法明确提供具体的攻击路径。

2.3 基于渗透测试的 Modbus/TCP 协议安全评估方法

还有研究人员通过搭建测试平台、实施渗透测试等方法,对 Modbus/TCP 协议的安全漏洞进行系统性分析。文献[32]针对 Modbus 协议中存在重放攻击、错误命令注入和错误访问注入三种安全漏洞,研究人员搭建一个 SCADA(Supervisory Control And Data Acquisition)测试平台,通过使用 Ettercap 和 Wireshark 等软件截获 Modbus/TCP 数据包,将数据包进行篡改后使用搭建平台中帧过滤和校验机制,判断是否能够检测出存在安全隐患的数据包。文献[33]使用 IGSS 系统(Interactive Graphical SCADA)模拟智能电网环境,并使用 Modbus/TCP 协议进行通信。研究人员通过在通信双方部署敌手,利用 ARP 欺骗技术实现对 Modbus/TCP 协议的中间人攻击。最后,通过 Wireshark 捕获数据包后证实 Modbus/TCP 协议易受中间人攻击和 DoS 攻击。文献[34]提出一种基于 Modbus/TCP 协议的入侵检测算法,此算法通过五个不同模块捕获 Modbus/TCP 流量并对流量进行分析,通过分析发现此入侵检测算法可以有效检测到 Modbus/TCP 协议中存在的拒绝服务攻击。文献[35-37]通过搭建 Modbus/TCP 协议仿真环境,使用 Smod 渗透测试方法分析协议安全性。现有的 Modbus/TCP 协议渗透测试分析方法通过构建测试平台、实施 ARP 欺骗和中间人攻击模拟等手段,揭示了该协议中的常见漏洞。然而,这些方法在覆盖所有安全威胁方面存在局限性,特别是在检测复杂攻击时表现不足,且无法全面识别所有潜在风险。

通过对现有研究工作的整理可以发现,学者对 Modbus/TCP 协议的安全性分析主要依赖于基于已知漏洞的代码编写,并构建工业控制平台进行协议安全性评估。尽管此类方法能够有效识别协议中潜在的安全漏洞,但其分析范围通常局限于已知的攻击手段和漏洞类型。在异常检测算法和模糊测试方法的应用中,研究者通过解析协议数据包的特征,建立 Modbus/TCP 协议请求与响应报文特征值之间的映射关系,从而检测数据行为特征是否偏离正常

模型,以判定协议是否遭受攻击。然而,由于工业控制系统数据具有高度异构性,且数据集易受噪声、数据缺失等因素干扰,同时包含大量无关或异常数据,因此无法直接用于模型训练,需对数据集进行预处理。此外,此类算法属于“黑箱”建模,虽然模型具备一

定的准确性,但其对数据安全性判断的解释能力不足。更为关键的是,上述方法仅能基于协议数据包判断攻击行为,而无法揭示协议数据包在通信过程中遭受攻击的具体路径。表 1 详细列举了 Modbus/TCP 协议的具体安全评估方法。

表 1 Modbus/TCP 协议安全评估方法统计

Table 1 Statistical analysis of security evaluation methods for the Modbus/TCP protocol

方案	分析方法	分析结果	不足之处
文献[27]	基于机器学习异常检测算法	数据包被篡改从而导致接收方被欺骗	
文献[28]	模糊测试、遗传算法	DoS 攻击	无法识别具体的攻击路径,通常只能分析出协议遭受了攻击,却未能准确指出攻击的具体类型
文献[29-30]	基于卷积神经网络提出一种 modlstm 神经网络架构,用于识别恶意流量	数据包中存在 DoS 攻击	
文献[31]	基于卷积神经网络的异常报文检测方法	检测出存在异常的报文	
文献[33]	模拟基于 Modbus/TCP 协议的智能电网系统,对数据包篡改后分析协议中存在的安全漏洞	中间人攻击、DoS 攻击	未能全面覆盖所有潜在的攻击手段,且对复杂攻击的检测能力较弱
文献[34]	提出一种入侵检测算法	DoS 攻击	
文献[35-37]	搭建实验平台, Smod 渗透测试方法	协议存在 DoS 攻击漏洞,并且未进行身份认证	

2.4 形式化分析方法

形式化分析方法在协议安全性验证中的应用最早由 Needham 和 Schroeder 提出的,随后由 Dolev 和 Yao 将过程具体化,并于 1983 年发表了重要成果 Dolev-Yao 攻击者模型^[38]。近年来,相关研究工作者开发了多种基于模型检测的自动协议形式化分析工具,如 Scyther^[39]、SPIN^[40]、AVISPA^[41]和 ProVerif^[42]等。然而, Gong 等^[43]指出 ProVerif 作为一种基于逻辑编程的方法,能够确保在协议执行过程中不泄露私有信息,并声称可以计算出多条攻击路径。但其计算的攻击路径通常受到限制,大多数情况下只能包含单一攻击路径。Bojjagani 等^[44]指出 Scyther 是一种在完美密码学假设下能有效识别并揭示协议中存在安全漏洞的形式化分析工具,但是 Scyther 在分析过程中通常依赖于一些假设和抽象,例如理想化的网络模型或加密算法假设,而这些假设在实际情况中可能不完全成立,影响分析结果的准确性。Wang 等^[45]认为 Tamarin Prover 是一种以自动化为代价,但是无法通过特定抽象和解析方法捕获的数据间的关系。CPN 作为一种强大的形式化分析工具,通过图形和颜色的方式来表示系统的状态和变化,不仅可以用于描述系统的行为和逻辑,还能有效地模拟和分析系统中可能出现的各种动态过程。目前已有大量学者针对工业控制协议、车载协议、智能楼宇协议等,使用基于 CPN 形式化分析方法对这些协议进行安全评估,如表 2 所示。

表 2 基于 CPN 形式化分析方法的工业控制协议

Table 2 Formal analysis of industrial control protocols based on CPN

协议	分析结果
DNP3 ^[46]	无法抵御重放、篡改、欺骗三类攻击漏洞
EtherCAT ^[47]	无法抵御重放、篡改、欺骗三类攻击漏洞
Wireless HART ^[48]	无法抵御重放、篡改、欺骗三类攻击漏洞
BACnet ^[49]	无法抵御重放、篡改、欺骗三类攻击漏洞
CAN ^[50]	无法抵御重放、篡改、欺骗三类攻击漏洞
Z-Wave ^[51]	控制器与恶意节点共享 ECDH 密钥
EnOcean ^[52]	无法抗已知密钥泄露伪装攻击

3 背景知识

3.1 CPN 形式化分析方法

有色 Petri 网^[53]是原始 Petri 网的一种扩展,属于高级 Petri 网的范畴。CPN 作为一种高级 Petri 网,提供了对复杂系统建模与验证的功能。与原始 Petri 网相比,CPN 的一个显著特点是引入“颜色”概念为托肯赋予了多样化的含义,极大地增强了 Petri 网在数据表示方面的能力。在 CPN 中,库所被定义为颜色集,托肯被定义为颜色集的多重集,从而允许模型以更灵活的方式表达不同类型的数据和操作,从而支持对系统行为进行多角度的分析和验证。

CPN 融合了传统 Petri 网的图形符号和基本原语,为并发、通信和同步的建模提供了基础,同时引入了标准函数式编程语言(Standard ML, SML)。将 SML 语言融合进 CPN,赋予了 CPN 模型定义数据类型、

描述数据操作以及建立参数化模型的进阶功能。这些功能不仅显著简化了模型的建立过程, 提升了系统分析的效率, 同时也为模型验证提供了强大的原语支持。借助 SML, 研究人员和开发者能够以更加灵活和高效的方式, 构建、分析及优化复杂系统模型, 充分满足不同领域的实际应用需求。通过这种方式, CPN 模型不仅能够准确描述系统的静态结构, 还能有效地模拟和分析系统的动态行为, 极大地增强了模型在复杂系统设计和验证中的应用价值。

3.2 Smod 渗透测试分析方法

Smod^[35-37]是一种专门为 Modbus/TCP 协议开发的渗透测试分析方法, 其基于 Python 开发, 并借鉴了 Metasploit 框架的部分特征, 旨在评估 Modbus/TCP 协议安全性。Smod 渗透测试采用模块化设计, 提供了多种攻击手段, 如重放攻击、DoS 攻击及未经授权访问攻击。其优势在于具备高度的灵活性和可扩展性, 使得模块能够快速开发, 以应对不断演变的安全威胁或特定的测试需求。通过为安全研究人员提供全面的库函数, Smod 渗透测试方法有助于评估和增强基于 Modbus/TCP 的工业控制系统的安全性, 从而促进构建更安全的工业网络环境。

表 3 Smod 渗透测试分析方法库函数

Table 3 Library functions in Smod penetration testing method

编号	攻击函数	攻击类型
1	modbus/dos/arp	DoS 攻击
2	modbus/function/readCoils	未经授权访问
3	modbus/function/readDiscreteInput	未经授权访问
4	modbus/function/readHoldingRegister	未经授权访问
5	modbus/function/readInputRegister	未经授权访问
6	modbus/sniff/arp	DoS 攻击

表 4 颜色集及变量定义

Table 4 Color set and variable definition

颜色集	描述
colset Transaction_Identifier=STRING;	事务处理标识
colset Protocol_Identifier=STRING;	协议标识
colset Length = STRING;	长度
colset Unit_Identifier = STRING;	单元标识符
colset Function_Code=STRING;	功能码
colset MBAP = record TI:Transaction_Identifier*PI:Protocol_Identifier *L:Length*UI:Unit_Identifier;	Modbus/TCP 数据包中报文头部分
colset SData = with RedOpen RedClose GreenOpen GreenClose Red Green;	发送数据
colset RData = with Open Close;	接收数据
colset SPDU = product Function_Code*SData;	发送的 Modbus/TCP 数据包中功能码部分
colset RPDU = product Function_Code*RData;	接收的 Modbus/TCP 数据包中功能码部分
colset MSG1 = product MBAP*SPDU;	数据包 1
colset MSG2 = product MBAP*RPDU;	数据包 2
val switch1 = true;	开启攻击
val switch1 = false;	停止攻击

4 Modbus/TCP 协议安全评估模型

为了使建立的 Modbus/TCP 协议模型更加清晰明了并且易于理解, 本研究采用自顶向下的层次结构来搭建 Modbus/TCP 协议的 CPN 模型, 其中包括 1 个顶层模型和 3 个底层模型, 具体 Modbus/TCP 协议的顶层模型如图 2 所示。在 CPN 模型中, 矩形表示变迁, 椭圆表示库所, 双线矩形表示替代变迁, 表明该变迁有更详细的子模型。此外, 该模型广泛采用了融合库所的设计结构, 其中每个融合库所由库所及其对应的标签(Fusion ID)共同构成。具有相同标签的融合库所在逻辑上被视为同一库所, 且这些库所可以在同一页面或不同页面中多次出现。根据 CPN 模型的变迁规则, 每个变迁通过输入弧和输出弧连接不同的库所, 触发条件依赖于输入位置标记的颜色和数量, 执行后改变系统状态。为了使 CPN 模型完全按照协议原始步骤进行模拟, 其颜色集的定义与图 1 中 Modbus/TCP 协议数据帧定义完全相同, 具体颜色集定义如表 4 所示。主站 Poll 和从站 Slave 通过网络 Net 进行数据交互。Poll 通过 Net 将请求消息发送给 Slave, Slave 处理完请求消息之后, 通过 Net 将响应消息发送给 Poll。在上述模型中使用 Dolev-Yao 攻击者控制整个网络, 替代变迁 Net 是攻击者对 Modbus/TCP 协议模型进行攻击的位置。

替代变迁 Poll 的 CPN 模型如图 3 所示。Poll 是整个模型模拟运行的起点和终点。库所 TI、PI、L、UI、fc 和 data 分别存储了与 Modbus/TCP 协议相关的令牌。当消息发送时, 这些库所中的令牌将被传递至相应的变迁。具体而言, 变迁 CombinationMBAP 接收来自库所 TI、PI、L 和 UI 的令牌, 这些令牌分别代表事务标识符、协议标识符、长度和单元标识

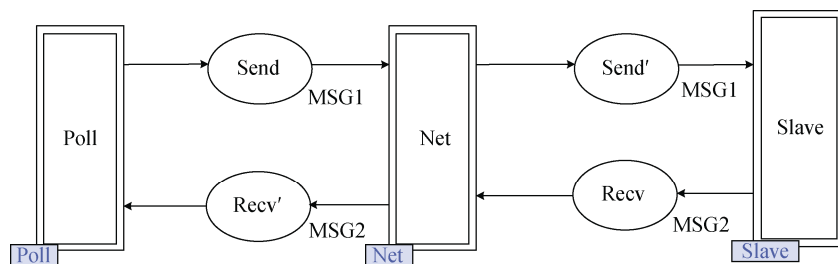


图 2 Modbus/TCP 协议顶层模型

Figure 2 Top-level model of the Modbus/TCP protocol

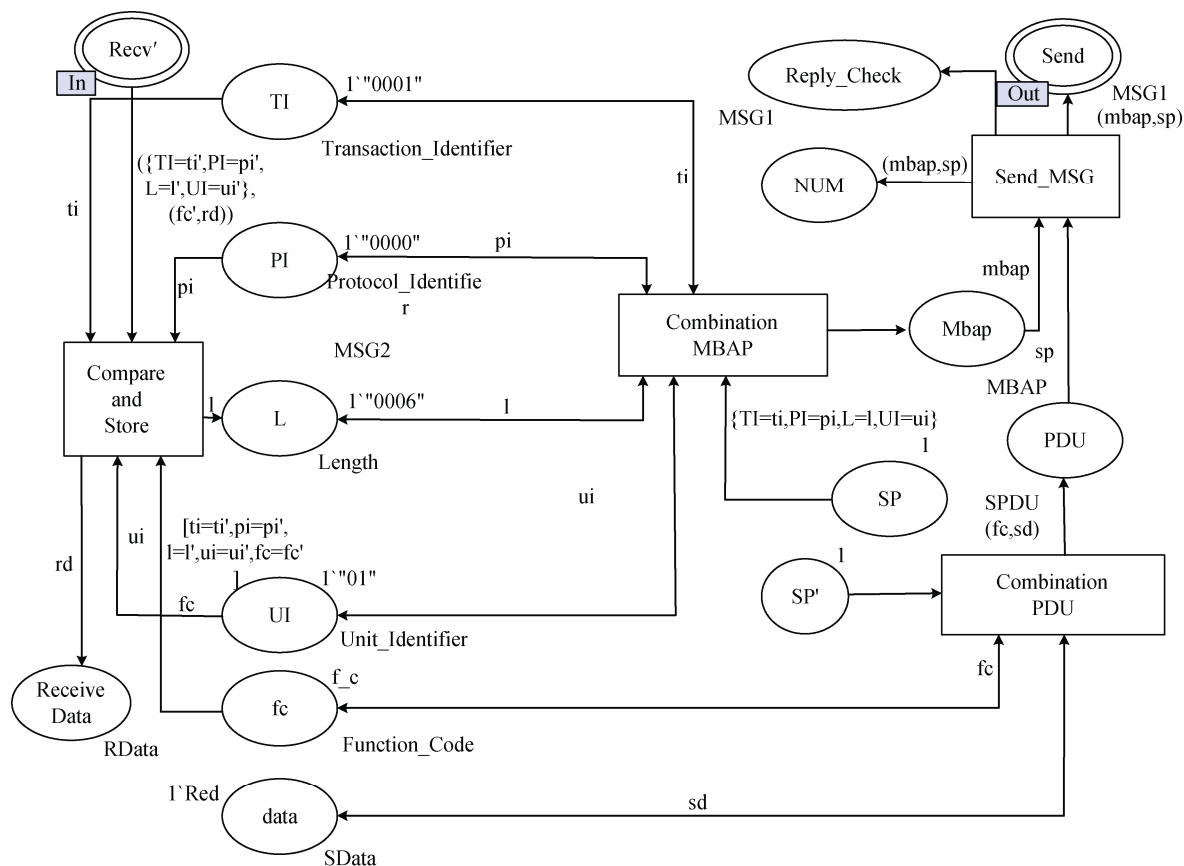


图 3 替代变迁 Poll 内部模型

Figure 3 Internal model of the substitution transition Poll

符。通过处理这些令牌, 变迁 CombinationMBAP 构造出 Modbus/TCP 协议的 MBAP 报文头部分。与此同时, 变迁 CombinationPDU 接收来自库所 fc 和 data 的令牌, 这些令牌分别代表功能码和发送数据。变迁 CombinationPDU 利用这些令牌构造出 Modbus/TCP 协议的 PDU 功能码部分。最终, 变迁 Send_MSG 将 MBAP 和 PDU 部分整合, 构造出完整的 Modbus/TCP 数据包, 并通过库所 Send 将该数据包发送至从站。

替代变迁 Net 的 CPN 模型如图 4 所示。Net 是主站 Poll 和从站 Slave 进行通信的介质。变迁 Net 和 Net' 分别模拟了在建立连接过程中, 主站向从站发起请求消息的传输路径以及从站响应主站发送相

关信息的传输路径。攻击者就是通过该传输路径进行攻击来达到破坏主从站正常通信的目的。

替代变迁 Slave 的 CPN 模型如图 5 所示。从站 Slave 通过库所 Send' 接收主站 Poll 发送的请求消息 MSG1 之后, 通过变迁 ReceiveMSG 将消息 MSG1 拆分为报文头部信息 MBAP 和报文数据信息 PDU, 然后通过变迁 WhetherResponse 判断请求消息 MSG1 是否是 pi 为 0000, ui 为 01 的主站发送的。变迁 ResponseContent 对消息 MSG1 的数据部分进行判断, 功能码 fc 对应操作位的操作类型是否与原数据消息中的操作位的操作类型相同。变迁 SendMSG 整合响应报文的头部信息 MBAP 和数据

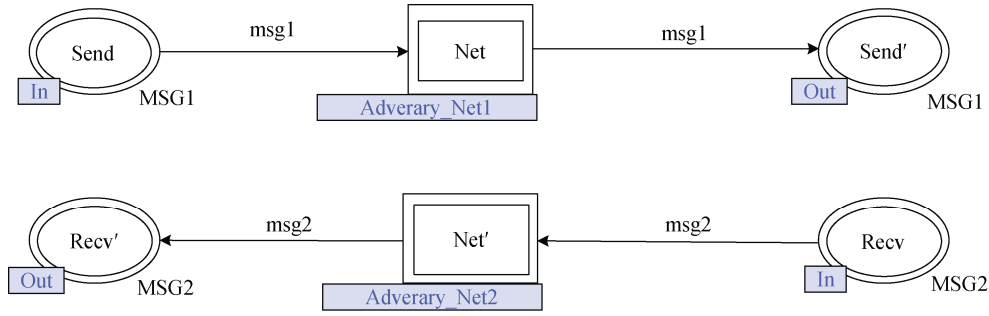


图 4 替代变迁 Net 内部模型

Figure 4 Internal model of the substitution transition Net

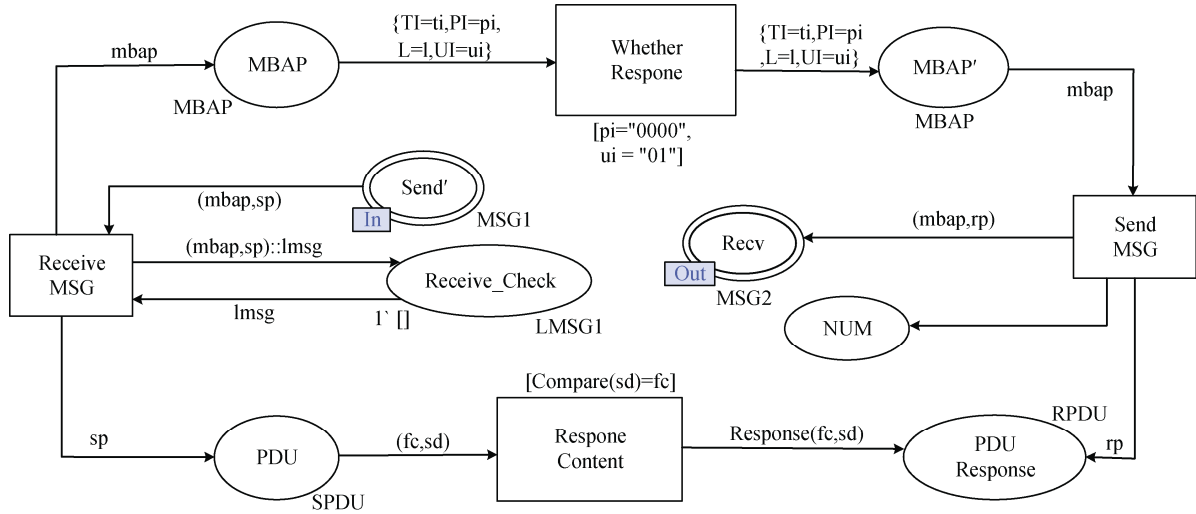


图 5 替代变迁 Slave 内部模型

Figure 5 Internal model of the substitution transition Slave

信息 RPDU, 构造消息 MSG2, 并通过库所 Recv 发送给主站。

基于攻击者的 Modbus/TCP 协议安全验证模型 Adversary_Net1 和 Adversary_Net2 如图 6 和图 7 所示。在该攻击模型中, 配置项 switch1 和 switch2 用于控制攻击行为的触发条件: 若其赋值为 true, 则攻击者对协议通信中的请求消息流实施攻击; 若为 false, 则保持正常通信。当主站 Poll 或从站 Slave 发送的消息被截获后, 变迁 T0 和变迁 GetInformation 被激活, 并对捕获的消息进行解析与拆分, 将分解后的消息存储至攻击者的知识库 OriginalMessage 中。随后, 攻击者从库所 OriginalMessage 中随机选择若干令牌, 通过变迁 T2 对消息进行重组, 并将篡改后的消息发送至目标从站 Slave 或主站 Poll, 从而完成攻击过程。

5 安全评估模型分析

5.1 模型状态空间报告

通过对状态空间表 5 的分析可知, Modbus/TCP 协议模型状态空间节点数与强连通组件的节点数一致,

且有向弧和强连通弧的数量相同。这表明模型在状态变化过程中具备完全的可达性, 且不会导致状态无限扩展或行为的无穷迭代。借助状态空间分析工具对模型运行结果进行查询, 通过图 8 可以发现, 除了 Poll 模型中的 NUM 库所外, 其他所有库所均未包含颜色集, 这表明颜色集已经在模型的整个运行过程中完全消耗完毕。进一步地, 使用变迁查询语句对 CPN 模型运行结束后的变迁状态进行查询, 确认在模型运行结束时, 没有死变迁或活变迁。这意味着所有变迁都被激活, 且不存在任何未激活的变迁。因此, 可以确认原始模型已完成所有可能的状态转换。

通过表 5 可以发现, 引入重放攻击的 Modbus/TCP 协议模型中仅存在的 4 个死变迁是由于重放攻击仅在 Adversary_Net1 模型中开启, Adversary_Net2 模型中未开启, 从而此模型中 Get_Information、Get_Information'、T0 和 T2 变迁并没有被激活, 所以为死变迁。模型中其他变迁均成功激活, 可以说明协议并没有检测到受到攻击, 一直在正常运行。其中, 64 个死节点表示此模型存在 64 种模型最终的停止状态。

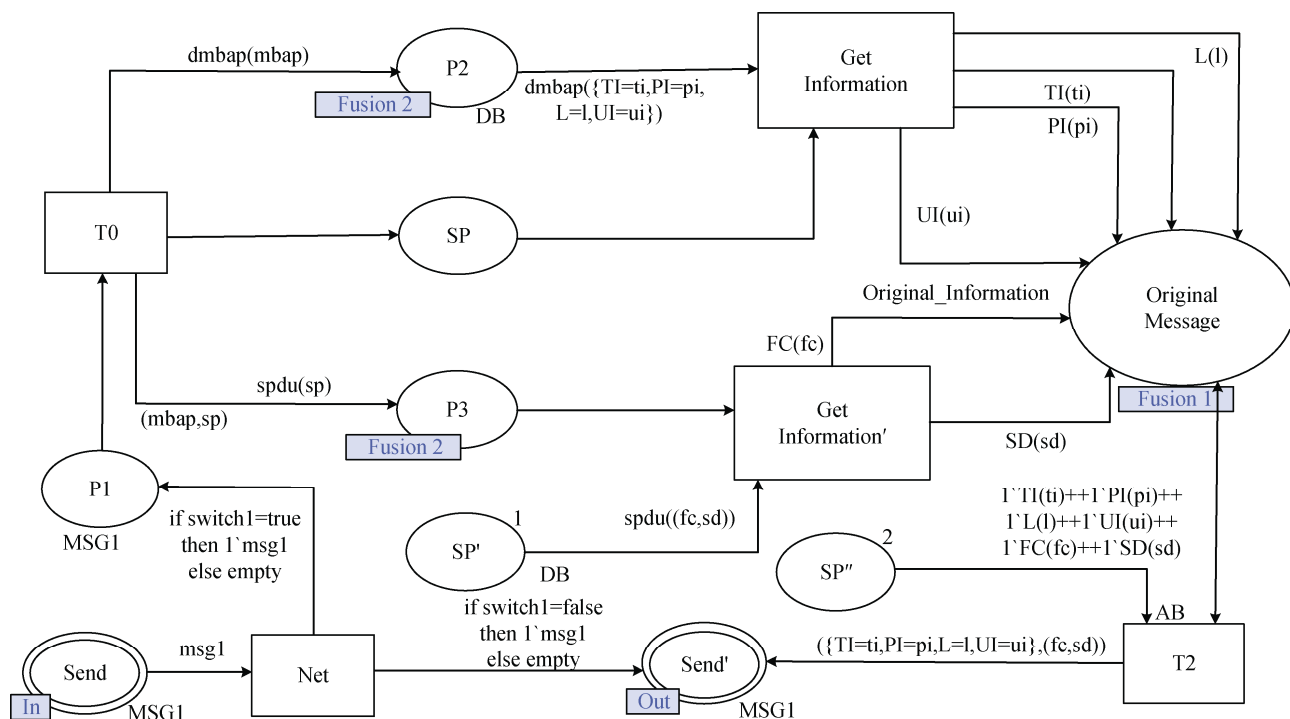


图 6 替代变迁 Adversary_Net1 内部模型

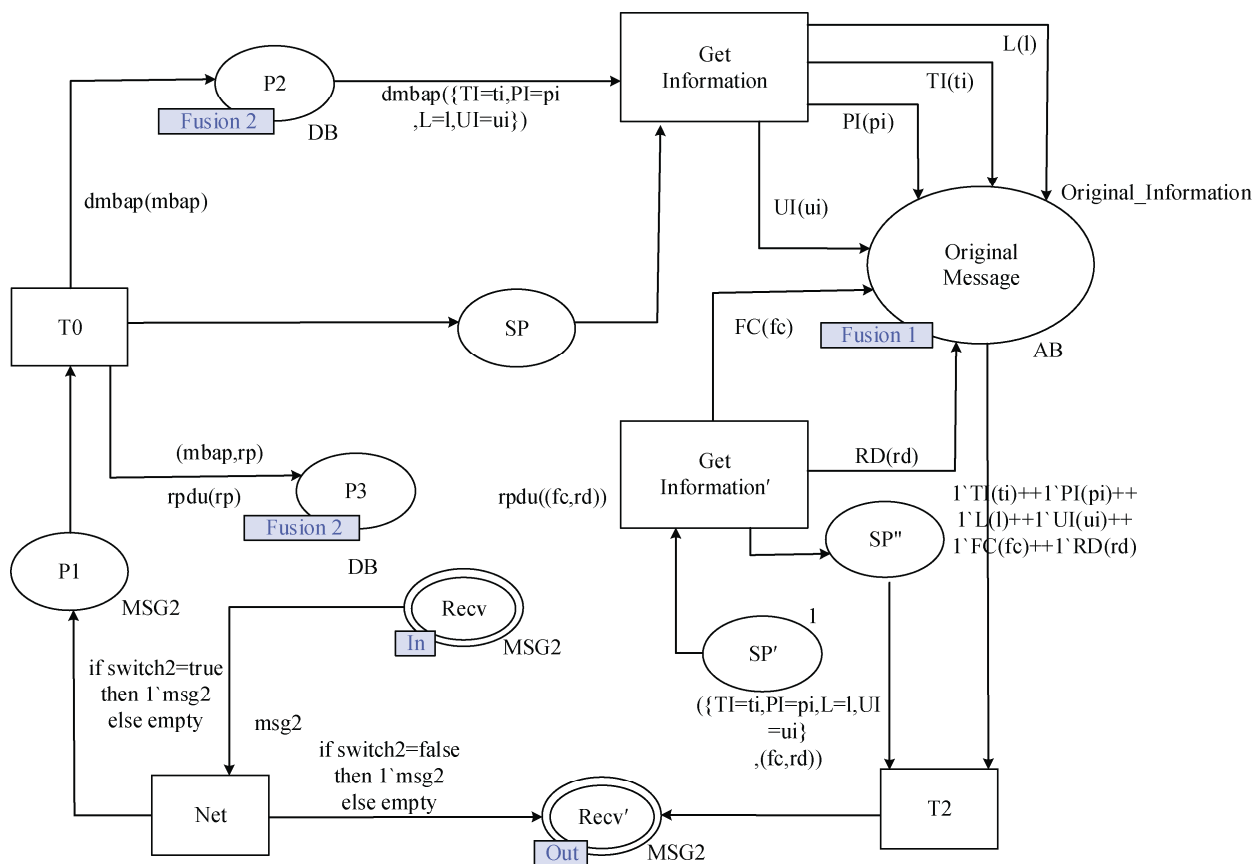
 Figure 6 Internal model of the substitution transition $Adversary_Net1$


图 7 替代变迁 Adversary_Net2 内部模型

 Figure 7 Internal model of the substitution transition $Adversary_Net2$

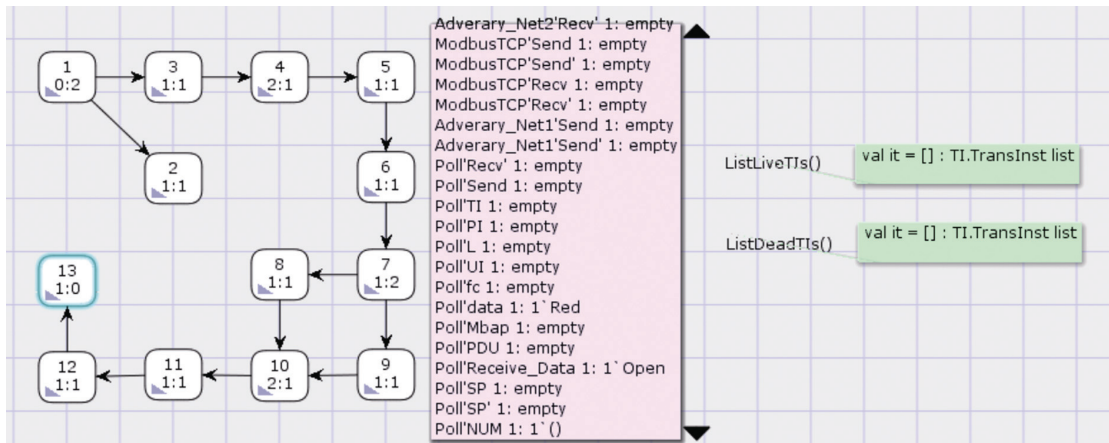


图 8 Modbus/TCP 原始模型运行查询结果

Figure 8 Query results of the Modbus/TCP original model

表 5 Modbus/TCP 协议攻击模型状态空间对比表

Table 5 Comparison of state spaces for Modbus/TCP protocol attack models

类型	原始模型	重放攻击/DoS 攻击	中间人攻击
状态空间节点	13	2570	936
有向弧	14	6163	2241
强联通节点	13	2570	936
强连通弧	14	6163	2241
死节点	1	64	48
死变迁	0	4	0

引入中间人攻击后的 Modbus/TCP 协议模型中死变迁个数为零, 说明该模型中所有变迁均被激活且该协议也没有检测到自身被攻击, 协议保持正常运行状态。其中, 48 个死节点表示此模型存在 48 种模型最终的停止状态。

5.2 Modbus/TCP 协议安全测试分析规则

根据 Modbus/TCP 协议模型攻击者所具备的能力, 使用状态空间分析工具查找具体的攻击路径。采用 $S_0 \sim S_6$ 对不同条件下满足条件的死节点进行分类, 每一类所具有的状态数量和含义见表 6。根据攻击者所重放消息内容的不同进行分类, 第一类(S_1): 攻击者将窃听到的消息复制后直接发送至接收端; 第二类(S_2): 攻击者将根据自己的知识库所, 重新组合出一个新的消息发送至接收端。

规则 1 $S_1 : A \xrightarrow{MSG} B, IN(A) \xrightarrow{MSG} B$

规则 2 $S_2 : A \xrightarrow{MSG} B, IN(A) \xrightarrow{MSG'} B$

根据攻击者进行中间人攻击规则的不同进行分类, 第一类(S_3): 攻击者伪装成合法的主从站与该双方进行通信, 然而合法的主从站认为他们正在直接通信; 第二类(S_4): 攻击者伪装成合法的从站与主站

进行通信, 而主站认为在和真正的从站进行通信并且从站并不知晓有主站在与之通信; 第三类(S_5): 攻击伪装成合法的主站, 拦截从站的响应消息, 使得主站认为从站并未接收到请求消息, 而从站认为在与真正的主站进行正常通信。

规则 3 $S_3 : A \xrightarrow{MSG1} B, IN(A) \xrightarrow{MSG1} B,$
 $B \xrightarrow{MSG2} A, IN(B) \xrightarrow{MSG2} A$

规则 4 $S_4 : A \xrightarrow{MSG1} B, IN(B) \xrightarrow{MSG2} A$

规则 5 $S_5 : IN(A) \xrightarrow{MSG1} B, B \xrightarrow{MSG2} IN(A)$

表 6 死节点分类规则

Table 6 Classification rules for dead markings

名称(数量)	含义
$S_0(64)$	整个模型的死节点数量
$S_1(1)$	攻击者发起重放攻击的死节点数量
$S_2(6)$	攻击者发起 DoS 攻击的死节点数量
$S_3(24)$	第一类中间人攻击的死节点数量
$S_4(2)$	第二类中间人攻击的死节点数量
$S_5(4)$	第三类中间人攻击的死节点数量

5.3 搜索攻击路径

表 7 列出对死节点分类过程中主要用到的 SML 函数。以表 7 中 S_1 函数为例, 此函数可以根据预设的状态节点, 根据库所 NUM、Reply_Check 和 Receive_Check 所包含数据集和数量完成对死节点的抽取; exists(p, l) 函数表明在列表 l 中查询是否包含 p 元素。

为进一步分析攻击者发起重放攻击的路径, 通过查看状态空间结果可以发现: 当协议运行结束后, 从站库所 NUM 中包含两个空白颜色集, 主站库所 Receive_Data 中接收到目标从站的状态数据。并且根

表 7 死节点查询函数
Table 7 Query functions of dead markings

名称	SML 语句	结果
S_0	length(ListDeadMarkings())	64
S_1	<pre> val AttLog2 = let fun checkPlace1 x=Mark.Poll'NUM 1 x fun checkPlace2 y=Mark.Slave'NUM 1 y fun checkPlace3 z=Mark.Poll'Reply_Check 1 z fun checkPlace4 w=Mark.Slave'Receive_Check 1 w fun exists(p,l)=if l = [] then false else if p=(List.hd l) then true else exists(p,List.tl l); In PredAllNodes (fn a => size(checkPlace1 a)=1 andalso size(checkPlace2 a)=2 andalso (OutArcs a)=[] andalso exists(List.hd (checkPlace3 a), (List.hd (checkPlace4 a)))=true andalso List.hd (List.hd (checkPlace4 a))=List.hd (List.tl (List.hd (checkPlace4 a)))) end; val AttLog3 = let fun checkPlace1 x = Mark.Poll'NUM 1 x fun checkPlace2 y = Mark.Slave'NUM 1 y fun checkPlace3 z = Mark.Poll'Reply_Check 1 z fun checkPlace4 w = Mark.Slave'Receiv_Check 1 w fun exists(p,l)= if l = [] then false Else if p=(List.hd l) then true else exists(p,List.tl l); in PredAllNodes (fn a =>size(checkPlace1 a)=1 andalso size(checkPlace2 a)=2 andalso (OutArcs a)=[] andalso exists(List.hd (checkPlace3 a), (List.hd (checkPlace4 a)))=true andalso List.hd (List.hd (checkPlace4 a))<>List.hd (List.tl (List.hd (checkPlace4 a)))) end; val AttLog2 = let fun checkPlace1 x=Mark.Poll'ReceiveData 1 x fun checkPlace2 y=Mark.Slave'NUM 1 y fun checkPlace3 z=Mark.Slave'Receive_Check 1 z In PredAllNodes (fn a =>size(checkPlace1 a)<0 andalso size(checkPlace2 a)=0 andalso List.hd (checkPlace3 a)=empty andalso (OutArcs a)=[]) end; val AttLog3 = let fun checkPlace1 x=Mark.Poll'ReceiveData 1 x fun checkPlace2 y=Mark.Slave'NUM 1 y In PredAllNodes (fn a =>size(checkPlace1 a)=0 andalso size(checkPlace2 a)=1 andalso (OutArcs a)=[]) end; end; </pre>	1
S_2		6
S_4	<pre> val AttLog2 = let fun checkPlace1 x=Mark.Poll'ReceiveData 1 x fun checkPlace2 y=Mark.Slave'NUM 1 y fun checkPlace3 z=Mark.Slave'Receive_Check 1 z In PredAllNodes (fn a =>size(checkPlace1 a)<0 andalso size(checkPlace2 a)=0 andalso List.hd (checkPlace3 a)=empty andalso (OutArcs a)=[]) end; val AttLog3 = let fun checkPlace1 x=Mark.Poll'ReceiveData 1 x fun checkPlace2 y=Mark.Slave'NUM 1 y In PredAllNodes (fn a =>size(checkPlace1 a)=0 andalso size(checkPlace2 a)=1 andalso (OutArcs a)=[]) end; end; </pre>	2
S_5	<pre> val AttLog2 = let fun checkPlace1 x=Mark.Poll'ReceiveData 1 x fun checkPlace2 y=Mark.Slave'NUM 1 y In PredAllNodes (fn a =>size(checkPlace1 a)=0 andalso size(checkPlace2 a)=1 andalso (OutArcs a)=[]) end; end; </pre>	4

据从站接收库所中可以发现, 从站接收到的两次数据均为 ((TI="0001",PI="0000",L="0006",UI="01"), ("0x01",Red)), 并且从站对接收到的两次数据均进行正常执行过程, 从而接收方未发现已遭受重放攻击。从图 5 中的替代变迁 Slave 内部模型和图 6 中的替代变迁 Adversary_Net1 内部模型可以看出, 攻击者通过库所 OriginalMessage 构造消息 MSG2 并发送至接收端。由于接收端并未实施任何身份验证或时间戳验证机制, 接收到的消息仅通过检查事务处理标识或协议标识是否正确来判断其有效性。若标识正确, 消息将被正常接收。因此, 攻击者能够伪造大量无效

请求, 进而耗尽目标系统的资源, 证明 Modbus/TCP 协议中存在 DoS 攻击漏洞及未授权访问漏洞。

针对中间人攻击的具体攻击路径, 使用 NodeDescriptor()函数对相应类型的死节点信息进行查询并进行验证分析。根据中间人攻击规则 S_3 , S_4 和 S_5 对 Modbus/TCP 协议进行验证分析, 其中在该阶段, 配置项 switch1 和 switch2 的值均为 true, 即攻击者对请求消息 MSG1 和响应消息 MSG2 均进行恶意操作。从而, 根据中间人攻击规则对攻击具体路径进行查询。

1) 第一类中间人攻击: 通过 NodeDescriptor()函数对编号为 934 的死节点进行分析, 分析结果表明, 尽管主站与从站之间的通信一直保持正常, 实际上攻击者已对消息流进行了篡改。从图 5 中的替代变迁 Slave 内部模型、图 6 中的替代变迁 Adversary_Net1 内部模型以及图 7 中的替代变迁 Adversary_Net2 内部模型可以看出, 在主站向从站发送请求的过程中, 原始请求消息(库所 Reply_Check 中的内容为 ((TI="0001",PI="0000",L="0006",UI="01"),("0x01",Red)))经 Adversary_Net1 模型时, 攻击者首先对接收到的消息进行拆分并保存至库所 OriginalMessage。库所 OriginalMessage 中包含了攻击者原有的知识及新的消息内容, 基于这些信息, 攻击者能够自由组合新的消息并发送至从站。通过查询从站库所 Receive_Check 中的内容, 发现此消息为 ((TI="0001",PI="0000",L="0006",UI="01"),("0x05",RedOpen)), 其中显示了与原请求不同的指令集。这表明攻击者成功介入并篡改了消息传输过程, 而从站未察觉到该篡改行为。在从站返回响应消息的过程中, 通过 Adversary_Net2 模型可见, 攻击者再次拆分此消息, 并根据 OriginalMessage 库所中的信息进行重新组合, 构造了一个伪造的响应报文 ((TI="0001",PI="0000",L="0006",UI="01"),("0x01",open)), 该报文与从站构造的消息 ((TI="0001",PI="0000",L="0006",UI="01"),("0x05",open))的内容不一致, 但主站最终接收并保存了状态数据("open")。由于该消息采用明文传输方式, 主站无法判断消息内容的真实性, 主站仅通过发送消息与接收消息的数据帧是否一致判断消息的正确性, 因此未能发现整个消息传输过程已被攻击者所操控。本研究分析表明, 在攻击者篡改主从站之间的通信消息的情况下, 主站与从站均未能察觉到通信协议已遭篡改, 暴露了现有 Modbus/TCP 协议在缺乏身份认证机制及消息加密等安全措施下的漏洞。第一类中间人攻击路径查询结果如图 9 所示。

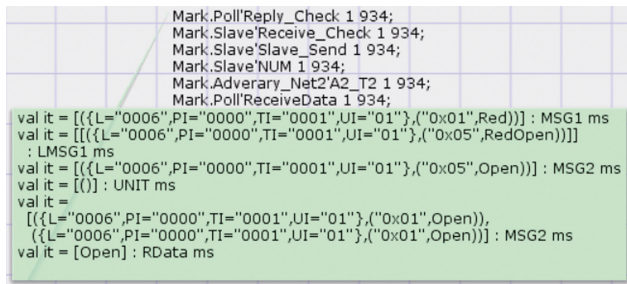


图 9 934 节点的详细信息

Figure 9 Detailed information of node 934

2) 第二类中间人攻击: 通过 NodeDescriptor()函数对编号为 117 的死节点进行分析, 分析结果表明, 攻击者伪装为从站与主站进行通信, 而真实从站并未接收到任何消息。从图 5 中的替代变迁 Slave 内部模型和图 6 中的替代变迁 Adversary_Net1 内部模型可以发现, 主站发送请求消息((TI="0001",PI="0000",L="0006",UI="01"),("0x01",Red))至从站。在攻击者通过 Adversary_Net1 模型截获主站发送至从站的请求消息后, 攻击者对该消息进行拆分, 并根据库所 OriginalMessage 中保存的消息, 伪造了一个响应报文((TI="0001",PI="0000",L="0006",UI="01"),("0x0",Close))。然而, 实际从站设备通过 Slave 模型中的库所 NUM 可以确认, 从站并未发送任何请求, 也未执行任何命令。主站接收到伪造的响应消息后, 根据 Poll 模型中的库所 RData 查询到“Close”状态数据。由于攻击者伪造的响应报文中的数据帧与主站发送的请求报文格式一致, 主站未能察觉到该消息来自伪装的从站, 最终接收并保存了由攻击者伪装的状态数据。这表明攻击者成功地干扰了通信过程, 伪造了从站的响应信息, 从而使主站误以为其接收到来自合法从站的有效数据。具体第二类中间人攻击路径由图 10 所示。

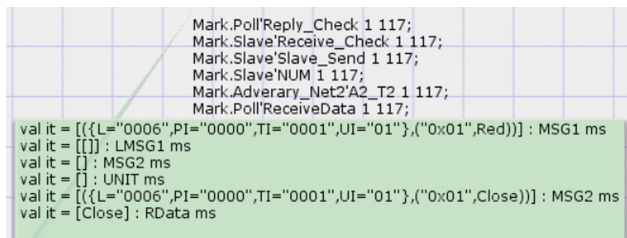


图 10 117 节点的详细信息

Figure 10 Detailed information of node 117

3) 第三类中间人攻击: 攻击者拦截了主站发送的消息, 并伪装成主站向从站发送请求消息, 然而主站并未接收到从站的响应消息。从图 5 中的替代变迁 Slave 内部模型、图 6 中的替代变迁 Adversary_Net1

内部模型以及图 7 中的替代变迁 Adversary_Net2 内部模型可以看出, 主站通过库所 Send 发送请求消息((TI="0001",PI="0000",L="0006",UI="01"),("0x01",Red))。在经过 Adversary_Net1 模型后, 攻击者通过变迁 GetInformation 拆分消息, 并利用库所 OriginalMessage 伪造消息((TI="0001",PI="0000",L="0006",UI="01"),("0x05",RedClose)), 该消息的功能码和数据已与原始消息完全不同。当从站接收到伪造的消息后, 仅通过变迁 WhetherResponse 验证消息的事务处理标识和协议标识是否正确, 确认无误后返回报文。在经过 Adversary_Net2 模型后, 攻击者结束了通信。因此, 在 Poll 内部模型中, 库所 RData 保持为空, 表明主站未收到来自从站的任何有效消息。具体第三类攻击者攻击路径如图 11 所示。

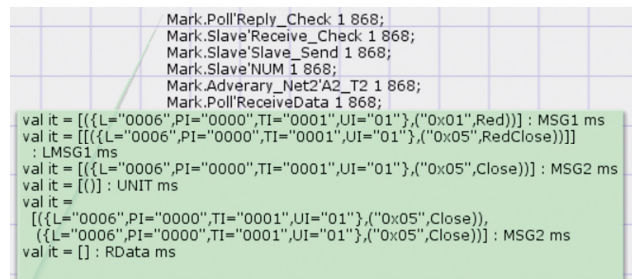


图 11 868 节点的详细信息

Figure 11 Detailed information of node 868

6 实验平台搭建

普渡企业参考架构^[54](Purdue Enterprise Reference Architecture, PERA)作为一种工业系统架构模式, 其主要功能在于为制造企业的信息系统及控制系统的设计与整合提供一套系统化的框架与方法。本研究构建了一种基于物联网开放式软件的工业互联网安全检测与分析虚实实验架构, 具体架构如图 12 所示。根据 PERA 的指导, 本研究所构建的工业互联网安全检测与分析架构主要分为四个层级, 具体包括: ①传感器设备层, 涵盖传感器及相关信息采集设备; ②指令接收层, 其中的 PLC 主要负责接收来自 HMI 或设备管理服务服务器的指令, 并向下游设备传达运行指令; ③指令发送层, HMI 可直接向 PLC 发送操作命令, 安全评估设备则可以通过网关连接至具体的 PLC 设备并下达攻击指令; ④设备管理层, 该层主要涉及非工控系统(ICS)设备, 负责监控和控制 ICS 设备产出的数据。

6.1 执行环境

通过上述基于物联网开放式软件的工业互联网安全检测与分析虚实实验架构, 本研究搭建工业互联网安全检测与分析虚实实验平台, 平台分为三个

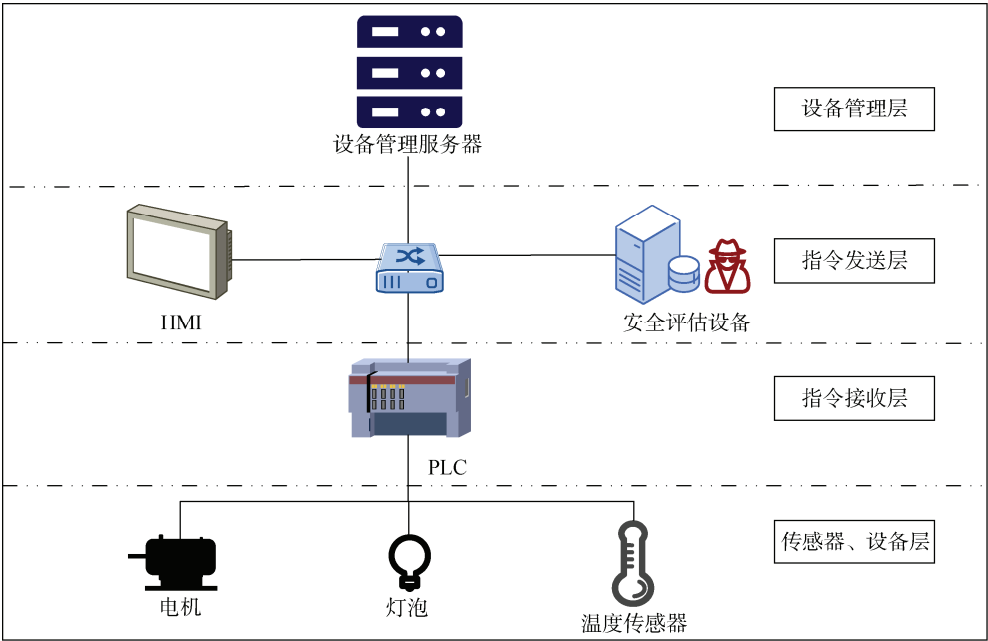


图 12 工业互联网安全检测与分析虚实实验架构

Figure 12 Industrial internet security detection and analysis virtual-physical experimental architecture

独立部分，具体如下：第一部分为奇安信工业控制安全仿真平台，该平台能够在虚拟环境中复现真实工业网络及其运行环境，通过虚拟仿真技术重现工业环境的网络、主机、流量及潜在威胁，并能接入实际工业设备与 IO 模块，以复现完整的生产工艺流程。第二部分是寄云工业互联网开发平台，此平台支持对多元异构数据进行有效的访问控制，拥有数据访问层的抽象处理能力，可通过物模型实现物理及

逻辑实体到资产的映射，构建各类公式指标，实现基于模型和资产属性的规则及报警管理。第三部分包括六方云工业控制仿真平台和 Tridium 物联网开放式软件平台，这两个平台以提供高效的通信服务为核心，涵盖了工业自动化控制设备和现场常用的典型设备，包括主流的 PLC、HMI 等。工业互联网安全检测与分析虚实实验平台中具体设备组件与消息传输过程如图 13 所示。

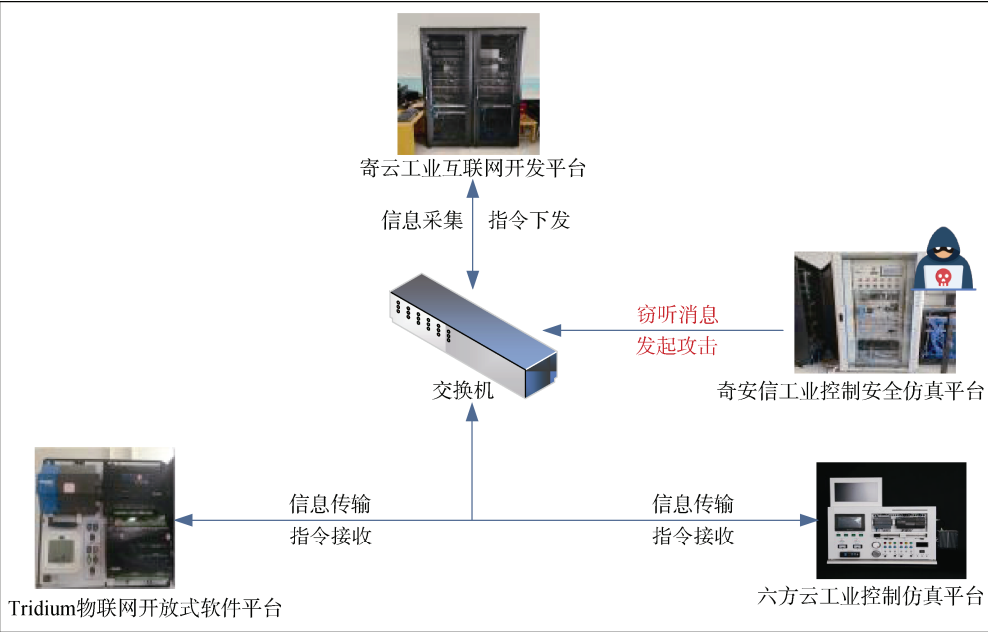


图 13 工业互联网安全检测与分析虚实实验平台

Figure 13 Industrial internet security detection and analysis virtual-physical experimental platform

工业互联网安全检测与分析虚实实验平台中, 三个部分间通过有线网络连通。寄云工业互联网开发平台负责采集以及监控六方云工业控制仿真平台和 Tridium 物联网开放式软件平台。奇安信工业控制安全仿真平台模拟工业控制场景或者直接对六方云工业控制仿真平台和 Tridium 物联网开放式软件平台进行安全评估。其中, 工业互联网安全检测与分析虚实实验平台中包括常见的 ICS 通用协议, 如 Modbus/TCP、OPC UA、S7、EtherNet/IP 和 MQTT 等。为了测试 Smod 渗透测试分析方法的安全评估能力, 本研究使用 Smod 渗透测试分析方法模拟攻击者, 对六方云工业控制仿真平台中的和利时 PLC 进行攻击。具体工控实验场景如图 14 所示。

6.2 测试验证策略

通过表 3 可以发现, Smod 渗透测试库函数的攻击类型主要分为未授权访问攻击和 DoS 攻击两类。因此本研究拟在两类攻击类型下各选取一个库函数以验证此攻击者的攻击路径以及攻击能力。①使用 Smod

测试框架中的 modbus/scanner/uid 模块扫描目标 Modbus 网络, 查看此模块是否能识别并记录目标设备的唯一标识符并发送查询以确定每个设备的 UID, 此模块用于测试 Modbus/TCP 协议中存在的未授权访问漏洞; ②使用 Smod 测试框架的 modbus/sniff/arp 模块嗅探和分析目标 Modbus 网络上的 ARP 请求和响应, 并向目标 PLC 持续发送报文, 此模块用于测试 Modbus/TCP 协议是否可以抵抗重放或 DoS 攻击。

调用 modbus/scanner/uid 模块对 IP 地址为 192.168.3.140 的 PLC 设备实施 UID 暴力破解攻击, 通过图 15 可以发现, 此攻击模块可以成功破解 Modbus/TCP 协议中的 UID 为 1。调用 modbus/sniff/arp 模块, 通过配置攻击模块中的主站和目标 PLC 设备 IP 地址参数并启动攻击。实验结果显示, 对目标 PLC 设备进行了 ARP 洪泛攻击后, 目标 PLC 设备在收到 HMI 发送的红灯开启命令后, 无法正常对指定红灯进行开启, 从而证实了攻击的成功实施。具体实验配置如图 16 所示。

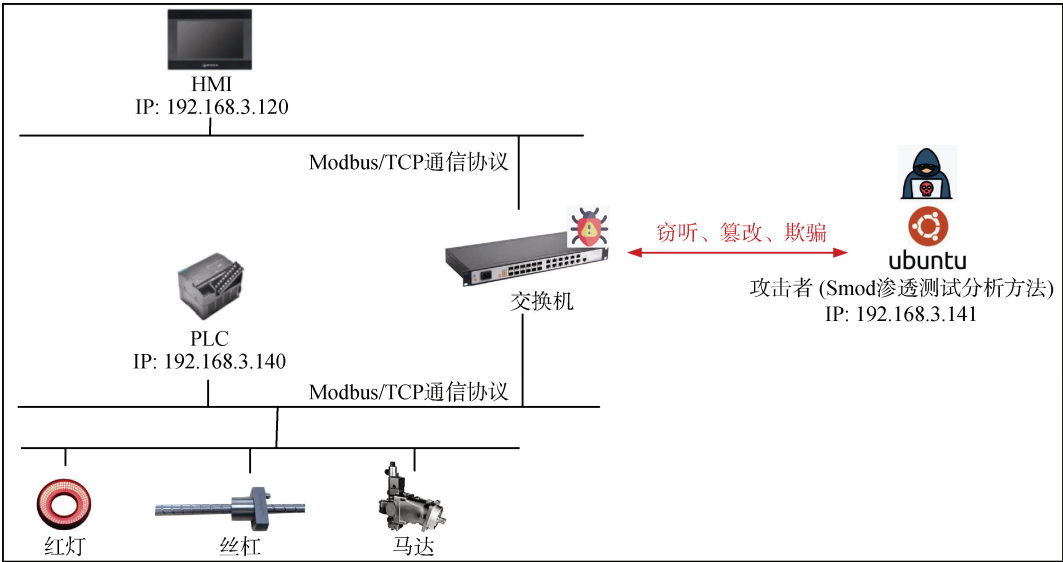


图 14 Smod 渗透测试分析方法测试评估架构

Figure 14 Testing and evaluation framework for the Smod penetration testing analysis method

```
modbus/scanner/getfunc
modbus/scanner/uid
modbus/scanner/uid
SMOD > use modbus/scanner/uid
SMOD modbus(uid) > set RHOSTS 192.168.3.140
SMOD modbus(uid) > ex
exploit
exit
SMOD modbus(uid) > exploit
[+] Module Brute Force UID Start
[+] Start Brute Force UID on : 192.168.3.140
[+] UID on 192.168.3.140 is : 1
```

图 15 modbus/scanner/uid 模块测试

Figure 15 Testing and evaluation of modbus/scanner/uid module

通过对 Smod 渗透测试分析工具的源代码进行

详细审查, 可以发现该工具在检测 Modbus/TCP 协议中存在的未授权访问漏洞方面, 采用了以下方法: 首先, 工具会对指定的 IP 地址范围进行遍历, 并为每个 IP 地址启动多个线程以实现并发处理。在每个线程中, 程序通过构造 Modbus 应用数据单元和 Modbus 协议数据单元来生成 Modbus 请求报文。随后, 程序发送这些请求报文以测试不同的 UID。若目标设备对某一 UID 返回有效响应, 则表明该 UID 为有效标识符, 程序将输出该 UID。在评估 Modbus/TCP 协议对重放攻击或 DoS 的抵抗能力时,

程序首先获取目标设备的 MAC 地址, 并启用 IP 转发功能。接着, 程序通过发送伪造的 ARP 数据包, 将网关的 IP 地址与目标设备的 MAC 地址进行绑定, 同时将目标设备的 IP 地址与网关的 MAC 地址进行绑定。这一操作导致目标设备与网关之间的通信被中断并重定向。通过持续发送伪造的 ARP 数据包, 攻击得以持续进行, 直至用户手动终止该过程。

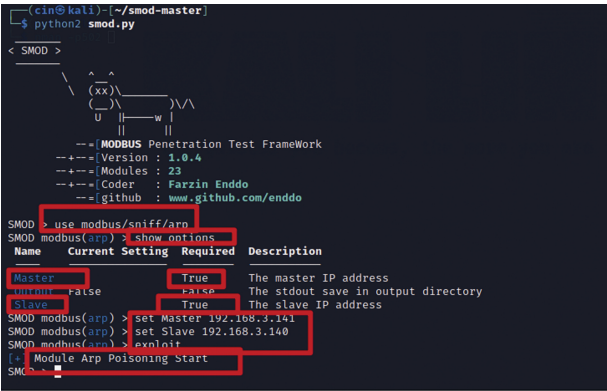


图 16 modbus/sniff/arp 模块测试

Figure 16 Testing and evaluation of modbus/sniff/arp module

然而, Smod 渗透测试工具在进行 Modbus/TCP 协议的中间人攻击测试时, 存在一定的局限性, 主要体现在其攻击方法的固定性上。Smod 工具主要基于预定义的读写线圈或寄存器操作发起攻击, 其内置脚本依赖于这些固定的读写操作来执行渗透测试。然而, Smod 的攻击逻辑是以预定义的读写线圈和寄存器操作为核心进行设计的, 其无法根据实际攻击需求动态调整或组合攻击代码。这种固定的攻击模式限制了 Smod 在不同 Modbus/TCP 通信场景中的适应性, 使其在面对多样化的攻击目标或更为复杂的攻击路径时, 难以充分发挥其效能。因此, Smod 在复杂的中间人攻击测试中的适用性受到显著限制。

为提升 Smod 渗透测试工具在 Modbus/TCP 协议中间人攻击测试中的适用性, 可以考虑从多个方面进行改进。首先, 增加动态攻击脚本生成机制, 使其能根据实时网络环境和攻击目标灵活调整攻击策略, 避免固定的读写操作限制。其次, 结合机器学习或人工智能算法, 实时优化攻击行为, 增强应对复杂攻击场景的能力。此外, 将 Smod 与其他攻击工具结合, 创建混合式渗透测试框架, 提升对多样化攻击路径的适应性。这些改进将有效突破 Smod 的局限, 使其可以创建更复杂的攻击测试。

6.3 评估方法对比

本研究通过 Smod 渗透测试分析方法对 Modbus/TCP 协议的安全性进行了实验性评估, 具体

对比结果如表 8 所示。Smod 渗透测试分析方法能根据不同攻击模块的脚本, 有效地实施重放攻击、拒绝服务攻击及未授权访问攻击。然而, 与 CPN 形式化分析方法相比, Smod 渗透测试分析方法在识别中间人攻击方面显示出明显的局限性。由于 Smod 渗透测试模块被置于通信双方中间, 只有当攻击者窃听到消息后, 才能根据消息内容进行内容篡改, 或向消息接收方重复发送此消息。相对而言, CPN 形式化分析方法能够根据预置的知识和窃听到的消息重构消息内容, 假装为合法的通信方, 从而欺骗协议的接收方实施中间人攻击。这一优势使得 CPN 形式化分析方法通过状态空间分析工具, 能够清晰描述出攻击者如何利用漏洞重构消息并进行欺骗, 揭示潜在的攻击路径, 从而比 Smod 方法更为灵活和强大。

表 8 CPN 与 Smod 安全性验证结果对比

Table 8 Comparison of security verification results between CPN and Smod

安全分析方法	重放攻击	DoS 攻击	中间人攻击
Smod 渗透测试分析方法	√	√	×
CPN 形式化分析方法	√	√	√

此外, 将使用 CPN 形式化分析方法的结果与其他文献中基于 Modbus/TCP 协议安全性评估的分析结果相比较, 文献[27-29]和[31]均使用通信数据包判断协议中存在的安全漏洞, 无法提供数据包可能被篡改的具体位置, 从而无法提供协议中具体存在的攻击路径。文献[32-34]通过搭建工业控制环境, 使用 Ettercap、Wireshake 和 Smod 等渗透测试工具, 通过在具体位置部署攻击者对协议进行安全测试。但是此类方法通过撰写固定攻击脚本, 只能发现固定的安全漏洞, 并且无法修改攻击者在环境中存在的位置, 从而只能分析协议在一条路径中存在的安全漏洞。反之, 本研究所提出的 CPN 形式化分析方法利用状态空间分析工具, 不仅能发现已知的攻击路径, 还能揭示并清晰描述出更多未知的中间人攻击路径。具体比较结果见表 9。

6.4 基于 Modbus/TCP 协议的中间人攻击模块设计

根据 CPN 形式化分析发现的三种不同的攻击路径, 建议在 Smod 渗透测试框架中加入以下三种攻击, 以增强框架的攻击检测能力, 从而更全面地评估和测试 Modbus/TCP 协议的安全性。

第一类中间人攻击: 攻击者截获主站发送至从站的通信信息, 并对其进行拆解。基于已有的知识和截获的消息, 攻击者重新构造并发送新的信息至接

表 9 CPN 与其他 Modbus/TCP 协议安全性评估方法对比

Table 9 Comparison of CPN and other Modbus/TCP protocol security evaluation methods

方法	篡改攻击	重放攻击	中间人攻击	DoS 攻击	是否检测到数据包受到篡改	是否提供具体攻击路径
文献[27]	√	×	×	√	√	×
文献[28]	×	×	×	√	√	×
文献[29]	×	×	×	√	√	×
文献[31]	×	×	×	×	√	×
文献[32]	√	√	×	×	√	1 条
文献[33]	√	√	√	×	√	1 条
文献[34]	×	×	×	√	√	×
本文方案	√	√	√	√	√	5 条

收端。与此同时,攻击者在接收到返回消息后,会首先验证该消息是否符合主站所需的正确响应。如果符合,则继续利用已掌握的信息生成相应报文,并将其发送回主站。攻击者循环上述操作,使主站与从站误以为其间的通信是真正的、有效的交互。具体第一类中间人攻击步骤如算法 1 所示。

算法 1 第一类中间人攻击

输入: 攻击者 A , 主站 S , 从站 R , 攻击者知识库所 $OriginalMessage$, 主站发送至从站的消息 msg_1 。

输出: 攻击者将消息 msg_1' , msg_2' 发送至主站和从站。

Intercept (msg_1) from S

($mbap, fc, sd$) $\leftarrow$ Splite (msg_1)

$OriginalMessage \leftarrow Store (mbap, fc, sd)$

WHILE $i < NumberofPackets$ DO

($mbap, fc', sd'$) $\in OriginalMessage$

$msg_1' \leftarrow CreateModbusPacket (mbap, fc', sd')$

Send (msg_1') to R

Intercept (msg_2)

($mbap, fc', sd'$) $\in OriginalMessage$

IF $mbap = mbap'$ and $fc = fc'$

$msg_2' = CreateModbusPacket (mbap', fc', rd)$

Send (msg_2') to S

$i \leftarrow i + 1$

第二类中间人攻击: 攻击者直接截获主站发送给从站的通信信息, 并对其进行拆解。基于已有的知识及截获的数据, 攻击者重构主站所需的正确报文, 并伪装成合法的从站, 将构造的报文发送回主站。这样, 主站误以为其正在与真实的从站进行通信, 然而实际上从站并未接收到任何信息。具体第二类中间人攻击步骤如算法 2 所示。

算法 2 第二类中间人攻击

输入: 攻击者 A , 主站 S , 攻击者知识库所 $OriginalMessage$, 主站发送给从站的消息 msg_1 。

输出: 攻击者将消息 msg_2 发送至主站。

Receive (msg_1) from S

($mbap, fc, sd$) $\leftarrow$ Splite (msg_1)

WHILE $i < NumberofPackets$ DO

($mbap, fc', sd'$) $\in OriginalMessage$

IF $mbap = mbap'$ and $fc = fc'$

$msg_2' = CreateModbusPacket (mbap', fc', rd)$

Send (msg_2') to S

$i \leftarrow i + 1$

第三类中间人攻击: 攻击者可以基于算法 1 和算法 2 中窃听到的消息内容及已有的知识, 拦截合法主站发送给从站的请求报文, 并伪装成合法的主站。攻击者通过构造请求报文, 使主站误认为从站未接收到请求消息, 并持续向目标从站发送伪造的通信报文。同时, 从站则误以为其正在与真实的主站进行正常通信。具体第三类中间人攻击步骤如算法 3 所示。

算法 3 第三类中间人攻击

输入: 攻击者 A , 主站 S , 攻击者知识库所 $OriginalMessage$, 主站发送给从站的消息 msg_1 。

输出: 攻击者将消息 msg_2 发送至主站。

Receive (msg_1) from S

($mbap, fc, sd$) $\leftarrow$ Splite (msg_1)

WHILE $i < NumberofPackets$ DO

($mbap, fc', sd'$) $\in OriginalMessage$

IF $mbap = mbap'$ and $fc = fc'$

$msg_2' = CreateModbusPacket (mbap', fc', rd)$

Send (msg_2') to S

$i \leftarrow i + 1$

7 结束语

本研究针对 Modbus/TCP 协议的安全性进行了深入分析与评估, 主要工作包括三个方面。首先, 本研究利用有色 Petri 网形式化分析方法结合 Dolev-Yao 攻击者模型对 Modbus/TCP 协议进行建模, 通过状态空间分析工具计算并验证协议模型的正确性, 同时撰写查询函数进一步识别协议中的具体攻击路径。其次, 本研究构建了一个工业互联网安全监测与分析实验平台, 在该平台上通过 Modbus/TCP 协议实现 HMI、PLC 和设备节点之间的通信, 并使用 Smod 渗透测试方法对协议进行安全性评估, 验证了 Smod 库函数攻击模块的有效性。最后, 通过将 Smod 渗透测试结果及其他 Modbus/TCP 协议安全评估方法结果与 CPN 形式化分析方法的结果进行对比分析, 发现 CPN 形式化分析方法能够识别更多的 Modbus/TCP 协议潜在攻击漏洞, 并提供具体的攻击路径。通过研究结果可以发现, CPN 形式化分析方法在识别工业协议潜在攻击漏洞和揭示具体攻击路径方面具有更高的有效性和准确性, 相较于其他评估方法具有明显优势。此发现为进一步提升工业控制系统的安全性提供了新的思路和方法, 有助于构建更加安全、可靠的工业网络环境。

参考文献

- [1] Mohy-Eddine M, Guezzaz A, Benkirane S, et al. An Ensemble Learning Based Intrusion Detection Model for Industrial IoT Security[J]. *Big Data Mining and Analytics*, 2023, 6(3): 273-287.
- [2] Kumar R, Agrawal N. Analysis of Multi-Dimensional Industrial IoT (IIoT) Data in Edge-Fog-Cloud Based Architectural Frameworks: A Survey on Current State and Research Challenges[J]. *Journal of Industrial Information Integration*, 2023, 35: 100504.
- [3] Babayigit B, Abubaker M. Industrial Internet of Things: A Review of Improvements over Traditional SCADA Systems for Industrial Automation[J]. *IEEE Systems Journal*, 2024, 18(1): 120-133.
- [4] Xu H S, Wu J, Pan Q Q, et al. A Survey on Digital Twin for Industrial Internet of Things: Applications, Technologies and Tools[J]. *IEEE Communications Surveys & Tutorials*, 2023, 25(4): 2569-2598.
- [5] Khan R, Maynard P, McLaughlin K, et al. Threat Analysis of BlackEnergy Malware for Synchrophasor Based Real-Time Control and Monitoring in Smart Grid[J]. *Electronic Workshops in Computing*, 2016: 53-63.
- [6] Jiang X B, Lora M, Chattopadhyay S. An Experimental Analysis of Security Vulnerabilities in Industrial IoT Devices[J]. *ACM Transactions on Internet Technology*, 2020, 20(2): 1-24.
- [7] Serror M, Hack S, Henze M, et al. Challenges and Opportunities in Securing the Industrial Internet of Things[J]. *IEEE Transactions on Industrial Informatics*, 2021, 17(5): 2985-2996.
- [8] Choo K-K R, Yan Z, Meng W. Blockchain in industrial IoT applications: Security and privacy advances, challenges and opportunities[J]. *IEEE Transactions on Industrial Informatics*, 2020, 16 (6): 4119-4121.
- [9] Tange K, De Donno M, Fafoutis X, et al. A Systematic Survey of Industrial Internet of Things Security: Requirements and Fog Computing Opportunities[J]. *IEEE Communications Surveys & Tutorials*, 2020, 22(4): 2489-2520.
- [10] Conti M, Donadel D, Turrin F. A Survey on Industrial Control System Testbeds and Datasets for Security Research[J]. *IEEE Communications Surveys & Tutorials*, 2021, 23(4): 2248-2294.
- [11] Buscemi A, Turcanu I, Castignani G, et al. CANMatch: A Fully Automated Tool for CAN Bus Reverse Engineering Based on Frame Matching[J]. *IEEE Transactions on Vehicular Technology*, 2021, 70(12): 12358-12373.
- [12] Yu K P, Tan L, Mumtaz S, et al. Securing Critical Infrastructures: Deep-Learning-Based Threat Detection in IIoT[J]. *IEEE Communications Magazine*, 2021, 59(10): 76-82.
- [13] Wei X Y, Yan Z, Liang X Q. A Survey on Fuzz Testing Technologies for Industrial Control Protocols[J]. *Journal of Network and Computer Applications*, 2024, 232: 104020.
- [14] Bella G, Biondi P, Bognanni S, et al. PETIoT: PENetration Testing the Internet of Things[J]. *Internet of Things*, 2023, 22: 100707.
- [15] Jiang H J. Formal analysis of security protocols based on model checking[D]. Shanghai: East China Normal University, 2022. (蒋洪剑. 基于模型检测方法的安全协议形式化分析[D]. 上海: 华东师范大学, 2022.)
- [16] Xue R, Feng D G. The Approaches and Technologies for Formal Verification of Security Protocols[J]. *Chinese Journal of Computers*, 2006, 29(1): 1-20. (薛锐, 冯登国. 安全协议的形式化分析技术与方法[J]. *计算机学报*, 2006, 29(1): 1-20.)
- [17] Nardone R, Rodríguez R J, Marrone S. Formal Security Assessment of Modbus Protocol[C]. *2016 11th International Conference for Internet Technology and Secured Transactions*, 2016: 142-147.
- [18] Wang Y S, Fan K F, Lai Y X, et al. Intrusion Detection of Industrial Control System Based on Modbus TCP Protocol[C]. *2017 IEEE 13th International Symposium on Autonomous Decentralized System*, 2017: 156-162.
- [19] Chochtoul D, Ilias A, Stamatiou Y C, et al. Integrating Elliptic Curve Cryptography with the Modbus TCP SCADA Communication Protocol[J]. *Future Internet*, 2022, 14(8): 232.
- [20] Yi F M, Zhang L, Yang S M, et al. A Security-Enhanced Modbus TCP Protocol and Authorized Access Mechanism[C]. *2021 IEEE Sixth International Conference on Data Science in Cyberspace*, 2021: 61-67.
- [21] Ferst M K, de Figueiredo H F M, Denardin G, et al. Implementation of Secure Communication with Modbus and Transport Layer Security Protocols[C]. *2018 13th IEEE International Conference on Industry Applications*, 2018: 155-162.
- [22] Fovino I N, Carcano A, Masera M, et al. Design and Implementa-

- tion of a Secure Modbus Protocol[M]. Critical Infrastructure Protection III. Berlin, HeidelbergSpringer2009: 83-96.
- [23] Katulić F, Sumina D, Groš S, et al. Protecting Modbus/TCP-Based Industrial Automation and Control Systems Using Message Authentication Codes[J]. *IEEE Access*, 2023, 11: 47007-47023.
- [24] Ametov F R, Bekirov E A, Asanov M M. Organizing the Information Security in Modbus TCP Interfaces for Use in the Energy Complex[J]. *IOP Conference Series: Materials Science and Engineering*, 2021, 1089(1): 012007.
- [25] Liu Z C, Liang T, Wang W S, et al. Design and Implementation of a Lightweight Security-Enhanced Scheme for Modbus TCP Protocol[J]. *Security and Communication Networks*, 2023, 2023: 5486566.
- [26] Martins T, Oliveira S V G. Enhanced Modbus/TCP Security Protocol: Authentication and Authorization Functions Supported[J]. *Sensors*, 2022, 22(20): 8024.
- [27] Phillips B, Gamess E, Krishnaprasad S. An Evaluation of Machine Learning-Based Anomaly Detection in a SCADA System Using the Modbus Protocol[C]. *The 2020 ACM Southeast Conference*, 2020: 188-196.
- [28] Che X, Geng Y Y, Zhang G, et al. Fuzzing Technology Based on Information Theory for Industrial Proprietary Protocol[J]. *Electronics*, 2023, 12(14): 3041.
- [29] Zhang H, Min Y D, Liu S Y, et al. MODLSTM: A Method to Recognize DoS Attacks on Modbus/TCP[C]. *2022 IEEE International Performance, Computing, and Communications Conference*, 2022: 319-324.
- [30] Zhang H, Min Y D, Liu S Y, et al. Improve the Security of Industrial Control System: A Fine-Grained Classification Method for DoS Attacks on Modbus/TCP[J]. *Mobile Networks and Applications*, 2023, 28(2): 839-852.
- [31] lv P W, Ge Y C, Li N, et al. Abnormal Detection in Modbus TCP Based on Convolutional Neural Network[J]. *Journal of Information Security Research*, 2019, 5(7): 635-638.
(吕佩吾, 葛雅川, 李楠, 等. 基于卷积神经网络的工控协议 Modbus TCP 异常检测[J]. *信息安全研究*, 2019, 5(7): 635-638.)
- [32] Rajesh L, Satyanarayana P. Detection and Blocking of Replay, False Command, and False Access Injection Commands in SCADA Systems with Modbus Protocol[J]. *Security and Communication Networks*, 2021, 2021: 8887666.
- [33] Kulkarni S, Rahul R K, Shreyas R, et al. MITM Intrusion Analysis for Advanced Metering Infrastructure Communication in a Smart Grid Environment[C]. *Trends in Computational Intelligence, Security and Internet of Things*, 2020: 256-267.
- [34] Radoglou-Grammatikis P, Siniosoglou I, Liatifis T, et al. Implementation and Detection of Modbus Cyberattacks[C]. *2020 9th International Conference on Modern Circuits and Systems Technologies*, 2020: 1-4.
- [35] Li T X, Wang Y, Zou C M, et al. Research on DoS Attack Detection Method of Modbus TCP in OpenPLC[J]. *Journal of Computer and Communications*, 2021, 9(7): 73-90.
- [36] Luswata J, Zavarsky P, Swar B, et al. Analysis of SCADA Security Using Penetration Testing: A Case Study on Modbus TCP Protocol[C]. *2018 29th Biennial Symposium on Communications*, 2018: 1-5.
- [37] Parian C, Guldemann T, Bhatia S. Fooling the Master: Exploiting Weaknesses in the Modbus Protocol[J]. *Procedia Computer Science*, 2020, 171: 2453-2458.
- [38] Dolev D, Yao A. On the Security of Public Key Protocols[J]. *IEEE Transactions on Information Theory*, 1983, 29(2): 198-208.
- [39] Lu S Q, Zhou S Y, Mao Y. Formal Analysis and Optimization of TLS1.3 Protocol in Strong Security Model[J]. *Journal of Software*, 2021, 32(9): 2849-2866.
(陆思奇, 周思渊, 毛颖. 强安全模型下 TLS1.3 协议的形式化分析与优化[J]. *软件学报*, 2021, 32(9): 2849-2866.)
- [40] Perrig A, Szewczyk R, Wen V, et al. SPINS: Security protocols for sensor networks[C]. *Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001: 189-199.
- [41] Zhang X L, Zhu Y F, Gu C X, et al. C2P: Formal Abstraction Method and Tool for C Protocol Code Based on Pi Calculus[J]. *Journal of Software*, 2021, 32(6): 1581-1596.
(张协力, 祝跃飞, 顾纯祥, 等. C2P: 基于 Pi 演算的协议 C 代码形式化抽象方法和工具[J]. *软件学报*, 2021, 32(6): 1581-1596.)
- [42] Blanchet B, Cheval V, Cortier V. ProVerif with Lemmas, Induction, Fast Subsumption, and much More[C]. *2022 IEEE Symposium on Security and Privacy*, 2022: 69-86.
- [43] Gong X, Feng T, Du J Z. Formal Modeling and Security Analysis Method of Security Protocol Based on CPN[J]. *Journal on Communications*, 2021, 42(9): 240-253.
(龚翔, 冯涛, 杜谨泽. 基于 CPN 的安全协议形式化建模及安全分析方法[J]. *通信学报*, 2021, 42(9): 240-253.)
- [44] Bojjagani S, Reddy Y C A P, Anuradha T, et al. Secure Authentication and Key Management Protocol for Deployment of Internet of Vehicles (IoV) Concerning Intelligent Transport Systems[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2022, 23(12): 24698-24713.
- [45] Wang W S, Huang W C, Meng Z Y, et al. Advancing the Automation Capability of Verifying Security Protocols[J]. *IEEE Transactions on Dependable and Secure Computing*, 2024, 21(6): 5059-5070.
- [46] Lu Y, Feng T. Research on Trusted DNP3-BAE Protocol Based on Hash Chain[J]. *EURASIP Journal on Wireless Communications and Networking*, 2018, 2018: 108.
- [47] Feng T, Wang S S, Gong X, et al. Formal Security Evaluation and Improvement of Industrial Ethernet EtherCAT Protocol[J]. *Journal of Computer Research and Development*, 2020, 57(11): 2312-2327.
(冯涛, 王帅帅, 龚翔, 等. 工业以太网 EtherCAT 协议形式化安全评估及改进[J]. *计算机研究与发展*, 2020, 57(11): 2312-2327.)
- [48] Luo F Y, Feng T, Zheng L. Formal Security Evaluation and Improvement of Wireless HART Protocol in Industrial Wireless Network[J]. *Security and Communication Networks*, 2021, 2021: 8090547.
- [49] Xie P S, Zhu J F, Kang Y P, et al. Analysis and Improvement of the BACnet/IP Based on Identity Authentication[J]. *Journal on Communications*, 2024, 45(3): 227-243.
(谢鹏寿, 朱家锋, 康永平, 等. 基于身份认证的 BACnet/IP 分析与改进[J]. *通信学报*, 2024, 45(3): 227-243.)
- [50] Zheng L, Feng T. Research on a Vehicle Authentication and Key

Transmission Protocol Based on CPN[J]. *Symmetry*, 2022, 14(11): 2398.

[51] Du J Z, Jin-Ze Du J L, Jun-Wei Liu T F, et al. Formal Analysis and Improvement of Z-Wave Protocol[J]. *Journal of Computers*, 2023, 34(4): 25-39.

[52] Wu Y, Feng T. An Anonymous Authentication and Key Update Mechanism for IoT Devices Based on EnOcean Protocol[J]. *Sen-*

sors, 2022, 22(17): 6713.

[53] Jensen K, Kristensen L M, Wells L. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems[J]. *International Journal on Software Tools for Technology Transfer*, 2007, 9(3/4): 213-254.

[54] Williams T J. The Purdue Enterprise Reference Architecture[J]. *Computers in Industry*, 1994, 24(2/3): 141-158.



吴毅 于 2024 年在兰州理工大学计算机应用技术专业获得硕士学位。现在兰州理工大学制造业信息化系统专业攻读博士学位。CCF 学生会员。研究领域为网络与信息安全。研究兴趣包括: 工业互联网安全, 工业控制协议安全。Email: 241080292002@lut.edu.cn



冯涛 于 2008 年在西安电子科技大学计算机系统结构专业获得博士学位。现任兰州理工大学计算机与通信学院院长, 博士生导师、CCF 会员。研究领域为网络与信息安全。研究兴趣包括: 区块链、工业互联网安全等。Email: fengt@lut.edu.cn



王龙鑫 于 2023 年在兰州理工大学网络安全专业获得学士学位。现在兰州理工大学计算机科学与技术专业攻读硕士学位。CCF 学生会员。研究领域为工业互联网安全。研究兴趣包括: 工业互联网通信协议安全。Email: 232081203008@lut.edu.cn



马蓉 于 2019 年在福建师范大学计算机应用技术专业获得硕士学位。现在兰州理工大学制造业信息化系统专业攻读博士学位。研究领域为网络与信息安全。研究兴趣包括: 数据安全与隐私保护、区块链安全。Email: MiaRonGer@163.com



石建明 于 2020 年在兰州理工大学控制工程专业获得硕士学位, 现在兰州理工大学计算机与通信学院制造业信息化系统专业攻读博士学位, 研究领域为网络与信息安全。研究兴趣包括: 工业互联网协议安全分析, 数控设备通信安全及 AI 安全分析。Email: xbsjm@163.com