基于杀伤链模型的 PLC 安全分析

孙 越[1,2], 游建舟[1,2], 宋站威[1,2], 黄文军[1,2], 陈 曦[3], 孙利民[1,2]

[1] 中国科学院信息工程研究所 物联网信息安全技术北京市重点实验室, 北京 中国 100093
[2] 中国科学院大学 网络空间安全学院, 北京 中国 100049
[3] 北京大学软件与微电子学院, 北京 中国 102600

摘要 可编程逻辑控制器(Programmable Logic Controller, PLC)是现代工业控制系统中至关重要的组成部分, 其安全性对于维持工业过程的安全和连续运行至关重要。然而, 由于 PLC 特殊的系统架构和通信协议, 缺乏针对其安全性分析的标准框架和程序。网络杀伤链(Cyber Kill Chain)模型是一种被广泛应用于描述入侵者利用漏洞的策略和技术的方法论, 并已被广泛应用于网络安全领域。本文基于杀伤链模型总结了近年来 PLC 安全攻防技术, 旨在为网络安全从业者提供技术参考, 并协助研究人员了解最新进展。首先, 我们介绍了 PLC 的基本架构、工作原理和通信协议, 这对于分析 PLC 的漏洞和攻击至关重要。然后, 我们使用杀伤链模型对各种 PLC 攻击技术进行了详细分类。具体而言, 我们将 PLC 攻击技术分为侦查识别、武器构建、载荷投递、漏洞利用、隐蔽驻留、远程控制和目的实现七个阶段。对于每个阶段, 我们详细分析了攻击者使用的技术。我们的分析有助于全面了解攻击的各个阶段, 并可帮助开发主动的安全措施。除了对 PLC 攻击技术的详细分析, 本文还讨论了多种 PLC 防御技术, 包括协议安全保护、控制程序验证、执行过程监控和 PLC 取证技术。通过总结这些方法, 我们希望为网络安全从业者提供实用的指导, 更好地保护 PLC 免受威胁。此外, 我们从不同的角度, 如嵌入式设备、工业控制器和工业控制网络组件等, 突出了当前 PLC 安全领域的研究趋势, 这可以作为未来研究的路线, 增强关键基础设施的安全防护。

关键词 PLC 安全; 工控系统安全; 杀伤链模型; PLC 攻击技术; PLC 防御技术
中图法分类号 TP391.8 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2025.03.10

A Cyber Kill Chain Based Analysis of PLC Security

SUN Yue[1,2], YOU Jianzhou[1,2], SONG Zhanwei[1,2], HUANG Wenjun[1,2], CHEN Xi[3], SUN Limin[1,2]

[1] Beijing Key Laboratory of IoT Information Security Technology, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
[3] School of Software & Microelectronics, PKU, Beijing 102600, China

Abstract Programmable Logic Controllers (PLCs) are integral components of modern industrial control systems, where their security is crucial for maintaining the safe and continuous operation of industrial processes. However, the unique architecture and communication protocols of PLCs pose a significant challenge for their security analysis, as standard frameworks and procedures are lacking. The Cyber Kill Chain model is a well-established methodology for describing the tactics and techniques used by attackers to exploit vulnerabilities, and it has been widely adopted in the field of cybersecurity. This paper provides an overview of PLC security in recent years, utilizing the Cyber Kill Chain model to present the latest advances in this field. The objective of this paper is to provide a technical reference for cybersecurity practitioners and to facilitate researchers in their understanding of PLC security. Firstly, we introduce the basic architecture, operation principle, and communication protocols of PLCs, which are fundamental to analyzing the vulnerabilities and attacks on PLCs. We then use the Cyber Kill Chain model to classify the various stages of PLC attack techniques, which includes reconnaissance, weaponization, delivery, exploitation, installation, command and control, and execution. For each stage, we provide a detailed analysis of the techniques used by attackers. Our analysis helps to provide a comprehensive view of the various stages of an attack and can aid in developing proactive security measures. In addition to the detailed analysis of PLC attack techniques, we also discuss various techniques for securing PLCs in this article. These include measures such as protocol security protection, control program verification, execution process monitoring, and PLC forensics technology. By highlighting these methods, we hope to provide practical guidance for cybersecurity practitioners to better protect PLCs from threats. Moreover, we also highlight the current research trends on PLC security from different perspectives, such as embedded devices, industrial controllers, and industrial control network components, which can serve as a roadmap for future research in this field, and promote the security and resilience of critical infrastructure.

通讯作者: 宋站威, 硕士, 助理研究员, Email: songzhanwei@iie.ac.cn
本课题得到科技部国家重点研发计划(No. 2018YFC1201102), 国家自然科学基金联合基金项目(No. U1766215), 国家自然科学基金(No. 61702506)资助。
收稿日期: 2020-09-24; 修改日期: 2021-01-29; 定稿日期: 2023-02-17

Key words PLC security; industrial control system security; cyber kill chain model; PLC attack technology; PLC defense technology

1 引言

工业控制系统(Industrial Control System, ICS)在现代电力、冶金、交通等领域发挥着重要作用。在工业 4.0 的新时代背景下, 工业控制网络将与传统信息网络深度融合, 实现由信息化向智能化迈进的重大飞跃。然而, 在全球工业控制系统快速发展的过程中, 针对工业控制系统的攻击事件频发, 给社会经济发展带来了重大损失。

近些年来, 针对 ICS 的网络攻击相较于传统信息安全领域认证绕过、盗取数等[1]攻击目的, 更专注于对物理设施的控制甚至破坏[2]。2010 年伊朗核电站遭受“Stuxnet”病毒[3]攻击, 内部浓缩铀工厂五分之一的离心机被摧毁, 导致伊朗核计划推迟数年。2017 年 Dragons 团队发现了针对 ICS 量身定制的恶意软件“TRITON”[4], 它能锁定施耐德电气 Triconex 安全仪表系统(SIS), 替换其控制逻辑, 导致目标无法正常运行。2017 年安全厂商 ESET 公布一款针对电力变电站系统进行攻击的恶意软件“Industroyer”[5], 支持四种工控协议, 可以直接控制断路器状态, 进而导致变电站断电, Dragos 推测其与 2016 年底持续半小时的乌克兰停电事件有关。2019 年美国可再生能源电力生产商 sPower 收到网络攻击[6], 造成了持续 10 小时的停电事故, 北美电力可靠性公司(NERC)表示, 攻击者是利用思科防火墙固件漏洞发起 DoS 攻击。

工控系统是由计算机与工业过程控制组件组成的自动控制系统, 如图 1 所示, 普渡模型[7]描述了工控系统各个层次中重要组件之间的主要相互依赖与互联关系。

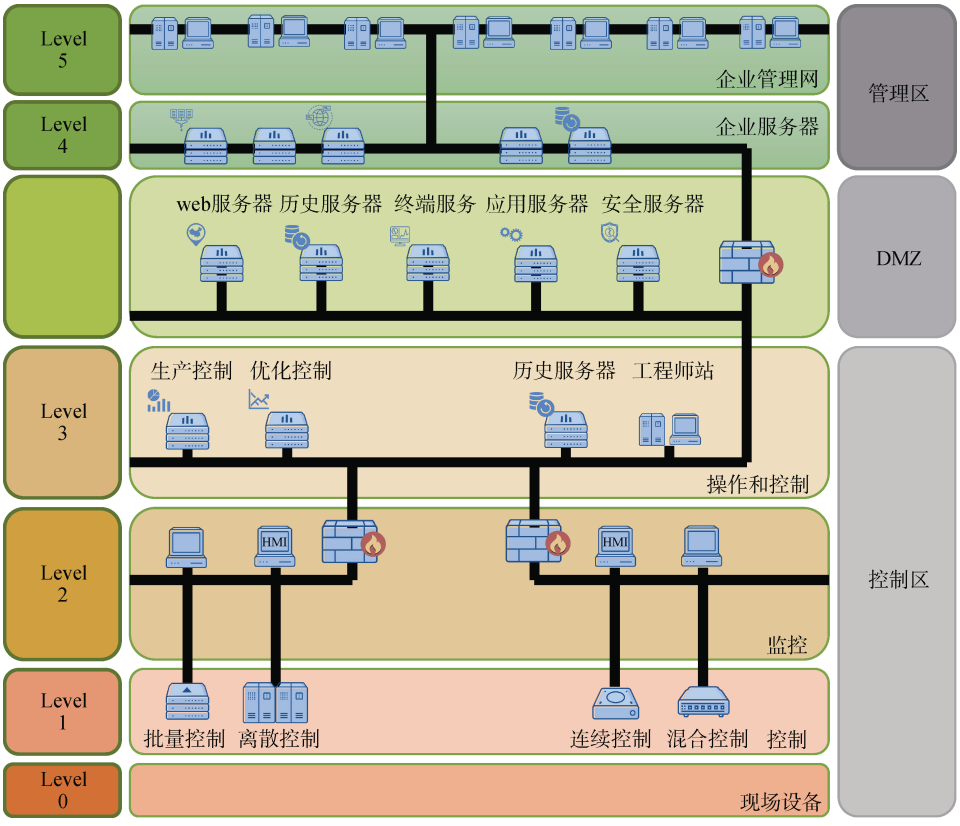


图 1 工控系统普渡参考模型

Figure 1 Purdue Reference Model of Industrial Control System

PLC 作为一种在工业环境下专用的逻辑控制器, 以其高可靠性、高扩展性、强大的逻辑控制能力、易于编程等特点, 已成为现代工控系统的核心组件。PLC 位于普渡模型控制区的控制层, 将工业控制过程数据实时返回给监控层的 SCADA(Supervisory Control And Data Acquisition)系统, 操作员可以通过 HMI(Human Machine Interface)查看实时过程事件和现场操作员级的实时画面, 并通过这些组件实现对

过程的自动控制。同时, PLC 按照一定控制逻辑对现场设备层物理设备如驱动器、阀门、电机等进行控制和数据采集, 以实现各种运动控制、工业过程监测与控制和工控数据处理等。

根据 Daniellep 等人[8]对 2010-2018 年间 ICS-CERT 988 项报告的数据统计, 近些年最容易受到攻击的工控组件为 HMI、SCADA、PLC。由于 PLC 是工控系统的核心组件, 其面临的安全形势也更加严峻。近些年对 PLC 攻防技术的相关研究也逐渐成为热门, 但由于不同 PLC 在系统架构和网络通信等方面的差异性, 多数研究人员往往将特定厂商及型号的 PLC 作为研究对象。Irfan Ahmed[9]从网络层和设备层对 PLC 攻击技术进行总结, 并提出一种针对 PLC 的取证技术。Abraham Serhane[10]从代码层和固件层对 PLC 攻击技术进行分类, 并提出相关建议如协议认证、访问控制、系统安全日志同步等提高 PLC 安全方法。徐震[11]等总结了 PLC 自身设计与运行过程中的脆弱性, 并从攻防两种角度对 PLC 安全技术进行了全面的总结与归纳。

随着工控系统攻防技术的演进, 针对 PLC 发起的攻击逐渐呈现出阶段性特征, 而现有研究采用的分类方法显示出两方面局限性:

(1) 忽视 PLC 设备之间的异构性, 针对性的防护方案结论往往借鉴意义有限。

(2) 未深入剖析阶段性特征, 难以有效对比分析相关工作以进行有效的防护。综上所述, 现有研究更多关注于 PLC 作为孤立的控制设备所存在的安全性问题, 而未从工控实际场景的角度进行分析。

本文将杀伤链模型应用于 PLC 攻击技术的分析中。一方面, 使用将针对特定类型 PLC 的攻击技术, 映射到杀伤链模型各个攻击阶段的分类方法, 从整体上降低了 PLC 异构性带来的影响; 另一方面, 通过对 PLC 攻击技术进行阶段性的划分, 提炼针对 PLC 攻击的主要方法与途径, 以深入洞察攻击者的行为模式, 并针对此类行为模式总结相关应对的防御技术。

本文的章节划分如下: 第 2 节简述 PLC 的基本架构和工作原理; 第 3 节详细阐述杀伤链模型的组成及应用; 第 4 节基于杀伤链模型对 PLC 攻击技术进行总结; 第 5 节总结针对 PLC 安全防护的方法, 以及 PLC 取证技术; 第 6 节结合以上分析对 PLC 安全研究提出展望。

2 PLC 基本架构和工作原理

PLC 是一种能够通过编程来进行实时控制的工业级计算机, 其模块化组件支持各种类型的生产环境, 具有使用灵活、响应迅速、操作简易、指令集丰富等[12]等特点, 已经成为现代工控系统的核心组件。随着 PLC 研发技术逐渐成熟, 未来 PLC 在高性能控制、连通性、安全通信、跨平台通用性等方面将会有显著提高[13]。

2.1 PLC 硬件架构

PLC 硬件架构主要包括电源模块、输入/输出模块、处理器模块[14], 使用工控上位机软件可以对 PLC 进行编程、组态与实时监控, 如图 2 所示:

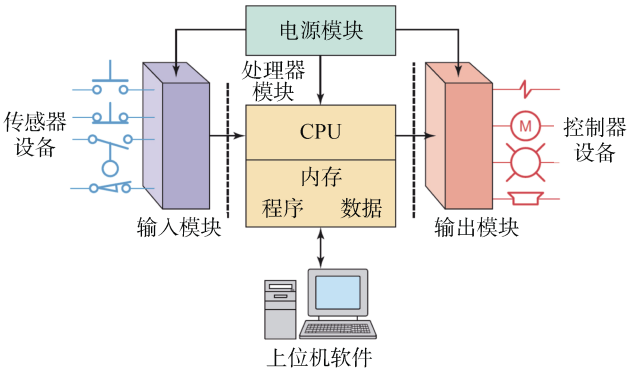


图 2 PLC 硬件架构

Figure 2 PLC Hardware Architecture

(1) 电源模块。PLC 内置稳压电源, 为背板上的其它模块提供直流电。一些 PLC 包含备用电池防止意外断电。在一些小型控制系统中 PLC 也为其他现场设备如传感器、小型控制器供电。

(2) 处理器模块。分为 CPU 和内存两部分。

CPU, 即中央处理器, 不同厂商使用的 CPU 种类有很大差异[15], 如 Schneider M340 系列 PLC 采用 ATMEL 芯片, Siemens S7-300 系列使用 Infineon 的芯片。CPU 功能主要包括内存管理、程序执行、I/O 控制等, 大型 PLC 如西门子 S7-400 通常会包含多个 CPU, 使用冗余技术提高安全性。

内存主要包括两种, 一种是只读存储器(ROM), 用于保存 PLC 的 Bootloader、固件和控制程序等, 断电后不会消失, 往往在厂商会将固件烧写到 ROM 中, 来执行用户编写的程序, 后续用户也可以根据需求更新固件; 另一种是随机访问存储器(RAM), 用于保存程序执行中的实时数据如输入输出数据、程序变量值等, 断电后会消失。

(3) 输入/输出模块。即 I/O 模块, 可以通过 I/O 点数判断 PLC 的型号。通常小于 256 点的为小型机, 点数介于 256~2048 的为中型机, 大于 2048 的为大型机。I/O 模块根据处理数据类型不同可分为数字量 I/O 和模拟量 I/O。I/O 模块会将实时采集的数据保存到 CPU 的输入/输出映像寄存器中, PLC 程序在运行过

程中会周期性访问寄存器内的数据。

2.2 PLC 软件架构

PLC 软件架构主要由固件、Runtime、用户程序三部分构成, 如图 3 所示:

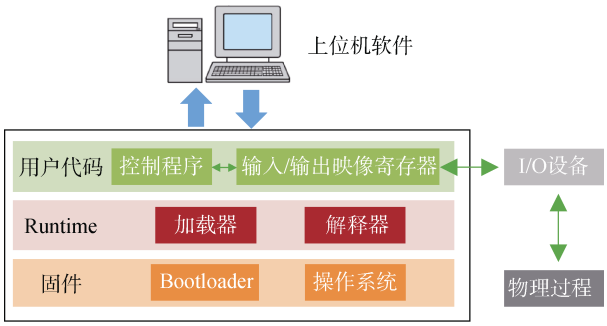


图 3 PLC 软件架构

Figure 3 PLC Software Architecture

(1) 固件。包括 Bootloader 和操作系统两部分。

PLC 中 Bootloader 除了引导操作系统启动, 通常还负责对固件进行校验, 如果固件未通过校验算法, PLC 会由于操作系统加载失败而出现故障。

PLC 使用的操作系统与传统操作系统有很大差异, 在功能上更偏重于提高实时性、可靠性、可扩展性。不同 PLC 厂商、不同的设备型号使用的操作系统通常也互不相同, 部分 PLC 操作系统如表 1 所示。

表 1 PLC 操作系统

Table 1 Operating Systems of PLCs

厂商	系列	操作系统
Siemens	S7-1200 V4	ADONIS
Siemens	SIMATIC WinAC	Windows
Schneider	Quantum	Vxworks
Allen Bradley	ControlLogix	Vxworks
Allen Bradley	PLC5	Microware OS-9
WAGO	PFC200	Linux
Yokogawa	FA-M3	Linux
GE	VersaMax	Vxworks

(2) Runtime。PLC Runtime 是运行在 PLC 操作系统上的一个进程, 其核心功能是执行 PLC 的控制程序。工控软件将用户程序编译成二进制代码, 并下载到 PLC 中由 Runtime 执行。

二进制代码执行过程通常分为两种, 一种是加载型, 即直接加载执行二进制代码, 如 WAGO PFC200 PLC, CODESYS IDE 会将用户程序编译成 ARM 汇编指令[16], 直接被加载执行; 另一种是解释型, 即先将代码解释成机器语言再执行, 如 Siemen TIA Portal 会将用户程序编译成 MC7 Code, 再由 Siemens PLC 解释执行[17]。

PLC 厂商通常会使用自己的 Runtime, 但随着德国 3S 公司推出 CODESYS[18], 很多 PLC 厂商选择 CODESYS 作为其软件平台, 如 ABB、Bachmann、IFM、HOLLYSYS、和利时等。CODESYS 架构包括开发层、通讯层和设备硬件层, 它不仅支持 CANopen、Profibus 和 EtherCAT 等多种现场总线, 而且可根据客户的具体需求将不同自动化厂商提供的产品和系统进行组合配置后统一编程, 从而真正实现了控制系统的开放性和可重构性。

(3) 用户程序。用户使用编程软件来编写控制程序。如图 4 所示, IEC 61131-3 规定了 5 种 PLC 编程语言:

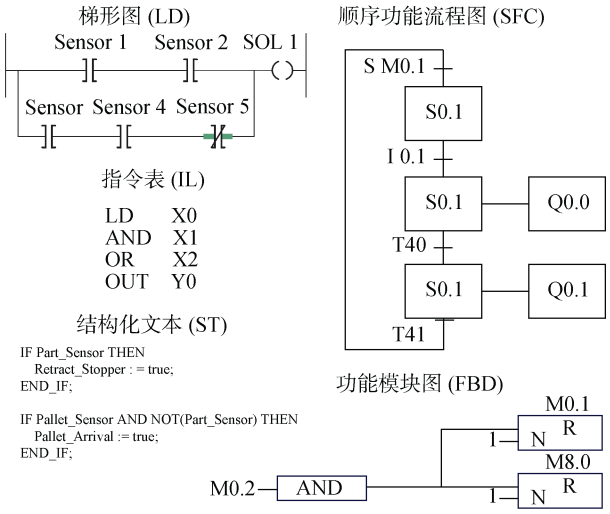


图 4 PLC 编程语言

Figure 4 PLC Programming Languages

PLC 编程实现与传统编程存在一些差异, PLC 程序总体上是周期性执行的, 周期一般为毫秒级; PLC 程序规模比较小, 单个周期内程序按顺序执行, 可靠性高; PLC 主要进行逻辑和运动控制, 不易实现复杂的算法。

2.3 PLC 工作原理

PLC 基本工作原理如图 5 所示:

(1) 启动扫描

PLC 在冷启动或热启动后, 首先会执行启动扫描, 来对一些系统变量进行初始化。如在 Siemens S7-300/400 系列 PLC 中, OB100、OB101 和 OB102 是用于启动扫描的组织块[19]。

(2) 系统诊断

PLC 会对电源、运行状态、输入输出模块等进行故障诊断, 若出现故障 CPU 会切换到 SF(System Fault)状态。

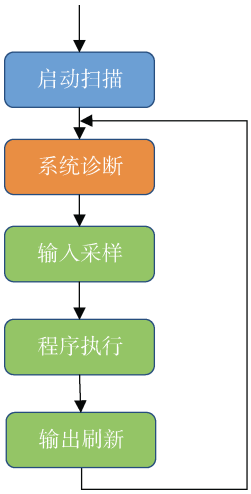


图 5 PLC 工作原理

Figure 5 PLC Principle of Operation

(3) 输入采样

对于数字信号, PLC 读取外部输入传感器信号, 并存入输入映像寄存器中; 而对于模拟信号, PLC 直接从外部模拟量传感器中进行读取。

(4) 程序执行

PLC 从输入映像寄存器中读入对应输入变量值, 按顺序执行控制程序, 并将程序执行结果输出到输出映像寄存器。

循环时间是指 CPU 在 RUN 模式下执行一个循环阶段所需的时间。每当 PLC 开始新的扫描周期时, CPU 会重启看门狗定时器。若循环时间超过看门狗定时器的预设值, 且控制程序不包括时间错误中断, PLC 就会产生时间错误事件, 并写入诊断缓冲区。

(5) 输出刷新

对于数字信号, PLC 将存放在输出映像寄存器区的程序执行结果, 刷新到数字量的外部输出模块中; 而对于模拟信号, PLC 直接将程序执行结果刷新到外部硬件模块中。

2.4 PLC 通信协议

PLC 通信协议属于工控协议, 而工控协议是工控设备与上位机、设备与设备之间进行信息传递的一种重要媒介。

从 OSI 参考模型角度分析, 大多数 PLC 通信协议都属于应用层协议, 下面列举几个常见的 PLC 通信协议, 并简要分析其功能特性和脆弱性。

(1) Modbus

Modbus[20]是 Modicon 公司在 1979 年为使用 PLC 通信而发明的, 已经成为工业领域通信协议的业界标准, 广泛应用于各类工业领域。

Modbus 协议可分为 Modbus-ASCII、Modbus-RTU、Modbus-TCP 三种, 前两种为串行通信协议, Modbus-TCP 是为让 Modbus 数据在以太网上能顺利传输而产生的, 是一种基于 TCP 的应用层协议, 使用 TCP 502 端口。

Modbus 不管采用何种方式传输, 功能码都是相同的, 通常可分为公共功能码和私有功能码, 公共功能码表示线圈和寄存器的读写、文件记录读写、系统诊断等等, 私有功能码是自定义功能码, 通常表示与 PLC 系统配置相关的操作, 如 PLC 启停控制。

Modbus 协议的脆弱性主要表现在缺乏认证, 而无法保证命令来自合法用户; 缺乏加密, 协议明文传输导致协议数据很容易被监听、篡改; 私有功能码不安全, 由于私有功能码由用户定义, 易出现协议栈漏洞。根据思科 Talos 团队的漏洞报告, Schneider Electric Modicon M580[21]私有功能码存在大量漏洞。

(2) S7Comm

S7Comm 协议是西门子的专有协议, 是西门子 S7 通信协议簇中的一种, 适用于 S7-300、S7-400 系列 PLC。

S7Comm 协议被封装在 TPKT 和 ISO-COTP 协议中, 这使得 S7Comm 协议数据单元能够通过 TCP 传送, 其使用 TCP 102 端口。

S7Comm 功能码分为一级功能码和二级功能码[22]。一级功能码包括通信建立、设备启停、变量读写、程序上传与下载等, 二级功能码是在主级功能码(CPU Function)下的功能码组, 涵盖设备识别、系统诊断、密码认证等系统功能。

S7Comm 协议的脆弱性主要表现在仅部分功能支持权限认证; 口令认证算法过于简单, 流量中的加密口令容易破解; 缺乏加密, 协议明文传输导致协议数据很容易被监听、篡改, 发起中间人攻击。

(3) DNP3

全称是分布式网络协议 3(Distributed Network Protocol 3), 由加拿大能源控制系统公司 Harris 于 1993 年提出。主要用于数据采集系统和远程设备之间的通信, 特别是电力系统中 SCADA 控制系统和远程变电站之间的通信。

DNP3 协议可通过 TCP/UDP 进行封装, 在以太网上传输。DNP3 协议通常采用主站/从站(Master/Slave)的配置模式, 如在电力系统中 SCADA 控制系统为主站, 远程变电站智能设备如 RTU 为从站, 从站设备默认开启 TCP 20000 端口进行通信。

DNP3 协议功能码主要包括数字量与模拟量数据的读写、文件目录获取、文件读写、应用程序控制和系统状态获取等。

DNP3 协议的脆弱性主要表现在缺乏认证, 主从站地址没有认证信息, 不能确定发送方的合法性; 缺乏加密, 协议明文传输导致协议数据很容易被监听、篡改; 缺乏完整性保护, 虽然校验和机制可以保证数据传输正确, 但机制相对简单, 难以保证数据完整性。

3 杀伤链模型

网络杀伤链模型(Cyber Kill Chain Model)[23]是洛克希德-马丁公司基于美国国防部的杀伤链模型, 提出的一种针对入侵行为的新型杀伤链。使用杀伤链模型来描述入侵过程, 将攻击链映射到各个防御阶段上, 并将单次入侵行为与广泛战役行动联系起来, 以构建基于情报驱动的现代计算机网络防御体系。网络杀伤链模型广泛应用于攻防能力评估、应急响应、APT 攻击追踪等领域。

3.1 攻击阶段

如图 6 所示, 杀伤链模型包含以下 7 个阶段:

(1) 侦察识别: 攻击者通过调研现有研究、开源情报与工具, 收集目标系统信息, 来寻找目标系统的脆弱点。此外, 也可以进行主动侦察, 通过向目标发送探测脚本, 确定其软件或系统版本。

(2) 武器构建: 将恶意载荷嵌入到合法载体中, 如 PDF 文件, 为后续的漏洞利用创造环境。武器构建并不是必须的步骤, 通过凭据入侵到某些网络后, 可以直接对目标系统进行漏洞利用。

(3) 载荷投递: 攻击者将恶意载荷投放到目标系统的过程, 通常需要首先获取对目标网络或系统的访问权限, 攻击者往往利用目标系统存在的认证漏洞来实现。

(4) 漏洞利用: 目标系统执行攻击者投递的恶意载荷, 其往往利用目标系统存在的软件或系统漏洞, 也可以使用系统默认支持的函数或功能, 来实现恶意行为。

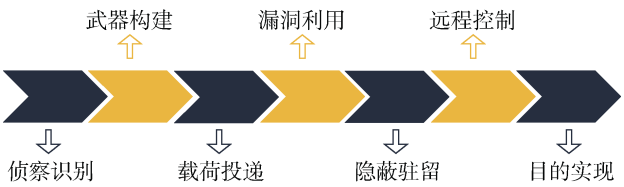


图 6 网络杀伤链模型

Figure 6 Cyber Kill Chain Model

(5) 隐蔽驻留: 恶意载荷在目标系统会通过修改系统配置或功能来隐蔽自身活动, 并通过安装自启动服务、木马后门等持续运行的服务, 进而长期驻留在目标系统上。

(6) 远程控制: 一旦恶意载荷在目标系统上开启了远程服务, 攻击者就可以远程向目标系统发送任意控制命令。

(7) 目的实现: 此时, 攻击者可以通过远程控制实现其原始目的, 如篡改系统参数、下载额外恶意组件或让目标系统成为僵尸节点等。

3.2 模型演进

杀伤链模型在实际应用的过程中也在不断演进。Sean T[24]拓展了杀伤链模型的攻击阶段, 引入了“内部杀伤链”和“目标篡改杀伤链”, 相比原始模型对攻击的描述更加全面。Verdasys 公司[25]提出了由计划、恶意代码引入、命令与控制、扩张等阶段组成的杀伤链模型, 其独特之处在于丰富了识别阶段的功能, 并设置攻击后及时撤离的策略。Hyeob Kim[26]等人认为在攻击过程中可能出现多个侦察识别、武器构建、漏洞利用等构成的攻击生命周期, 并提出将杀伤链模型中的威胁描述划分为外部威胁和内部威胁, 针对内外部威胁采用不同的防御手段与策略。

3.3 模型应用

在杀伤链模型中, 指示符用于客观描述入侵发生的信息, 是构成威胁情报的基本元素。当用户在系统中观察到一个指示符时, 说明系统很可能正在或将要被攻击, 如黑名单中的 IP 地址、C&C 服务器域名等。此外, 这些指示符可用于重构攻击者的入侵行为, 以及对相同攻击者在很长时间内对同一目标攻击的有效检测。

3.3.1 重构入侵

重构入侵, 即在某一阶段检测到入侵行为后, 继续分析此阶段前期或后期其他阶段的行动, 将整个杀伤链模型重构出来, 分为重构前期或后期入侵。

假如攻击者通过邮件附件的方式发送一个恶意 PDF 文件, 该文件中包含一个漏洞利用 shellcode 和一个木马安装程序; 利用一个 PDF 漏洞安装并执行木马程序, 并与外界 C&C 服务器建立通信。防御者检测到主机与外界有异常通信, 在攻击的 C&C 阶段检测到入侵。此时防御者可以基于杀伤链模型重构攻击者的入侵行为。

重构前期入侵指当攻击者已经完成入侵的前期阶段, 防御者检测到入侵后, 也需要基于杀伤链模型对此进行还原。攻击方考虑经济性, 会重用工具和方法, 防守方基于对入侵的还原, 在下一次入侵发生前期就可以展开相应的防御。如根据上述场景, 此时防御者沿着杀伤链模型向前分析可以得到恶意软

件的名称、路径; 恶意邮件发件人邮箱; 恶意 PDF 使用何种加密算法、密钥等等信息。

重构后期入侵指当防御者检测到入侵后, 即使已经将其拦截, 也要模拟合成入侵的后续行动, 以理解攻击者的战术目, 以及系统的脆弱性, 更好地制定针对性的防御策略。如根据上述场景, 防御者可从载荷投递阶段沿着杀伤链向后分析, 得到攻击使用的漏洞编号; 木马程序的名称、路径; 攻击者 C&C 服务器的域名; 攻击欲窃取的数据等。

3.3.2 战役分析

同一个攻击者在很长一段时间内, 对一个目标进行多次攻击, 这一系列攻击称之为战役。战役分析是对多条杀伤链进行横向对比, 识别多条杀伤链相同、重叠的指示符, 以分析杀伤链之间的相关性。可用于确定入侵者的行为模式、战术、策略以及能力, 了解攻击者意图, 以制定针对性的防御策略。

假如攻击者第二次发出类似的恶意邮件, 虽然攻击者做了改变, 但分析人员发现两次邮件的发件人一样。PDF 使用相同的加密算法和密钥等。经过分析, 发现两次行动之间的相似性, 将其拦截。此过程放映了防御者对两次入侵活动各个阶段指示符的横向对比。此外, 新的指示符可用于后续的检测, 提升了防御能力。

3.3.3 应用实例

相比于 ATT&CK、CAPEC 等用于全面描述攻击技术的公共分类模型或知识库, 杀伤链模型是一种更加抽象的高层次模型, 缺少相应的技术实现细节。研究人员需要结合杀伤链模型分析各类攻击, 并在具体技术场景中对其进行阐述, 系统地描述攻击者的行为[27]。

HosseiniNejad 等人[28]基于杀伤链模型分析远程控制木马的层次特征。Dargahi 等人[29]结合杀伤链模型分析恶意软件加密特征, 并提出一种对应于杀伤链模型的防御行为模型。Bahrami 等人[30]使用杀伤链模型分析 APT(Advanced Persistent Threat)攻击, 详细总结了 APT 相关策略、技术和过程, 并映射到杀伤链模型攻击阶段上。Duncan 等人[31]结合攻击树模型与杀伤链模型检测云计算过程中的恶意行为, 具体而言, 在杀伤链的基础上叠加攻击树, 可以进一步促进更高层次的间接检测机会, 同时允许提供商确定发现可疑活动时的攻击阶段, 提高云服务系统的应急响应能力。

4 基于杀伤链模型的 PLC 攻击技术分析

本节将近些年各类 PLC 攻击技术进行总结分类, 并映射到杀伤链模型的各个阶段中, 如表 2 所示, 进而从多个层次阐述各类 PLC 攻击技术的工作原理、特点以及局限性等。

4.1 侦察识别

目前对 PLC 的侦察识别获取的信息主要包括以下两类。

一类是设备类型、固件版本等硬件设备信息, 往往与设备漏洞直接相关, 通常可以利用公共资源获取, 包括 Shadon、Nmap[50]、Metasploit[51]等都包含对部分 PLC 的侦察识别脚本, 一些针对特定协议的客户端工具如 pymodbus[52]、snap7[53]等也提供侦察识别功能。这些脚本通常首先确认目标设备开放的端口号, 然后将其映射到对应的工控协议常见的工控协议与其使用的端口号映射关系如表 3 所示, 最后发送与工控协议对应的探测脚本, 来识别目标 PLC 的厂商名称、设备型号或固件版本。

表 2 杀伤力模型攻击阶段-PLC 攻击技术映射关系

Table 2 Mapping between Cyber Kill Chain Attack Phases and PLC Attack Techniques

攻击阶段学术研究	侦察识别	武器构建	载荷投递	漏洞利用	隐蔽驻留	远程控制	目的实现
Anastasis Keliris[16]	●	●	●				●
Ali Abbasi[17]		●	●		●		
Sushma Kalle[32]	●	●	●	●			
Stephen McLaughlin[33]		●	●				
Stephen McLaughlin[34]		●	●				
Naman Govil[35]		●	●				
Abraham Serhane[36]		●	●				
Carl Schuett[37]		●	●		●	●	
Ali Abbasi[38]		●	●		●	●	
Luis Garcia[39]		●	●		●	●	●
Matthias Niedermaier[40]			●				●
Thomas Weber[41]			●				

续表

攻击阶段学术研究	侦察识别	武器构建	载荷投递	漏洞利用	隐蔽驻留	远程控制	目的实现
Haroon Wardak[42]				●			
Ryan Grandgenett[43]			●	●			
Cheng Lei[44]				●			
Eli Biham[45]		●	●	●			
Johannes Klick[46]		●	●	●		●	
Ralf Spenneberg[47]		●	●	●		●	
Zachary H. Basnight[48]		●	●	●			●
Saranyan Senthivel[49]		●	●				●

表 3 常见工控协议端口号

Table 3 Port Numbers of Common Industrial Protocols

协议名称	传输层协议	端口号
Modbus	TCP	502
S7Comm	TCP	102
S7CommPlus	TCP	102
DNP3	TCP	20000
IEC 104	TCP	2404
CIP	TCP	44818
FINS	TCP	9600
GE SRTP	TCP	18245
MELSEC-Q	TCP	5006
	UDP	5007
PCWorx	TCP	1962
OPC UA	TCP	4840
Codesys2	TCP	2455
Bacnet	UDP	47808
EtherCAT	UDP	34980

而针对使用的私有协议 PLC 进行探测, 往往需要手动分析捕获的网络流量来构建探测脚本。一般而言, PLC 在返回硬件设备信息时有两种方式, 一种是直接返回可见的设备描述字符串, 如返回 6ES7 1214-1BG31-0XB0; V3.0, 表示西门子 S7-1214 PLC, 固件版本为 V3.0。另一种仅返回一个标识符, 其与特定型号的设备有默认的对应关系, 需要进行推测, 如采用对多种相同类型、具体型号不同的 PLC 进行探测的方法。

另一类是获取 PLC 的运行状态、控制程序等软件配置信息。运行状态通常为 PLC 的启动或停止状态。控制程序信息对于攻击者是相对重要的, 通过分析 PLC 控制程序可以获取其使用的 I/O 点位信息、控制逻辑等等。在构建此类探测脚本时, 即使工控协议中定义了数据读写、设备启停、控制程序上传下载等特定功能码, 由于获取此类信息往往包含复杂的会话序列, 各字段的依赖关系相对复杂, 较难实现完整的通信流程, 而私有协议格式和字段语义的未知更增加了实现的难度。

目前针对 PLC 侦察识别行为的防御技术主要使用工业防火墙和入侵检测系统(Intrusion Detection System, IDS)。Manuel Cheminod[54]等人针对 Modbus/TCP 协议提出一种应用层协议过滤技术, 其在保证高准确率、低丢包率的同时, 有效降低了延迟。R Dheeraj[55]等人构建了一个监测工控系统异常操作的 SCADA 防火墙, 通过深度解析数据包提取特征, 并使用机器学习的方法动态更新防火墙过滤规则。综上所述, 这种基于网络隔离的防护手段对部分探测脚本可以起到屏蔽作用, 但由于工业防火墙对吞吐量的要求, 且依赖于对工控协议的深度解析, 难以对复杂攻击进行实时分析。

4.2 武器构建

从 PLC 的软件架构层次角度分析, 针对 PLC 的攻击载荷可分为两类。一类是构建恶意 PLC 控制程序; 另一类是构建恶意 PLC 固件。

4.2.1 控制程序

PLC 控制程序的分析、构建与度量有别于传统的 Windows、Linux 系统程序[56], 攻击者可以使用 PLC 编程软件支持的系统函数, 构建任意 PLC 恶意程序。然而在实际工控系统环境中, PLC 通常用于控制实际生产过程, 其 I/O 地址与真实物理设备相对应, 编写恶意程序需要使用原始程序中已定义的 I/O 地址和程序变量, 才能对真实物理场景产生影响。因此, 构建恶意 PLC 程序分成两个阶段, 首先需要逆向分析 PLC 的原始程序, 然后再基于上述信息自动化构建 PLC 恶意程序。

对于编程软件编译后得到的原始二进制程序, 需要准确逆向分析得到原始程序, 并提取相关变量地址, 分析程序功能。

Sushma Kalle 等人[32]开发了一个针对施耐德 RX630 微控制器指令集的反编译器, Eupheus。它能将 RX630 机器码转换为 PLC 的 IL 指令, 如图 7 所示。

IL	Hex	Assembly Language
Rung 0		
LD %I0.1	7c 1c	BTST 1(imm), R12
AND %I0.8	23 04	BCnd.B 4(pcdsp) #cd: BNC(C==0)
	7c 8c	BTST 8(imm), R12
ST %M1	fc e6 72 00 00	BMCnd 1(imm), [R7].B #cd: BMC(C==1) #dsp: 0x0000
Rung 1		
LD %M307	f6 73 26 00	BTST 3(imm), [R7].B #dsp: 0x0026
ST %M498	fc ea 72 3e 00	BMCnd 2(imm), [R7].B #cd: BNC(C==1) #dsp: 0x003e
	02	RTS

图 7　IL 指令及其对应的 RX630 机器码

Figure 7　IL Instructions and Corresponding RX630 Machine Codes

从逆向分析得到的 IL 指令中可得到相关变量地址以及程序功能信息。此外，作者还开发了一个基于规则的恶意载荷编译器，能够根据给定 IL 指令和自定义规则自动生成可直接在 PLC 运行的恶意载荷。此外，此工作不仅局限于文章中的实验设备 Schneider Electric Modicon M221，而可以拓展到任何能够执行使用 SoMachine-Basic 编译程序的 PLC 或使用 RX630 系列微控制器的 PLC。

Anastasis Keliris 等人[16]开发了一个针对 CODESYS v2.3 套件编译出的 PRG 文件的逆向框架，ICSREF。作者使用 CODESYS v2.3 编译得到 PRG 文件，并以 WAGO 750-881 PLC 为例，作者通过手动分析得到 PRG 二进制文件的一般结构，如图 8 所示，其中蓝色的部分表示 CODESYS IDE 编译后得到的 ARM 指

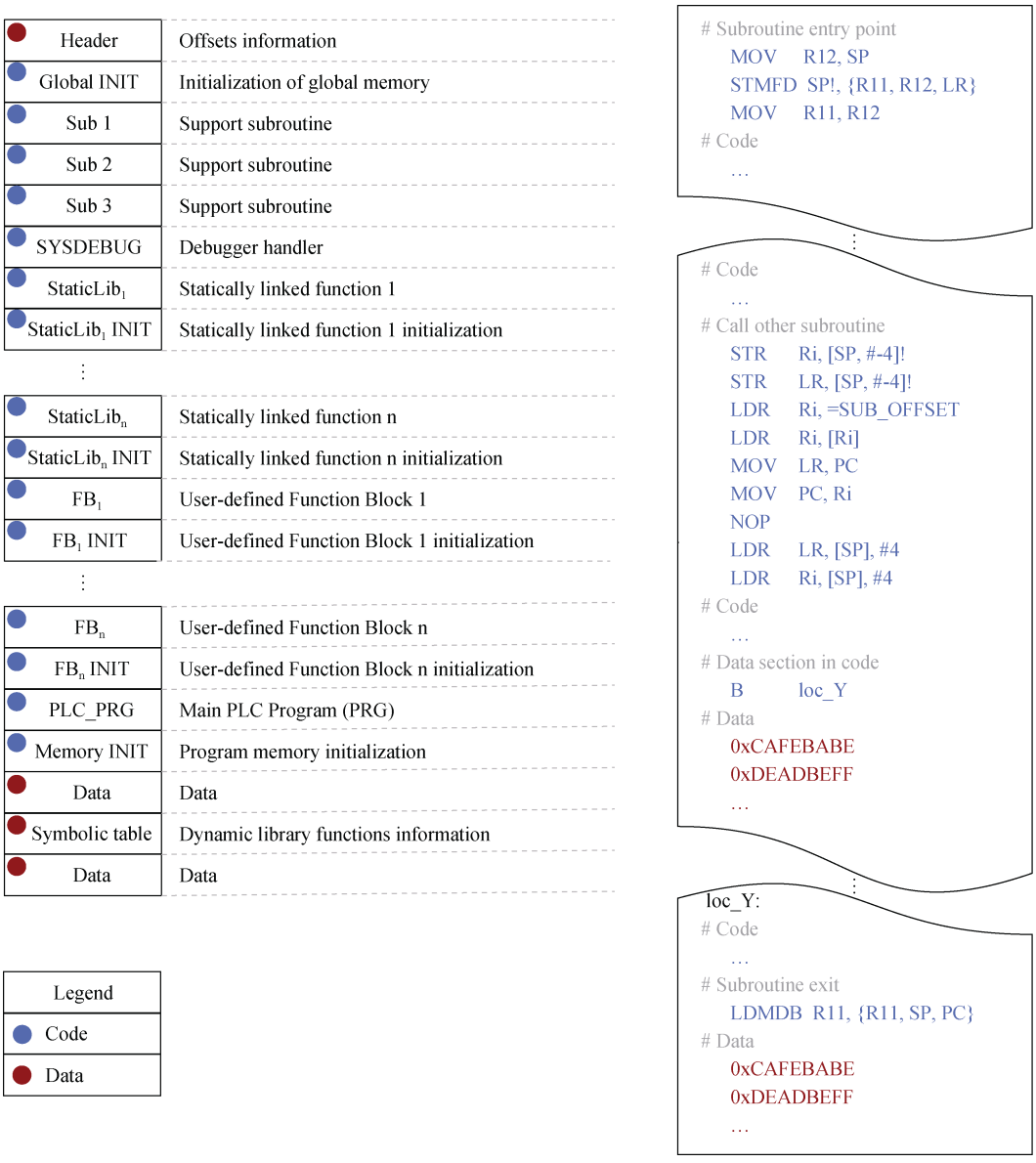


图 8　PRG 二进制文件格式

Figure 8　PRG Binary Format

令。基于 PRG 文件结构 ICSREF 会构建一个函数指纹库, 用于识别 ARM 程序中的函数, 以及一个 I/O 信息库, 用于保存 I/O 标签; 然后借助 radare2 工具反编译上述 ARM 机器码, 得到图 8 的 ARM 指令; 最后结合函数调用、变量类型等信息构建 PLC 程序控制流程图。

对于自动化构建 PLC 恶意程序, 现有研究一方面集中在基于原始程序的程序篡改方法, 另一方面尝试使用相关的 PLC 程序指令构建恶意程序。

Stephen McLaughlin 等人[33]提出一种针对特定 PLC 控制程序的异常数据注入攻击(False Data Injection, FDI), 通过使传感器输入数据异常, 误导控制算法输出异常, 进而对实际物理场景产生不良影响。传统的 FDI 攻击需要攻击者详细了解物理场景, 而作者提出的方案仅依赖于 PLC 控制程序。作者首先将 PLC 控制程序转换为线性时序逻辑 ϕ, 再提取有限状态机 M, 最后对 M 进行对抗性分析来获得恶意输入。

Stephen McLaughlin 等人[34]设计了一种自动生成 PLC 恶意程序的工具, SABOT。它首先逆向分析 PLC 原始程序, 基于程序中变量间的控制依赖关系, 构建图 10 所示的 NuSMV 模型。然后将其与目标控制系统的行为模型相结合, 自动推断出 PLC 程序变量与受 PLC 控制设备如传感器、控制器变量的映射关系。最后基于这种映射关系自动生成恶意程序, 进而有效影响目标控制系统行为。

Constraint	NuSMV Model M
input x	VAR x: boolean; ASSIGN init(x) := $\perp$; next(x) := {$\top$, $\perp$};
output or local y $c = y \leftarrow \alpha$	VAR y: boolean; ASSIGN init(y) := $\perp$; next(y) := α;
timer t $c = t \leftarrow \alpha$	VAR t: Boolean, t_p: boolean; ASSIGN init(t) := $\perp$; next(t) := $\alpha \wedge (t_p \vee t)$? $\top$: $\perp$; init(t_p) := $\perp$; next(t_p) := α;

图 9 PLC 控制依赖与 NuSMV 模型

Figure 9 Constraint and Corresponding NuSMV Model

Naman Govil 等人[35]开发了一个恶意梯形图程序炸弹, 能够直接插入到原始 PLC 程序中, 改变其行为或者等待后续触发来激活恶意行为。实验基于 SwaT 水处理系统仿真平台, 提出 3 种攻击方式:

1) 替换 ADD 指令发起 DoS 攻击, 在恶意 ADD 指令中加入一个无限循环, 使 PLC 循环时间超出最大值。

2) 如图 10 所示, 攻击者通过在每个程序段末端修改输出数据, 修改 HMI 发送到 RIO 的数据, 以及从 RIO 读入的数据。

3) 使用 FFL 指令块记录数据到 SD 卡中。

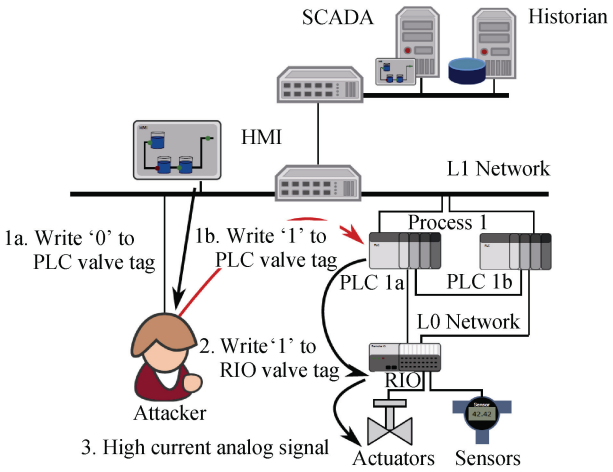


图 10 PLC 篡改 HMI 读写 RIO 设备数据

Figure 10 Manipulating Sensor Readings from RIO to HMI and Instructions from HMI to RIO

Abraham Serhane 等人[36]总结了 PLC 控制程序层相关漏洞, 大多是从编程的角度分析易出现的错误, 利用这些错误, 攻击者可以构造恶意载荷。如图 11 所示的条件竞争, 假定 X1 始终为 True, 当计时器 tmr1 计时到预设定值时, tmr1.DN 置为 True, 则 tmr1 失效, tmr1.DN 又置为 False 根据此程序 tmr1 会重新开始计时, 这将导致 Valve01 永远不会置为 False。

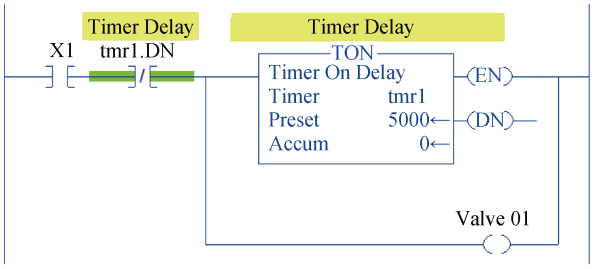


图 11 PLC 程序中的条件竞争

Figure 11 Racing Condition in PLC Logic

4.2.2 固件

PLC 固件相比传统的嵌入式设备固件通常规模更大, 搭载私有操作系统, 在功能上对实时性实现的要求会更高。针对 PLC 恶意固件的构建更专注于 PLC 的控制功能部分, 如 I/O 控制、协议栈处理等。

PLC 固件通常是编译好的系统程序, 因此首要

的任务是提取固件中的文件系统和重要代码, 通过逆向分析二进制固件, 来定位关键代码的位置, 最后植入恶意代码, 来实现对 PLC 固件的攻击。

Carl D. Schuett 等人[37]通过篡改 Allen Bradley Controllogix 1756-L61 PLC 的固件, 绕过固件校验算法将其下载到设备中, 最终达到可以远程触发 PLC 停止运行的目的, 总体方案如图 12 所示。作者首先使用 IDA PRO 逆向分析固件, 并手动恢复系统符号表信息, 得出固件信息如头部字段含义。然后通过 JTAG 调试的方式, 定位到系统模式诊断、CIP 对象管理等关键代码, 并在此处加入恶意控制代码, 最终实现对固件代码的篡改攻击。

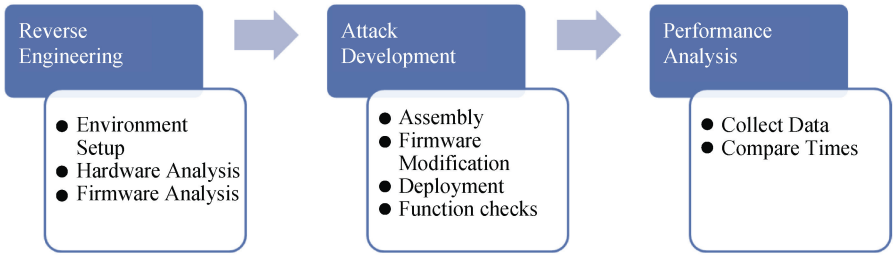


图 12 固件篡改攻击一般过程

Figure 12 General Process of Firmware Modification Attack

Ali Abbasi 等人[38]提出一种针对 PLC 引脚控制的新型 Rootkit, 可以在不被检测的前提下, 操纵或破坏 PLC 控制的物理过程。作者使用 Wago 750-820 PLC, 借助系统调试功能或驱动程序来访问和配置引脚, 对 PLC 的读写过程进行系统层面的 Hook, 总体步骤如图 13 所示。白色部分是原始控制程序的执行过程, 蓝色部分是引脚控制攻击执行的控制程序。由于被攻击后引脚输入输出模式在读写过程中被修改, 从 PLC 读取的值会被 Rootkit 篡改, 而向 PLC 写入值会失败。

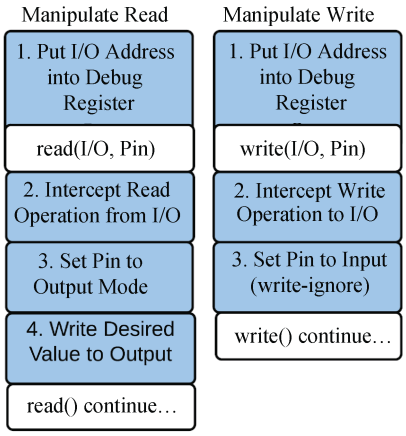


图 13 PLC 引脚控制攻击步骤

Figure 13 Steps of the PLC Pin Control Attack

Luis Garcia 等人[39]提出一种针对电网系统进行攻击的 PLC Rootkit, HARVEY, 能够在固件层对 PLC 输入输出命令进行篡改, 从而破坏物理电力设备。作者在 Allen Bradley 1769-L18ER-BB1B PLC 上实现了 HARVEY 原型。如图 14 所示, 作者借助 LM3S2793 芯片地址空间信息, 使用 IDA Pro 定位输入函数 UpdateMemoryFromGPIO 地址, 并通过 JTAG 调试接口在固件末尾处插入恶意数据篡改指令, 当设备处理输入数据之前会跳转到此指令处执行, 进而使 PLC 控制的外围设备出现异常。

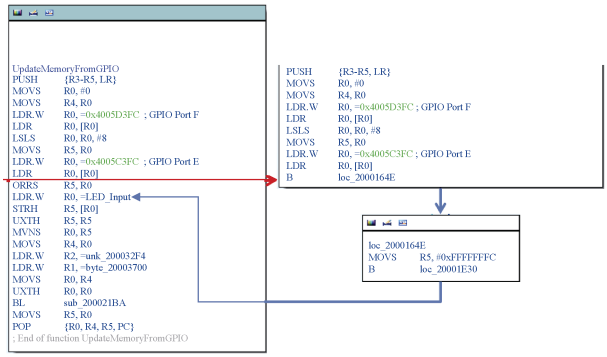


图 14 输入函数篡改实现

Figure 14 Implementation of Modifying Input Function

作者不仅实现 HARVEY 原型, 也在真实的电网系统上对其进行了评估, 证明了 HARVEY 在实践中的部署可行性。

4.3 载荷投递

将载荷投递到 PLC 一般可分为三种途径: 网络传输、串口传输、硬件烧写。

4.3.1 网络传输

由于工控协议大部分通过以太网传输, 只要网络可达, 攻击者无论是向 PLC 发送控制命令, 还是修改 PLC 的控制程序和固件, 都可以借助 PLC 通信使用的工控协议来实现。

事实上, 频繁向 PLC 发送探测报文对一些类型的 PLC 正常运行会产生影响。Matthias Niedermaier 等人[40]对 6 个厂商的 16 种类型的 PLC 进行探测数据包泛洪测试, 发现不同 PLC 对 TCP 泛洪攻击的承受能力有很大差别, 如图 15 所示。

作者将对 PLC 产生的影响分为六类: PLC 停止(Wago 750-831 PLC)、循环时间均值高度偏离(Wago 750-889)、循环时间均值中度偏离(Schneider TM221CE16T)、循环时间波动增加(Siemens S7-314)、循环时间加快(Phoenix ILC151)、无显著影响(其他设备)。此外, 从 CPU 负载的角度看, 多数基于 RTOS 系统的设备 CPU 负载波动很小, 而基于 Linux 系统的 Wago 750-8100 由于使用软中断处理网络数据, 其正常控制执行被迫停止。

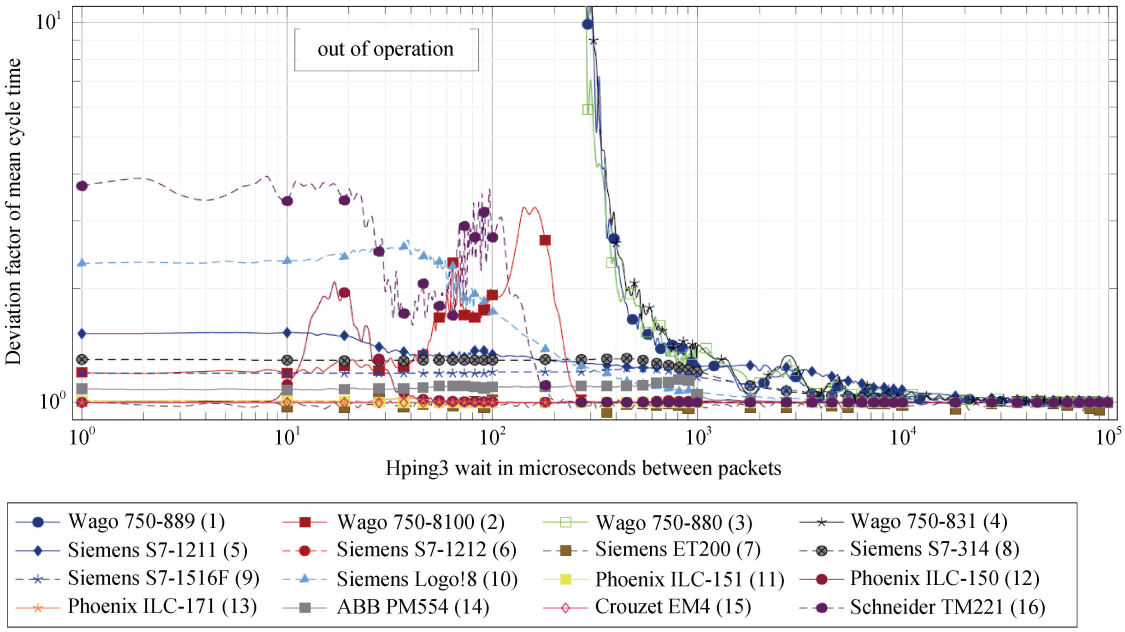


图 15 不同型号 PLC 循环时间均值偏离

Figure 15 Deviation of Mean Cycle Time on PLCs

4.3.2 串口传输

部分 PLC 支持 RS 232/485 串口通信。实现数据读写、强制复位/置位、程序上传/下载、系统诊断等功能。相比之下 RS-232 接口数据传输速率低, 传输距离有限, 抗干扰能力差。可以通过串口向 PLC 发送控制命令或传输控制程序和固件。

对串口的配置主要包括波特率(度量通信速度), 数据位(衡量通信中实际数据位的参数), 停止位(表示单个包的最后一位), 奇偶校验位(用于校验)。

常用的 PLC 串行协议如 Modbus, 包括 Modbus-RTU 与 Modbus-ASCII 两种。相比之下 Modbus-ASCII 具有开始和结束标记, 简单直观, 易于调试, 但由于传输的都是 ASCII 字符, 传输效率低, 二者的校验方式也有不同。

4.3.3 硬件烧写

嵌入式设备在测试阶段往往可以通过 UART、JTAG 接口进行调试, 如果这些端口被保留, 攻击者就可以利用这些端口对嵌入式设备系统进行攻击[57-59]。而针对 PLC 也可以使用的硬件攻击手段, 寻找到调试接口就可以对 PLC 固件进行实时调试、内存篡改甚至固件替换。

Thomas Weber 等人[41]使用 PCB 逆向工程的方法分析了西门子 S7-1211C 主板的 JTAG 调试接口, 并进行固件提取, 最终实现了对固件的调试功能。在此基础上, Ali Abbasi 等人[17]提取并分析西门子 S7-1200 PLC Bootloader 中的部分代码, 发现了一些特权功能。当被用于在 PLC 启动时通过串口发送特定命令即可触发, 这些特权包括内存读写、命令执行等。最后作者在 PLC 上实现了任意代码执行。

4.4 漏洞利用

PLC 的漏洞利用可以从两方面分析, 一方面是 PLC 网络通信层存在漏洞, 另一方面是 PLC 设备层存在可以被攻击者利用的系统功能。

4.4.1 网络层

根据 CVE、ICS-CERT 等批露的报告显示, PLC 网络通信协议设计上存在多种类型的漏洞, 下表总结了近些年 PLC 通信协议相关漏洞类型, 并列举了部分漏洞。

因此, 很多厂商为保证 PLC 通信过程的机密性和完整性, 对 PLC 使用的通信协议加入了认证机制; 或为了保证固件的安全性对固件增加了校验。因此, 首要的问题是绕过上述认证过程。

表 4 常见 PLC 通信协议漏洞类型

Table 4 Types of Protocol Vulnerability in PLCs

漏洞类型	漏洞编号
拒绝服务	CVE-2019-6571, CVE-2019-6848, CVE-2019-9590, CVE-2019-19279, CVE-2019-20045, CVE-2020-6986
信息泄露	CVE-2016-9159, CVE-2019-6584, CVE-2019-10920, CVE-2020-15791, CVE-2020-10628, CVE-2020-10276
配置篡改	CVE-2018-7790, CVE-2018-7791, CVE-2019-6279, CVE-2019-10943, CVE-2019-13533

Haroon Wardak 等人[42]人工分析了 Siemens S7Comm 协议中用于控制系统数据访问权限的私有加密算法。如图 16 所示, 此算法将用户的 8 位密码(P1,P2…P8)通过简单的 XOR 运算转化成 8 位密钥(C1,C2…C8), 安全性很弱易被伪造。在此基础上, 作者实现了针对 Siemens S7-400 PLC 用户密码的窃取、系统功能块的更改与删除攻击, 在真实系统上展示了其产生的影响。

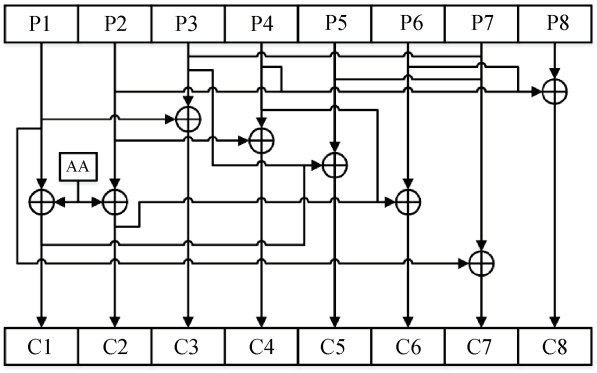


图 16 Siemens S7-400 PLC 密码编码机制

Figure 16 The Password Encoding Mechanism of Siemens S7-400 PLC

Ryan Grandgenett 等人[43]使用逆向工程的方法详细分析了 CIP(Common Industrial Protocol)协议通信的认证过程, 如图 17 所示。作者首先通过捕获 Allen Bradley's RSLogix 5000 与 Allen Bradly's ControlLogix 5573 PLC 间的通信流量, 定位加密字段; 然后结合对 RSLogix 5000 软件逆向分析, 推断出上位机使用基于 RSA 和 SHA-1 算法, 将 PLC 发送的 challenge 随机数转换成 20 字节的 response 字段; 最后作者构造攻击载荷对 PLC 发起远程 I/O 篡改攻击。本文不仅为协议加密分析开拓了新的思路, 而且对协议脆弱性进行了多角度分析: 数据未加密、使用硬编码密钥、挑战字段未实时更新、使用开源加密库等等, 对于协议安全设计有重要的指导意义。

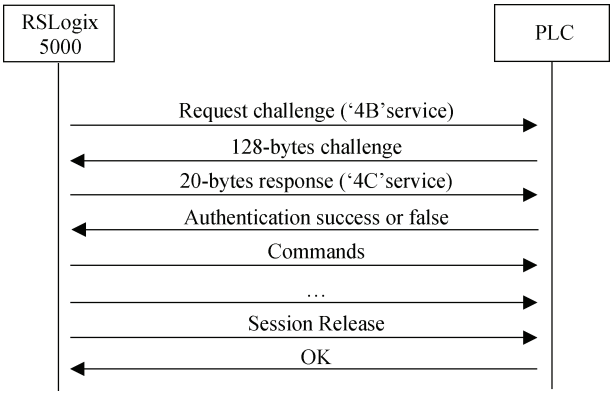


图 17 协议挑战-应答认证机制

Figure 17 Challenge/Response Authentication Machanism

Sushma Kalle 等人[32]破解了 Schneider Electric Modicon M221 读控制程序的认证过程。M221 PLC 会将用户设定的 SHA-256 密码哈希值与随机数 A 与随机数 B 进行异或运算, 并保存到固定地址, 如图 18 所示。当用户发起读控制程序的请求前, 需要发送原密码。作者利用了一种 0-day 漏洞: 由于密码存储位置固定, 只需要对保存的密码进行覆盖, 就可以任意篡改密码, 从而绕过认证。

Cheng Lei 等人[44]使用逆向工程方法破解了西门子 S7CommPlus 协议通信过程中的加密算法, 基于 S7-1200v4.1 PLC。此版本 S7CommPlus 协议加密存在两个阶段, 一是在建立通信阶段, 使用两种基于 XOR 的私有加密算法实现; 另一个是在发送控制命令时, 使用一种更复杂的私有加密算法实现。两种加密的实现程序都保存在 TIA Portal 目录下。作者认为这种私有加密算法仍然容易被破解, 并指出基于公私钥的加密算法更为安全。

Eli Biham 等人[45]使用逆向工程的方法破解了最新版本 S7CommPlus 协议中的加密算法, 基于 S7-1500 PLC。本文总结了 S7CommPlus 协议三种版本的加密方案, 并详细分析了最新版本的加密方案, 如图 19 所示。在双方确定密钥 Session Key 后, 后续通信数据中都必须包含由 HMAC-SHA256 算法加密的完整性检查字段, 来保证数据的完整性。

4.4.2 设备层

PLC 除了用于过程控制, 也可以与其他 IED 进行 TCP/UDP 通信, 传输数据。从用户代码的角度, 研

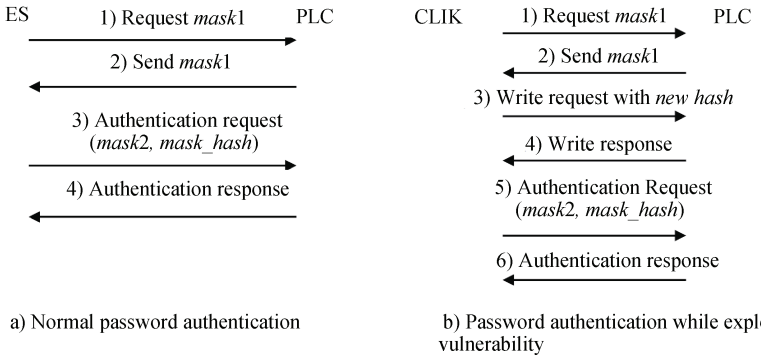


图 18　漏洞利用前后的密码认证过程

Figure 18　Password Authentication Before and After Exploiting the Vulnerability

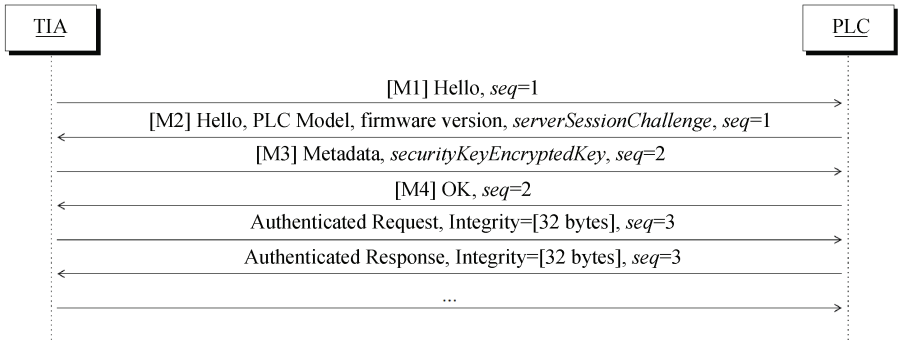


图 19　S7CommPlus 会话建立过程

Figure 19　The S7CommPlus Session Establishment

究人员可以利用这些系统函数, 来实现传统 IT 领域的恶意程序。但缺点在于会对原始程序执行的循环时间可能会产生影响, 同时通过上位机软件很容易检测到, 或者通过 IDS 对网络流量数据的实时监测, 捕获到 PLC 间异常的通信行为。

Johannes Klick 等人[46]提出一种针对 Siemens S7-300 PLC 的恶意后门程序, PLCinject, 分为两个阶段。如图 20 a)所示, 作者需要获取边界 PLC1 的 IP, 来计算其他子网地址, 借助 TCON、TUSEND 等系统函数构建 SNMP 内网扫描器, 以获取到子网内存活的 PLC。如图 20 b)所示, 作者使用 TCP 相关系统函数在 PLC 中实现了 SOCKS 5 代理后门, 来通过边界 PLC1 作为代理节点对探测到的 PLC 进行远程控制。

Ralf Spenneberg 等人[47]提出一种针对 Siemens S7-1200 PLC 的蠕虫病毒, 能够仅通过 PLC 之间自动传播到其子网内其他存活的 S7-1200 PLC, 并执行恶意载荷。此蠕虫病毒的设计思路可以扩展到多个厂家的 PLC 设备。

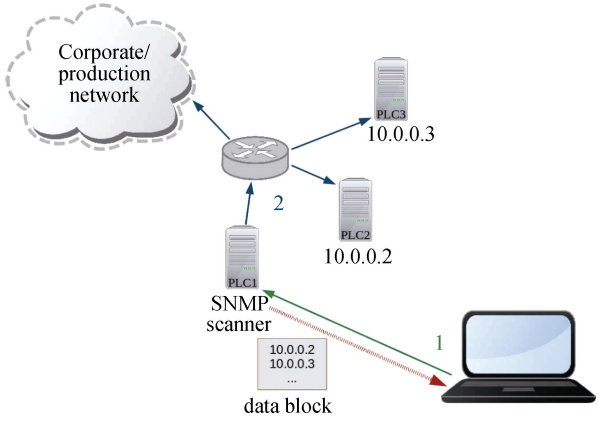
PLC 蠕虫总体架构如图 21 所示。作者首先使用 TCP 相关系统函数实现了内网扫描器, 能够侦测到子网中在同一网段内存活的 S7-1200 PLC; 然后作者分析了西门子上位机 TIA Portal 与 PLC 的通信过程, 基于西门子 S7CommPlus 协议伪造 PLC 程序下载过程, 将自身复制到已探测的其他 PLC 上, 包括自动传播载荷和恶意行为执行载荷, 以实现类似蠕虫的传播; 最后, 恶意行为执行载荷被插入到西门子 PLC OB1 功能块, 受感染的 PLC 启动后会顺序调用执行预先编写的恶意载荷, 实现恶意功能如构建远程 C&C 服务器、成为 Sock4 代理或输出值篡改等。

此外, 作者总结了实现 PLC 蠕虫的三个必备条件: 支持工业以太网传输、通过 TCP/UDP 传输程序、存在可编程的 TCP 相关系统函数。

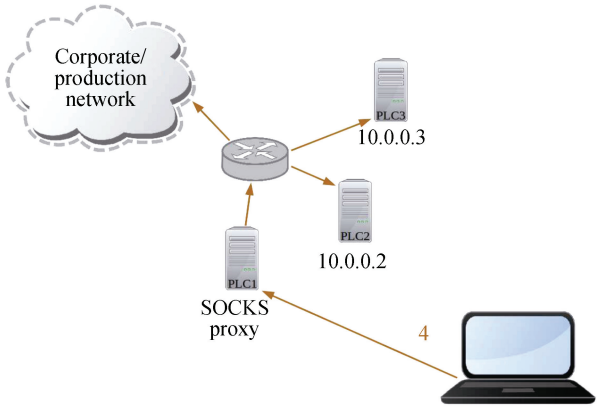
4.5　隐蔽驻留

针对 PLC 构建的恶意程序, 使用上位机可以检测到。而固件层的攻击很难检测, 可以实现在 PLC 上的长期隐蔽驻留。

Carl D. Schuett 等人[37]绕过固件校验算法将修改后的固件下载到设备中, 实现了借助固件中的后门程序远程触发 PLC 停止运行的目的。Zachary H. Basnight 等人[48]通过逆向分析 PLC 固件中的校验算法, 批露了用于校验固件、基于 CRC-32 的 ab_chsum 函数, 并绕过了针对 PLC 固件的 CRC 校验过程, 并将新的固件下载到 PLC 中, 实现了对 PLC 固件版本的篡改。Luis Garcia 等人[39]提出针对电网系统进行



a) 利用PLC1探测内网中的SNMP设备



b) 利用PLC1作为网关访问内网的其他PLC

图 20 PLC 后门攻击周期

Figure 20 PLC Back Orifice Attack Cycle

攻击的 PLC Rootkit, HARVEY, 总体架构如图 22 所示。其能够在固件层拦截 PLC 向 HMI 传送的数据并进行篡改, 导致操作员无法监视到异常, 如输入端口 GPIO E 和 GPIO F 对应内存地址默认值为 0xFFFFFFFF, 被篡改之后为 0xFFFFFFFC, 导致输入的默认值为 1, 但在 HMI 上检测到的数据仍为 0。

Eli Biham 等人[45]伪造了一种针对 Siemens S7-1500 PLC 的工程师站, 其能够向 PLC 中下载任意代码, 且即使在 PLC 上执行恶意代码, 在真实 TIA Portal 上位机检测到的仍为原始的用户代码, 以实现隐蔽执行的目的。实际上, 用户在 TIA Portal 看到的是 PLC 中 Source Code 转换得到的梯形图程序, 而 PLC 实际执行的是 Object Code 对应的二进制代码。作者首先使用 TIA Portal 编译得到两种代码, 利用伪造的工程师站将其下载到 PLC 中, 以达到隐蔽执行的目的。

4.6 远程控制

由于 PLC 通常位于工业内网中, 难以进行远程控制。攻击者通常需要在 PLC 中植入恶意后门程序, 实现远程对 PLC 发送任意控制命令。

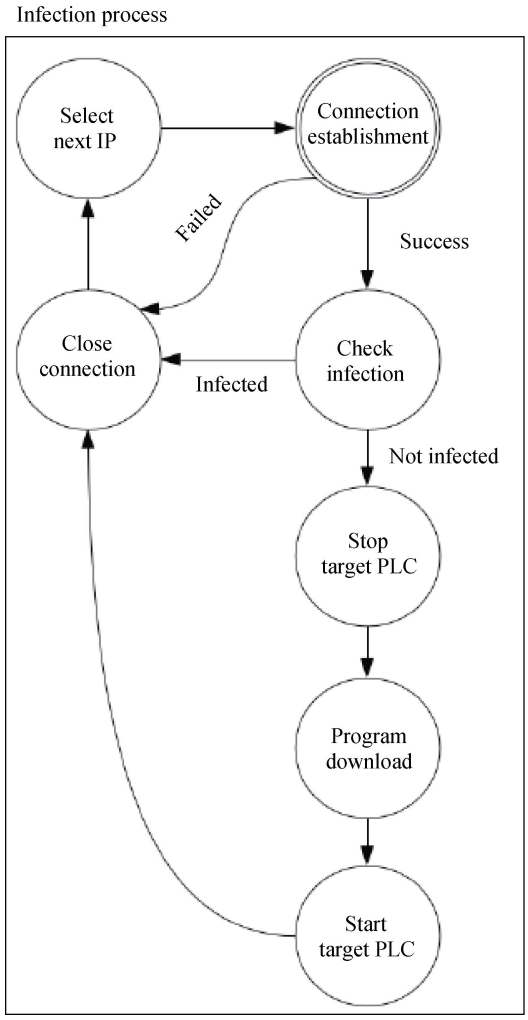


图 21 PLC 蠕虫执行过程

Figure 21 Execution Process of the PLC Worm

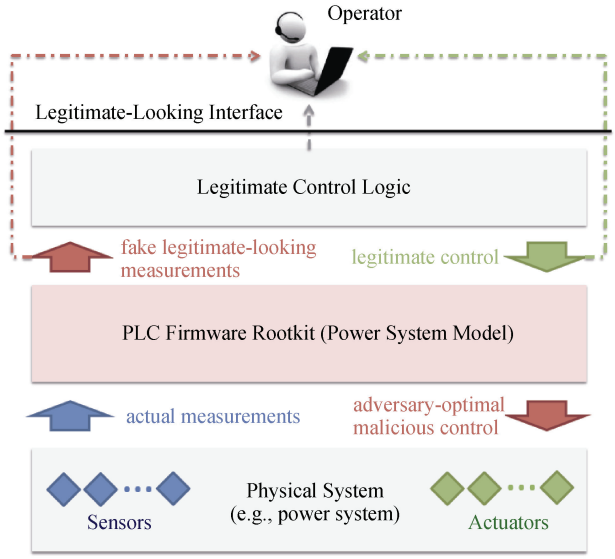


图 22 HARVEY 双向数据篡改攻击

Figure 22 HARVEY Two-Way Data Manipulation Attack

Johannes Klick 等人[46]在 PLC 上实现的 SOCKS 5 代理后门程序, 其支持任意数据的转发。攻击者可以远程直接向处于内网的 PLC 发送启停等恶意指令。Ralf Spenneberg 等人[47]提出一种针对 Siemens S7-1200 PLC 的蠕虫病毒在受感染 PLC 上会主动连接远程 C&C 服务器, 处理远程攻击者的任意控制命令。Carl D. Schuett 等人[37]通过篡改固件层上 CIP 协议栈实现的代码, 植入后门。进而通过远程发送恶意 CIP 数据包, 干扰 PLC 的系统状态。

4.7 目的实现

攻击者一方面可以通过攻击 PLC, 实现恶意操纵工控系统现场设备的目的。

Anastasis Keliris 等人[16]通过修改 PLC 中原始 PID 算法的控制参数, 实现了对核反应堆压力的控制, 实际上这种攻击可以拓展到到任意工控系统中的控制算法。Luis Garcia 等人[39]实现了两种攻击场景, 一种通过控制断路器开启关闭的频率, 降低电力系统稳定性; 另一种通过实现修改后得 OPF 算法, 使用输出的不合法控制策略干扰系统正常运行。

另一方面可以使用 PLC 作为代理节点, 进行数据窃取或攻击其他工控组件。

Saranyan Senthivel 等人[49]提出可以通过逆向 PLC 返回到编程软件的二进制程序数据, 并进行动态修改, 来挖掘编程软件的漏洞, 实现使编程软件崩溃甚至任意代码执行的目的。

5 基于杀伤链模型的 PLC 防御技术分析

PLC 防御技术可以从通信协议安全防护、控制程序验证、执行过程检测、PLC 取证技术 4 个方面进行分类, 表 5 描述了其与杀伤链模型攻击阶段的映射关系。

表 5 杀伤链模型攻击阶段-PLC 防御技术映射关系

Table 5 Mapping between Cyber Kill Chain Attack Phases and PLC Defense Techniques

攻击阶段方法	侦察识别	武器构建	载荷投递	漏洞利用	隐蔽驻留	远程控制	目的实现
协议安全防护	●		●	●			
控制程序验证		●				●	
执行过程检测		●		●	●	●	
PLC 取证技术	●		●			●	

5.1 协议安全防护

目前大多数 PLC 都支持通信过程的认证, 认证过程可以在通信会话建立的过程中, 也可以在实现特定功能的通信过程中。不同的 PLC 通信协议使用的加密方案往往有很大不同。研究人员也在探索更高效和安全的通信协议安全防护方案。

Ivan Bestak 等人[60]对多种 PLC 通信协议加密算法进行测试, 包括 AES、Twofish、3DES 等等, 这些通信算法不仅保证了流程的安全性, 也有效提高了网络吞吐量。在真实 PLC 网络中下进行测试, 实验结果表明 3DES 算法相比 AES 算法网络吞吐量下降了 20%。

Andrew Clark 等人[61]提出一种新型系统防御框架, 其可以对操作站发送至 PLC 的控制命令使用随机生成的密钥集进行认证, 对异常流量会记录警报事件, 密钥认证系统模型如图 23 所示。研究表明此认证机制在密钥量较大时能有效减低攻击者的成功概率, 加密层和 IPS 提高了 PLC 和监控软件之间通信的安全性, 有效防止拦截、注入和 DoS 攻击。

Thiago Alves 等人[62]针对 OpenPLC 系统提出一种加密方案, 且此方案不依赖于通信协议的种类, 如 Modbus、DNP3 等。加密过程如图 24 所示。通信双方使用预先确定的密钥, 对数据包内的所有通信数据使用 AES-256 加密, 并且双方都需要部署加密与解密模块, 以实现 PLC 功能的正常运转。本文将此加密方案集成到一个硬件模块内, 工程师可以使用此设备来批准或拒绝传输。

5.2 控制程序验证

由于针对 PLC 的恶意程序会改变原始程序的语义, 进而改变其控制流, 很多研究工作专注于验证 PLC 控制程序的合法性, 或者识别原始程序中的漏洞, 进而保证 PLC 程序的正确执行。

J. Malchow 等人[63]将传统的 ACCAT Guard 概念应用到 PLC 上, 提出在上位机与 PLC 之间设置 PLC Guard 对发送到 PLC 的控制程序进行合法性验证。作者在 Siemens S7-313C PLC 上实现了其原型, PLC Guard 会从 PLC 控制程序中提取各种功能块间的控制依赖关系, 并将新程序与原程序功能块进行比较以实现程序合法性的验证, 如图 25 所示, 黑色箭头为正常的程序运行过程, 功能块 l_lf4 被检测到行为异常。此外, 由于很多攻击会通过篡改 PLC 控制程序逻辑指令来干扰输出, PLC Guard 会对新程序与源

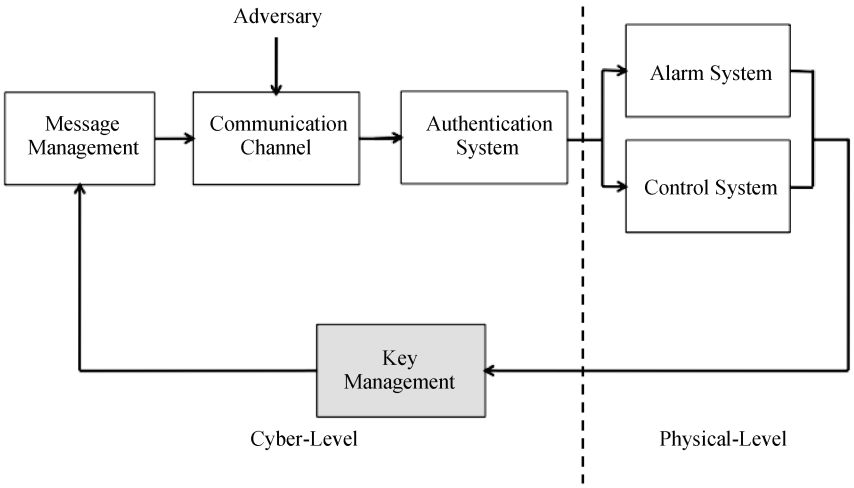


图 23 密钥解决方案系统模型

Figure 23 System Model of The Cryptographic Solution

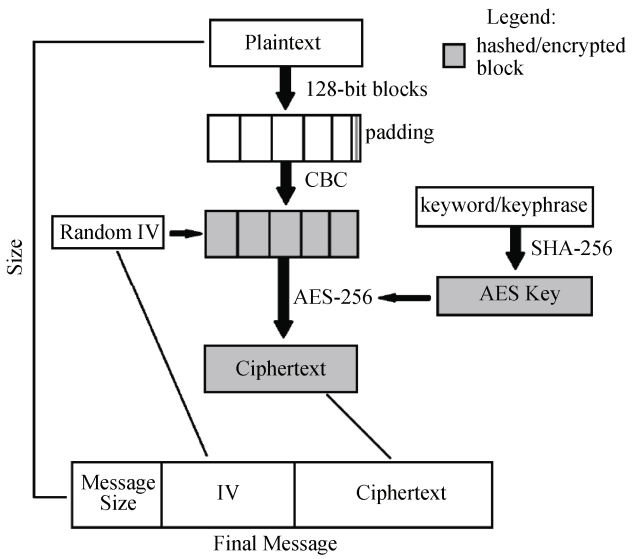


图 24 OpenPLC 加密过程

Figure 24 Encryption Process of OpenPLC

程序中常见逻辑指令如 AND、XOR、Call、Jump 等进行统计比较, 来确保新程序的合法性。

Yu Jiang 等人[64]提出使用 Runtime Reliability Model (RRM) 来分析 PLC 的可靠性, 即系统在特定条件下维持稳定的概率, 如某传感器、控制器组件故障对 PLC 系统的稳定性的影响。作者提出一种由 PLC 控制程序自动构建 RRM 的方法, 其反映了 PLC 系统内所有控制变量间的条件依赖关系。如图 27 b)表示由图 26a)生成的 RRM 模型, 其中节点表示 PLC 中的控制变量, 边表示控制变量间的依赖关系。最后, 基于 RRM 作者使用条件概率分布表来对系统稳定性进行评估。

Luis Garcia 等人[65]提出可以将用 ST 语言编写的、单任务的、无优先级调度的 PLC 程序与用微分动态逻辑 dL 表达的混合程序进行相互转换, 并实现了一种自动转换工具, HyPLC, 用于验证 PLC 程序的合法性, 系统架构如图 27 所示。HyPLC 将 PLC 程序转换成混合程序, 基于 KeYmaera X 理论对其进行逻辑验证, 以实现对 PLC 程序的合法性评估。然后 HyPLC 将验证后的合法程序再次转换成 PLC 程序, 传入 PLC 执行, 进而将复杂信息物理系统的形式验证与具体 PLC 代码实现联系起来。

Mu Zhang 等人[66]提出一种自动化审查 PLC 控制程序的系统, VETPLC, 将设备工作时间信息与事件信息结合, 来发现安全违规行为。VETPLC 能够基于 PLC 控制程序自动化生成对应的时序事件因果关系图(TECG), 其反映了程序中各个变量间得控制依赖与时序依赖关系, 可用于自动化代码安全审计。作者构建了汽车模拟生产线中零件的装卸过程, 包含 PLC、CNC、工业机器人, 生成的 TECG 模型如图 28 所示, 根据时序事件关系与实际场景寻找引起安全事件的因素, 同时也可用于工控设备异常行为的监测。

Denis Cousineau 等人[67]提出一种针对 PLC 梯形图程序的形式化验证方法。作者首先将 PLC 梯形图程序自动转化为基于 Why3 的程序模型, 然后根据此模型中变量的关系生成条件约束, 最后使用 SMT 求解器 CVC4 进行求解, 寻找可以导致 PLC 程序出现异常的变量组合。

Shengjian Guo 等人[68]提出一种针对 PLC 控制程序的符号执行工具, SYMPLC, 系统架构如图 29 所示。作者首先使用 MATIEC 工具将单任务与多任务 PLC 控制程序转换为 C 语言程序, 由于 PLC 程序执

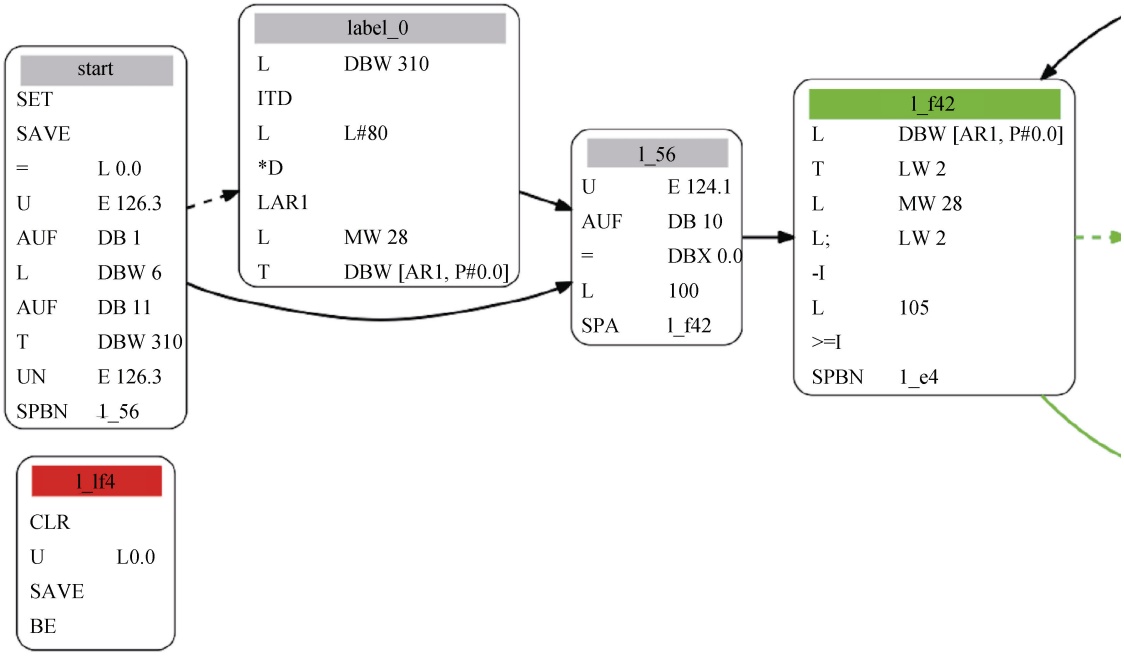
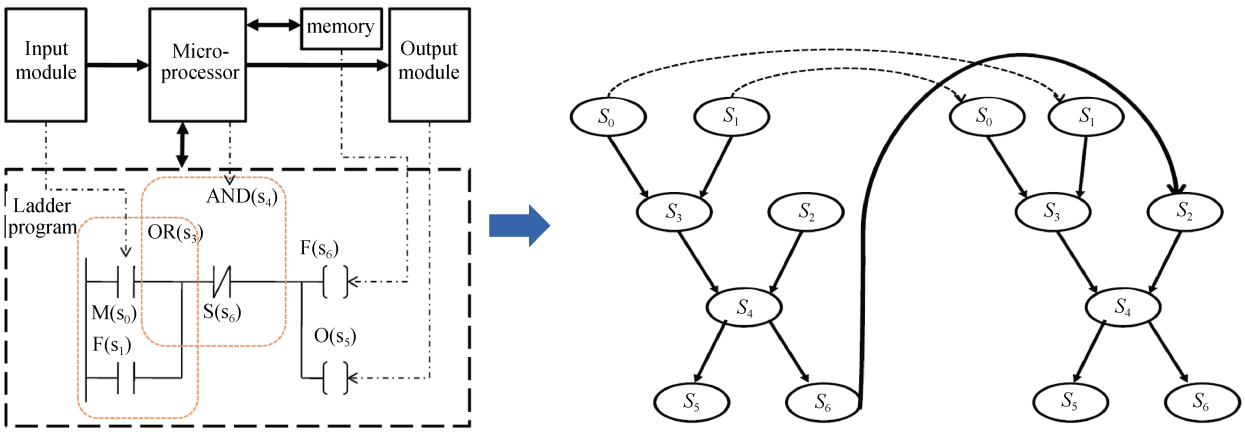


图 25 PLC 程序控制流程图

Figure 25 PLC Program Control Flow Graph



a) PLC控制程序 b) RRM模型

图 26 RRM 模型生成过程

Figure 26 RRM Model Generation

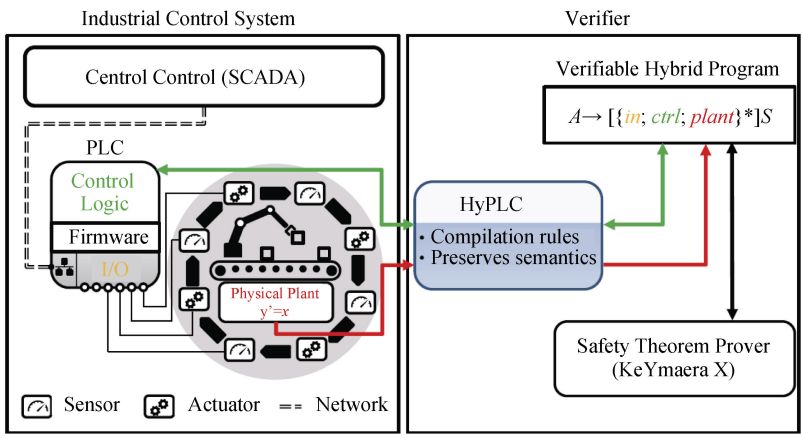


图 27 ST 程序及其对应的混合程序

Figure 27 ST Program and Generated Hybrid Program

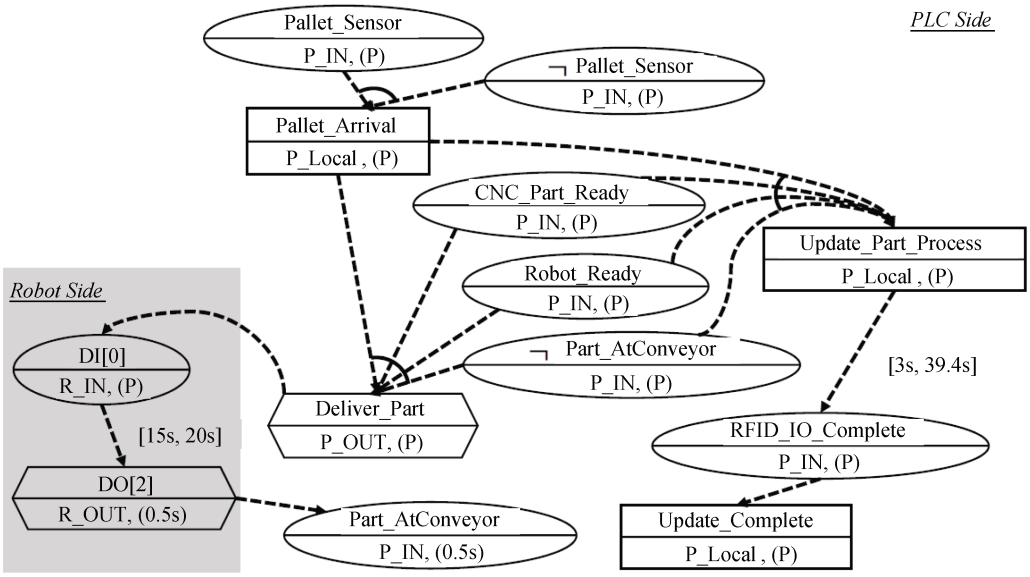


图 28　TECG 模型

Figure 28　Timed Event Causality Graphs Model

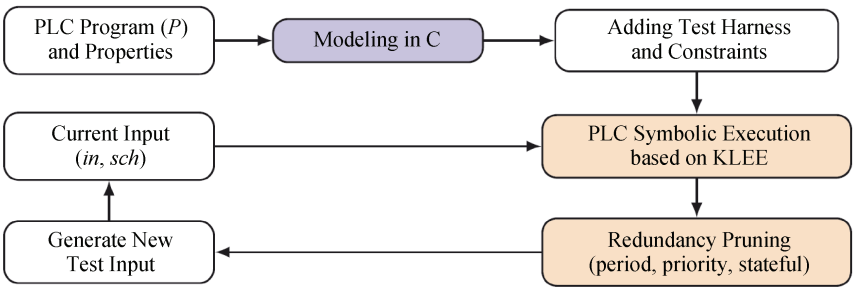


图 29　SYMPLC 系统架构

Figure 29　System Architecture of SYMPLC

行具有周期性、优先级等特点, 需要指定最大循环周期时间, 即多任务周期的最小公倍数, 并对全局变量的访问自动生成约束条件。SYMPLC 使用多线程符号执行技术, 基于 Cloud9 符号执行引擎对转换后的 C 语言程序进行模糊测试。实验结果表明能够有效挖掘多种 PLC 控制程序漏洞。Li Hao 等人[69]针对 PLC ST 语言编写的程序, 使用动态符号执行技术自动化生成测试用例。相比 SYMPLC 其减少了 87.5% 的测试生成用例, 但代码分支覆盖率与 SYMPLC 几乎一致, 均接近 100%。

5.3　执行过程监测

PLC 在执行过程中通常会将相关数据实时上传到 HMI 等监控设备。研究人员可以通过分析流量数据、建立相关状态模型对当前系统安全态势进行威胁评估。

Dina Hadžiosmanovic 等人[70]提出一种针对工控系统的入侵检测系统, 能够连续跟踪生产过程中的 PLC 的控制变量数据, 并以此建立自回归模型, 结合实际测量值进行安全监测, 攻击者修改了水箱阈值配置, 会使水箱实际水量超出其上限。如图 30 所示, 模型可以准确的预测实际水量的变化, 达到对攻击进行及时响应的目的。

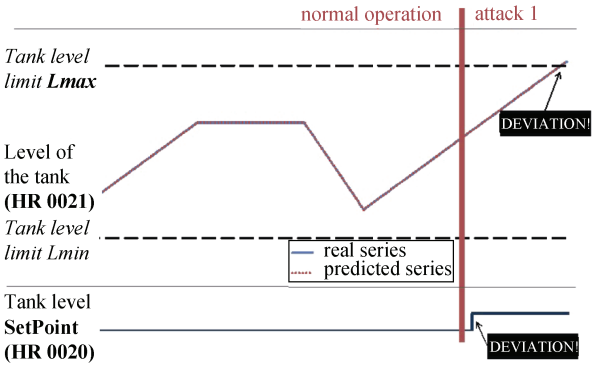


图 30　PLC 配置篡改攻击

Figure 30　Configuration Manipulation Attack on PLC

Niv Goldenberg 等人[71]提出一种有限状态机(DFA)模型, 用于对 Modbus 协议通信过程进行建模, 此模型通过检测异常状态序列来发现入侵事件。同时, 该模型能识别对 HMI 的攻击以及对 PLC 的错误配置。William Jardine 等人[72]提出一种无干扰的主动型入侵检测系统, SENAMI, 通过实时监测 PLC 状态信息检测恶意攻击, 如探测识别、拒绝服务、数据篡改等。SENAMI 针对 Siemens PLCs, 制定多种检测规则如 IP 源、请求时间、控制逻辑上传下载、PLC 内部数据块配置等, 来实现对上述攻击的检测识别。相比传统基于 Snort 的入侵检测规则, SENAMI 使用的规则能更有效、更深层次的监测 PLC 的实时状态以及行为, 且开销也在有限的范围内。Ken Yau 等人[73]提出使用 OCSVM 半监督学习算法对 PLC 行为进行建模, 针对一段时间内模拟交通信号系统的网络数据的捕获, 建立 OCSVM 模型, 来对异常信号状态或操作进行识别。

对 PLC 执行过程监测可以使用被动方法, 在其外部进行数据监视, 也可以将监测模块置于 PLC 内部, 在不影响原始程序执行的条件下, 实现更实时、更准确的响应。

Luis Garcia 等人[74]提出一种嵌入式监视程序, 其可以集成到 PLC 的扫描循环中, 并与 PLC 共享系统资源来实现协议分析与安全认证。如图 31 所示, 进程管理程序将 PLC 的实时数据导入 Deterministic Finite Automation (DFA)模型进行合法性验证, 以对 Siemens PLC 控制状态进行安全分析。

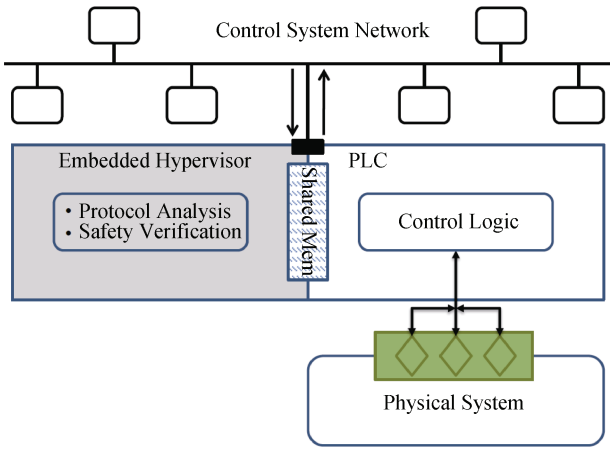


图 31 嵌入式监视器与 PLC 组合

Figure 31 Coupling of Embedded Hypervisors and PLC

Huan Yang 等人[75]提出使用 PLC Runtime 行为模型来检测恶意载荷攻击, 并将该检测器集成到 PLC 固件中。如图 32 所示, 此固件级的检测机制通过监视 I/O 访问模式、网络访问模式以及 PLC 控制程序时序特性, 建立时序行为矩阵, 并将其内置到 PLC 系统中, 以实现对恶意载荷进行识别。此外, 其内存和时间开销都不会影响 PCL 控制程序的执行。

I/O Event Table

I/O Event	Legitimate Values/Ranges
I:0/0	1 (HIGH), 0 (LOW)
⋮	⋮
I:1.0	0:3 (0~3V), 12:15 (12~15V)
⋮	⋮
O:2/8	1 (HIGH), 0 (LOW)
⋮	⋮
O:3.0	12:15 (12~15V)
⋮	⋮

Network Event Table

Network Event	Legitimate Packet Counts
receiving	1
sending	1, 3

Timing Behavior Matrix

	I:0/0:1	I:0/0:0	⋮	I:1.0:0:3	I:1.0:12:15	⋮	O:2/8:1	O:2/8:0	⋮	O:3.0:12:15	⋮	Recv:1	Send:1	Send:3
I:0/0:1													2000	
I:0/0:0														
⋮														
I:1.0:0:3										300				
I:1.0:12:15														
⋮														
O:2/8:1													500	
O:2/8:0													500	
⋮														
O:3.0:12:15														
⋮														
Recv:1													1500	1500
Send:1														
Send:1														

图 32 Runtime 时序行为矩阵

Figure 32 Runtime Timing Behavior Matrix

Ali Abbasi 等人[76]在 WAGO PFC200 PLC 系统内部实现了 PLC 程序控制流完整性验证机制, ECFI, 系统架构如图 33 所示。作者针对 PLC 程序的执行特性改进了传统的 CFI 机制, 如检测到异常后只会发出警告不会中断程序执行、CFI 验证过程与控制程序执行进行分隔等。作者提出使用 Ring Buffer 存储实时控制流数据。由于攻击者往往通过修改间接跳转中的寄存器值或返回指令前的返回地址, 来达到控

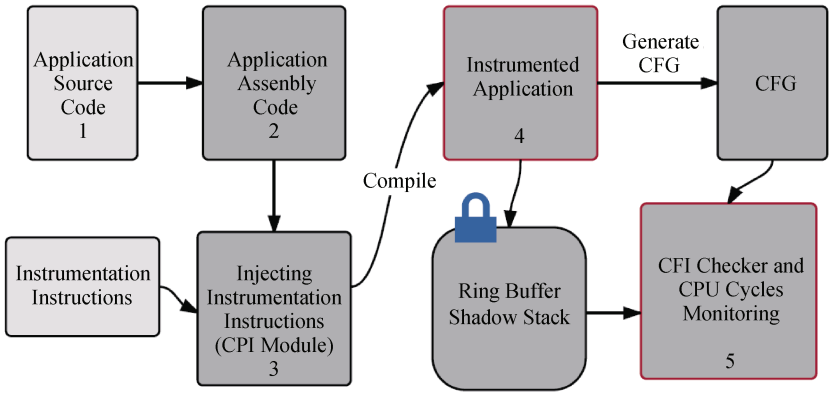


图 33 ECFI 系统架构

Figure 33 System Architecture of ECFI

制流劫持的目的, Ring Buffer 在上述指令执行前会获取执行权限, 通过比较其与预先生成的控制流图之间的差异来检测控制流劫持攻击。

5.4 PLC 取证技术

取证是网络安全生命周期中的一个必不可少的过程, 它不仅有助于识别事件发生的原因, 而且还有助于开发和设计未来的更安全的系统。现有针对 PLC 的攻击技术越来越隐蔽, 如何对受攻击的 PLC 控制系统进行取证显得尤为重要。由于工控系统相比传统控制系统对实时性的要求更高, 传统的取证工具难以直接应用, 难点主要体现在以下几个方面:

1) 专用组件与协议: 工控系统中使用的许多组件都是供应商特定的和专有的。它们包含供应商特定的软件, 包括固件、操作系统、专有的数据结构和协议。如果供应商发布了新的组件, 则未记录的更改也可能会影响已经开发和测试的取证工具。

2) 资源限制: PLC 等工控组件往往使用实时操作系统, 为保证实时性, 其处理能力与存储容量方面资源有限, 取证工具不能对这些组件产生负面影响。

3) 低层次的大量数据: 工控系统由多层组成。由于各个传感器生成大量数据, 因此在系统中捕获和分析来自较低层的数据非常具有挑战性。例如, 在电网中, 传感器每秒可产生多达 4,000 个测量值, 并且仅将压缩信息转发到更高的层。

4) 实时取证与响应: 一方面, 工控系统在运行中无法简单的关闭和分析, 在系统和设备运行时需要进行实时取证分析, 并减少对操作延迟的干扰。另一方面, 发生事件时, 取证证据将达到高峰。随着时间的流逝, 潜在的证据可能会被新的流程所覆盖, 因此迅速做出响应至关重要。

蜜罐作为一种新兴的主动防御技术, 通过构建可控的诱饵环境, 来捕获高质量的原始攻击数据。安全人员通过对日志进行数据分析, 来发现未知攻击威胁。游建舟等人[22]对工控蜜罐进行了深度分析, 工控蜜罐构建通常从工控协议解析出发, 通过仿真工控协议服务并绑定工控设备默认端口, 来诱导攻击者发起攻击。常见的工控蜜罐包括 Conpot[77]、snap7[53]、XPOT[78]、S7commTrace[79]等等。

此外, 一些研究人员提出可以在设备内部进行实时数据记录。Chun-Fai Chan 等人[80]通过在 PLC 内部添加监视和日志记录机制来实现对工控系统安全性的取证, 如图 34 所示。在程序执行的过程中 PLC 会将功能块数量、关键变量数值通过 Security Block 实时发送到历史数据库。Ken Yau 等人[81]提出了一种针对 PLC 的日志记录系统, 可捕获工控安全取证所需的数据。日志记录系统可以捕获 PLC 和其他网络设备之间的通信流量, 并剖析网络数据包, 提取取证信息, 最后以带有时间戳的记录的形式进行记录。

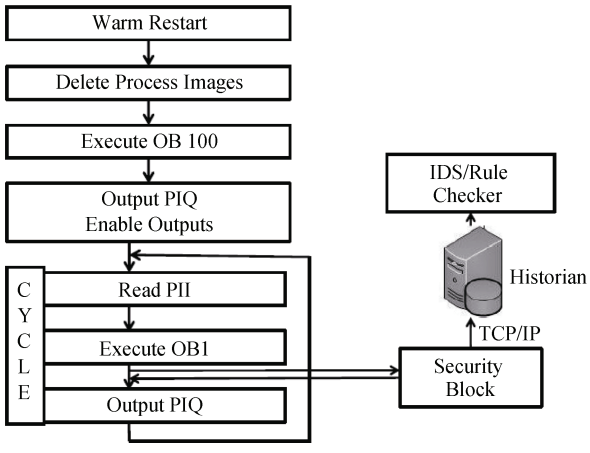


图 34 配置安全块的扫描循环

Figure 34 Scan Cycle with a Security Block

6 PLC 安全总结与展望

本文结合杀伤链模型对近些年 PLC 各类攻击与

防御技术进行了梳理。结合 PLC 软件架构可以发现, 现有PLC攻击技术涵盖了PLC软件架构的各个层次, 而现有针对 PLC 的防御技术多倾向于网络入侵检测以及控制程序的验证, 不足以应对未来更深层、更复杂的攻击方式。

综上, 本文针对 PLC 安全研究提出如下建议:

1) 从 PLC 作为嵌入式设备的角度, 研究安全 PLC 相关技术。在 I/O 层面, 增加双通道设计, 可以对 I/O 信号进行比较和检验, 以应对外部模块故障问题; 在CPU执行层面, 增加冗余设计, 当系统故障时可以使用另一个 CPU 继续执行, 有效应对工控系统紧急情况。

2) 从PLC作为工业控制器的角度, 研究PLC系统层面的控制程序异常监测技术, 即将控制流和数据流异常检测融合到 PLC 系统内部执行过程中, 以提高监测过程的准确性与时效性。

3) 从 PLC 作为工业控制网络组件的角度, 研究更安全、更高效的协议加密技术, 以提高工控协议的完整性与机密性, 以应对数据重放、篡改等对工控系统产生不良影响的攻击行为。

参考文献

[1] OWASP Top 10 Application Security Risks, OWASP. https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_Top_10, 2017.

[2] Waterfall, The Top 20 Cyberattacks on Industrial Control Systems Whitepapaer, 2019.

[3] W32.Stuxnet Dossier, Symantec. https://www.wired.com/ images_blogs/threatlevel/2010/11/w32_stuxnet_dossier.pdf, 2010.

[4] Attackers Deploy New ICS Attack Framework "TRITON" and Cause Operational Disruption to Critical Infrastructure, Fireeye. https://www.fireeye.com/blog/threat-research/2017/12/attackers-deploy-new-ics-attack-framework-triton.html, 2017.

[5] Industroyer: Biggest Malware Threat to Critical Infrastructure Since Stuxnet, ESET. https://www.eset.com/int/industroyer/, 2017.

[6] Energy Company Hit with DoS Attack Last Spring Identified as sPower, SC Magazine. https://www.cyber-consult.org/index.php/2019/11/01/energy-company-hit-with-dos-attack-last-spring-identified-as-spower/, 2019.

[7] Pascal. Ackerman, The Purdue Model for Industrial Control Systems, 2017.

[8] Gonzalez D, Alhenaki F, Mirakhorli M. Architectural Security Weaknesses in Industrial Control Systems (ICS) an Empirical Study Based on Disclosed Software Vulnerabilities[C]. *2019 IEEE International Conference on Software Architecture*, 2019: 31-40.

[9] Ahmed I, Obermeier S, Sudhakaran S, et al. Programmable Logic Controller Forensics[J]. *IEEE Security & Privacy*, 2017, 15(6): 18-24.

[10] Serhane A, Raad M, Raad R, et al. Programmable Logic Controllers Based Systems (PLC-BS): Vulnerabilities and Threats[J]. *SN Applied Sciences*, 2019, 1(8): 924.

[11] Xu Z, Zhou X J, Wang L M, et al. Recent Advances in PLC Attack and Protection Technology[J]. *Journal of Cyber Security*, 2019, 4(3): 48-69.
(徐震, 周晓军, 王利明, 等. PLC 攻防关键技术研究进展[J]. *信息安全学报*, 2019, 4(3): 48-69.)

[12] Petruzella F D, Programmable Logic Controllers, McGraw-Hill Companies, 2010.

[13] Altera, Industry 4.0 Drives New Approaches in PLC Design, 2015.

[14] Gatess S. A Beginner's PLC Overview, in PLC Processors (CPUs), 2017.

[15] Milinković S A, Lazić L R. Industrial PLC Security Issues[C]. *2012 20th Telecommunications Forum*, 2013: 1536-1539.

[16] Keliris A, Maniatakos M. ICSREF: A Framework for Automated Reverse Engineering of Industrial Control Systems Binaries[C]. *Proceedings 2019 Network and Distributed System Security Symposium*, 2019: 2.

[17] Abbasi A, Scharnowski T, Holz T, Doors Of-Durin The Veiled Gate To Siemens S7 Silicon[C]. in *Black Hat Europe*, 2019.

[18] CODESYS Software Architecture, CODESYS. http://www.codesys.cn/list-chanpinzongshu.html.

[19] Siemens, S7-1200 Programmable Logic Controllers, 2016.

[20] Modbus Application Protocol Specification V1.1b, Modbus. http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf, 2012.

[21] Schneider Electric Modicon M580 FTP Firmware Update Loader Service Denial-of-Service Vulnerability, Jared Rittle. https:// talosintelligence.com/vulnerability_reports/TALOS-2019-0822, 2019.

[22] You J Z, Lv S C, Sun Y Y, et al. A Survey on Honeypots of Internet of Things[J]. *Journal of Cyber Security*, 2020, 5(4): 138-156.
(游建舟, 吕世超, 孙玉砚, 等. 物联网蜜罐综述[J]. *信息安全学报*, 2020, 5(4): 138-156.)

[23] Hutchins E, Cloppert M, Amin R. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains[J], *Leading Issues in Information Warfare & Security Research*, 2011, 1(1): 1-14.

[24] Malone S T, The Expanded Cyber Kill Chain[C]. *BlackHat*, 2016.

[25] Verdasys, Cyber Attack Defense A Kill Chain Strategy, White Paper, 2013.

[26] Kim H, Kwon H, Kim K K. Modified Cyber Kill Chain Model for Multimedia Service Environments[J]. *Multimedia Tools and Applications*, 2019, 78(3): 3153-3170.

[27] Martin L, Seven Ways to Apply the Cyber Kill Chain with a Threat Intelligence Platform, 2015.

[28] HosseiniNejad R, HaddadPajouh H, Dehghantanha A, et al. A Cyber Kill Chain Based Analysis of Remote Access Trojans[M]. Handbook of Big Data and IoT Security. Cham: Springer, 2019: 273-299.

[29] Dargahi T, Dehghantanha A, Bahrami P N, et al. A Cyber-Kill-Chain Based Taxonomy of Crypto-Ransomware Features[J]. *Journal of Computer Virology and Hacking Techniques*, 2019, 15(4): 277-305.

[30] Bahrami P N, Dehghantanha A, Dargahi T, et al. Cyber Kill

Chain-Based Taxonomy of Advanced Persistent Threat Actors: Analogy of Tactics, Techniques, and Procedures[J]. *JIPS*, 2019, 15(4): 865-889.

[31] Duncan A, Creese S, Goldsmith M. A Combined Attack-Tree and Kill-Chain Approach to Designing Attack-Detection Strategies for Malicious Insiders in Cloud Computing[C]. *2019 International Conference on Cyber Security and Protection of Digital Services*, 2019: 1-9.

[32] Kalle S, Ameen N, Yoo H, et al. CLIK on PLCS! Attacking Control Logic with Decompilation and Virtual PLC[C]. *Proceedings 2019 Workshop on Binary Analysis Research*, 2019: 2.

[33] McLaughlin S, Zonouz S. Controller-Aware False Data Injection Against Programmable Logic Controllers[C]. *2014 IEEE International Conference on Smart Grid Communications*, 2015: 848-853.

[34] McLaughlin S, McDaniel P. SABOT: Specification-Based Payload Generation for Programmable Logic Controllers[C]. *The 2012 ACM conference on Computer and communications security*, 2012, 10: 439-449.

[35] Govil N, Agrawal A, Tippenhauer N O. On Ladder Logic Bombs in Industrial Control Systems[M]. Computer Security. Cham: Springer International Publishing, 2017: 110-126.

[36] Serhane A, Raad M, Raad R, et al. PLC Code-Level Vulnerabilities[C]. *2018 International Conference on Computer and Applications*, 2018: 348-352.

[37] Schuett C. Programmable Logic Controller Modification Attacks for use in Detection Analysis[D]. in DEPARTMENT OF THE AIR FORCE, AIR FORCE INSTITUTE OF TECHNOLOGY, 2014.

[38] Abbasi A, Hashemi M. Ghost in the PLC: Designing an Undetectable Programmable Logic Controller Rootkit via Pin Control Attack[C]. *Black Hat*, 2016.

[39] Garcia L A, Brasser F, Cintuglu M H, et al. Hey, my Malware Knows Physics! Attacking PLCS with Physical Model Aware Rootkit[C]. *Proceedings 2017 Network and Distributed System Security Symposium*, 2017: 2.

[40] Niedermaier M, Malchow J, Fischer F, et al. You Snooze, You Lose: Measuring PLC Cycle Times under Attacks[C]. in *WOOT @ USENIX Security Symposium*, 2018.

[41] Weber T. Reverse Engineering Custom ASICs by Exploiting Potential Supply-Chain Leaks[C]. in *Black Hat*, 2019.

[42] Wardak H, Zhioua S, Almulhem A. PLC Access Control: A Security Analysis[C]. *2016 World Congress on Industrial Control Systems Security*, 2017: 1-6.

[43] Grandgenett R, Mahoney W, Gandhi R. Authentication Bypass and Remote Escalated I/O Command Attacks[C]. *The 10th Annual Cyber and Information Security Research Conference*, 2015, 4: 1-7.

[44] Cheng L, Li D H, Ma L. The Spear to Break the Security Wall of S7CommPlus[C]. in *BlackHat*, 2017.

[45] Biham E, Bitan S, Carme A. Rogue7: Rogue Engineering-Station attacks on S7 Simatic PLCs[C]. in *BlackHat*, 2019.

[46] Klick J, Lau S, Marzin D, et al. Internet-facing PLCs-A New Back Orifice[C]. in *Black Hat*, 2015.

[47] Spenneberg R, Brüggemann M, Schwartke H. PLC-Blaster: A Worm Living Solely in the PLC[C]. in *BlackHat*, 2016.

[48] Zachary H. Basnight, Firmware Counterfeiting and Modification Attacks on Programmable Logic Controllers[D]. in DEPARTMENT OF THE AIR FORCE, AIR FORCE INSTITUTE OF TECHNOLOGY, 2013.

[49] Senthivel S, Dhungana S, Yoo H, et al. Denial of Engineering Operations Attacks in Industrial Control Systems[C]. *The Eighth ACM Conference on Data and Application Security and Privacy*, 2018: 319-329.

[50] Nmap - the Network Mapper, https://github.com/nmap/nmap,

[51] Metasploit Framework, rapid7. https://github.com/rapid7/ metasploit-framework,

[52] pymodbus, https://github.com/riptideio/pymodbus,

[53] Snap7, D. Nardella. https://sourceforge.net/projects/snap7/files/, Jun. 2018.

[54] Cheminod M, Durante L, Seno L, et al. Performance Evaluation and Modeling of an Industrial Application-Layer Firewall[J]. *IEEE Transactions on Industrial Informatics*, 2018, 14(5): 2159-2170.

[55] Dheeraj R, Guo H Q, Veeravalli B, et al. Design and Development of SCADA Firewall Security Features for Protecting Industrial Operations[C]. *2019 IEEE VTS Asia Pacific Wireless Communications Symposium*, 2019: 1-5.

[56] Muslija A, Enoiu E. On the Measurement of Software Complexity for Plc Industrial Control Systems Using TIQVA[C]. *The 35th Annual ACM Symposium on Applied Computing*, 2020: 1556-1565.

[57] BACKDOOR IN SONY IPELA ENGINE IP CAMERAS, SEC Consult. https://sec-consult.com/en/blog/2016/12/backdoor-in-sony-ipela-engine-ip-cameras/, 2016.

[58] From China, With Love, Craig. http://www.devttys0.com/2013/10/from-china-with-love/, 2013.

[59] GHOSTS FROM THE PAST: AUTHENTICATION BYPASS AND OEM BACKDOORS IN WIMAX ROUTERS, SEC Consult. https://sec-consult.com/en/blog/2017/06/ghosts-from-past-authentication-bypass/, 2017.

[60] Bestak I, Orgon M. The Use of Encryption Algorithms in PLC Networks[J], *International Journal of Information Technology and Business Management*, 2013, 13: 38-43.

[61] Clark A, Zhu Q Y, Poovendran R, et al. An Impact-Aware Defense Against Stuxnet[C]. *2013 American Control Conference*, 2013: 4140-4147.

[62] Alves T, Das R, Morris T. Embedding Encryption and Machine Learning Intrusion Prevention Systems on Programmable Logic Controllers[J]. *IEEE Embedded Systems Letters*, 2018, 10(3): 99-102.

[63] Malchow J O, Marzin D, Klick J, et al. PLC Guard: A Practical Defense Against Attacks on Cyber-Physical Systems[C]. *2015 IEEE Conference on Communications and Network Security*, 2015: 326-334.

[64] Jiang Y, Zhang H H, Liu H, et al. System Reliability Calculation Based on the Run-Time Analysis of Ladder Program[J]. *IEEE Transactions on Industrial Electronics*, 2014, PP(99): 1.

[65] Garcia L, Mitsch S, Platzer A. HyPLC: Hybrid Programmable Logic Controller Program Translation for Verification[C]. *The 10th ACM/IEEE International Conference on Cyber-Physical Systems*, 2019: 47-56.

[66] Zhang M, Chen C Y, Kao B C, et al. Towards Automated Safety

Vetting of PLC Code in Real-World Plants[C]. *2019 IEEE Symposium on Security and Privacy*, 2019: 522-538.

[67] Cousineau D, Mentré D, Inoue H. Automated Deductive Verification for Ladder Programming[J]. *Electronic Proceedings in Theoretical Computer Science*, 2019, 310: 7-12.

[68] Guo S J, Wu M, Wang C. Symbolic Execution of Programmable Logic Controller Code[C]. *The 2017 11th Joint Meeting on Foundations of Software Engineering*, 2017: 326-336.

[69] Hao L, Shi J Q, Su T, et al. Automated Test Generation for IEC 61131-3 ST Programs via Dynamic Symbolic Execution[C]. *2019 International Symposium on Theoretical Aspects of Software Engineering*, 2019: 200-207.

[70] Hadžiosmanović D, Sommer R, Zambon E, et al. Through the Eye of the PLC: Semantic Security Monitoring for Industrial Processes[C]. *The 30th Annual Computer Security Applications Conference*, 2014, 12: 126-135.

[71] Goldenberg N, Wool A. Accurate Modeling of Modbus/TCP for Intrusion Detection in SCADA Systems[J]. *International Journal of Critical Infrastructure Protection*, 2013, 6(2): 63-75.

[72] Jardine W, Frey S, Green B, et al. SENAMI: Selective Non-Invasive Active Monitoring for ICS Intrusion Detection[C]. *The 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*, 2016, 10: 23-34.

[73] Yau K, Chow K P. Detecting Anomalous Programmable Logic Controller Events Using Machine Learning[C]. IFIP International Conference on Digital Forensics. Cham: Springer, 2017: 81-94.

[74] Garcia L, Zonouz S, Wei D, et al. Detecting PLC Control Corruption via On-Device Runtime Verification[C]. *2016 Resilience Week*, 2016: 67-72.

[75] Yang H, Cheng L, Chuah M C. Detecting Payload Attacks on Programmable Logic Controllers (PLCS)[C]. *2018 IEEE Conference on Communications and Network Security*, 2018: 1-9.

[76] Abbasi A, Holz T, Zambon E, et al. ECFI: Asynchronous Control Flow Integrity for Programmable Logic Controllers[C]. *The 33rd Annual Computer Security Applications Conference*, 2017, 12: 437-448.

[77] Conpot, L. Rist. http://conpot.org/, May. 2013.

[78] Lau S, Klick J, Arndt S, et al. POSTER: Towards Highly Interactive Honeypots for Industrial Control Systems[C]. *The 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016: 1823-1825.

[79] Xiao F, Chen E H, Xu Q. S7commTrace: A High Interactive Honeypot for Industrial Control System Based on S7 Protocol[C]. International Conference on Information and Communications Security. Cham: Springer, 2018: 412-423.

[80] Chan C F, Chow K P, Yiu S M, et al. Enhancing the Security and Forensic Capabilities of Programmable Logic Controllers[C]. IFIP International Conference on Digital Forensics. Cham: Springer, 2018: 351-367.

[81] Yau K, Chow K P, Yiu S M. A Forensic Logging System for Siemens Programmable Logic Controllers[C]. IFIP International Conference on Digital Forensics. Cham: Springer, 2018: 331-349.

孙越 于 2018 年在厦门大学电子信息工程专业获得学士学位。现在中国科学院大学网络空间安全专业攻读博士学位。研究领域为工业控制系统安全、物联网安全。研究兴趣包括: 二进制安全、软件自动化攻防。Email: sunyue0205@iie.ac.cn

宋站威 于 2017 年在北京工业大学计算机科学与技术专业获得硕士学位。现任中国科学院信息工程研究所助理研究员。研究领域为物联网安全, 工控安全。研究兴趣包括二进制分析, 模糊测试, 漏洞分析与利用。Email: songzhanwei@iie.ac.cn

陈曦 高级工程师、信息系统高级项目管理师, 现任工业控制系统安全可靠测评共性技术工业和信息化部重点实验室副主任、中国软件评测中心工业控制系统与人工智能测评工程技术中心副主任, 研究方向为工业控制系统质量和安全测评。Email: chenxib@cstc.org.cn

游建舟 于 2015 年在厦门大学电子信息工程专业获得学士学位。现在中国科学院大学通信与信息系统专业攻读博士学位。研究领域为工控安全、物联网安全。研究兴趣包括:工控安全、工控蜜罐设计、大数据分析。Email: youjianzhou@iie.ac.cn

黄文军 于 2010 年在广西师范大学电路与系统专业获得工学硕士学位, 现任中国科学院信息工程研究所工程师, 研究领域为工控安全、物联网安全。研究兴趣包括: 物联网硬件安全、工控入侵诱捕。Email: huangwenjun@iie.ac.cn

孙利民 于 1998 年在国防科学技术大学计算机体系结构专业获得工学博士学位。现任中国科学院信息工程研究所第四研究室研究员。物联网安全、工业控制系统安全。研究兴趣包括: 工控入侵诱捕、工控态势感知。Email: sunlimin@iie.ac.cn