PLC 攻防关键技术研究进展

徐 震[1], 周晓军[1,2], 王利明[1], 陈泽龙[1], 陈 凯[1], 闫振博[1,2],
张 伟[1,2], 陈 聪[1,2]

[1] 中国科学信息工程研究所信息安全重点实验室 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

摘要 震网病毒爆发之后, 工控系统开始逐渐成为攻击者的主要攻击目标之一。随着对工业控制系统不断的不断了解, 攻击者的攻击手段日益复杂化, 攻击手段更加复杂, 应用技术更加先进, 攻击手法更加多样。PLC 作为工业控制系统中重要的基础性控制设备, 其面临的信息安全问题值得重视。论文从攻防的角度, 首先对 PLC 的基本结构和工作原理进行了深入剖析, 分析其脆弱性; 然后对 PLC 攻击技术进行了分类, 并详细分析了各类攻击技术的攻击原理; 对国内外 PLC 安全防护技术领域的研究进行了概括性的总结和归纳; 最后给出了 PLC 信息安全的未来研究趋势及展望。

关键词 PLC 攻击; PLC 安全; 工业控制系统; 工控系统安全; 工控系统攻击; 控制系统攻击; 设备攻击
中图法分类号 TP309.2 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2019.05.04

Recent Advances in PLC Attack and Protection Technology

XU Zhen[1], ZHOU Xiaojun[1,2], WANG Liming[1], CHEN Zelong[1], CHEN Kai[1],
YAN Zhenbo[1,2], ZHANG Wei[1,2], CHEN Cong[1,2]

[1] State Key Laboratory of Information Security, 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

Abstract After the outbreak of Stuxnet, Industrial Control System (ICS) began to become one of the targets of attack. With the continuous understanding of the industrial control system, attackers are more sophisticated, using complex means and more advanced technology, to launch various attack. As an important basic control equipment in industrial control system, PLC (Programmable Logic Controller) information security issues are worthy of attention. From the perspective of both the attacker and defender, the paper first analyzes the basic architecture and working principle of the PLC and points out its vulnerabilities, then classifies the PLC attack, and analyzes the attack principle of all kinds of attack technology thoroughly; it gives a general summary of PLC information security research progress at home and abroad. Finally, future research trend and prospects for PLC information security are presented.

Key words PLC attack; PLC security; Industrial Control System ; Industrial Control System Security; Industrial Control System attack; Control System attack; Device attack

1 引言

随着工业化和信息化的不断融合, 越来越多的信息技术涌入到工业生产的环境当中, 使得工业生产的效率得到了极大的提升, 同时缩减了用于通信方面的线缆费用。但是, 在享受信息技术带来的便利的同时, 也引入了以前在工业领域从未面临的安全问题。

原有的工业生产过程中的“信息孤岛”不复存在, 越来越多的系统集成为一个大系统。加之攻击者对

通讯作者: 王利明, 博士, 博导, 正高级工程师, E-mail: wangliming@iie.ac.cn
本课题得到(1)天基资源网络化服务体系构建与在轨验证 课题 6: 天基信息安全共享与服务机制研究的支持, No. ZDRW-KT-2016-02-06; (2)北京市科学技术委员会(Beijing Municipal Science & Technology Commission)的课题“国家关键基础设施安全监管平台核心技术研究(Research on Core Technologies of national key infrastructure security super-vision platform)”, No. Z161100002616032; (3)国家重点研发计划基金资助项目(China National Key R&D Program) No. 2016QY06X1205 的资助。
收稿日期: 2018-1-2; 修改日期: 2018-3-22; 定稿日期; 2019-5-14

于工业生产过程了解的不断深入，工业控制系统也正在成为攻击者重点的攻击的目标之一。2010 年伊朗纳坦兹核电站遭受 Stuxnet(震网)[1]病毒攻击，打破了封闭系统绝对安全的谬误，导致核电站中约五分之一的离心机报废，极大推迟了伊朗的核进程。2011 年出现与震网非常类似的 duqu 木马[2]，duqu 攻击的目标主要是工控系统，用于盗取私密信息。2014 年，芬兰信息安全厂商 F－secure 曝光了一种专门针对 ICS / SCADA 系统的恶意软件 Havex[3]，它有能力禁用水电大坝、让核电站过载，已经有黑客利用它攻击了欧美能源行业工控系统。2015 年 12 月 23 日，乌克兰电力部门遭受到恶意代码攻击[4]，乌克兰新闻媒体 TSN 在 24 日报道称：“至少有三个电力区域被攻击，并于当地时间 15 时左右导致了数小时的停电事故”；“攻击者入侵了监控管理系统，超过一半的地区和部分伊万诺-弗兰科夫斯克地区断电几个小时。”

典型的工业网络如图 1 所示[13]:

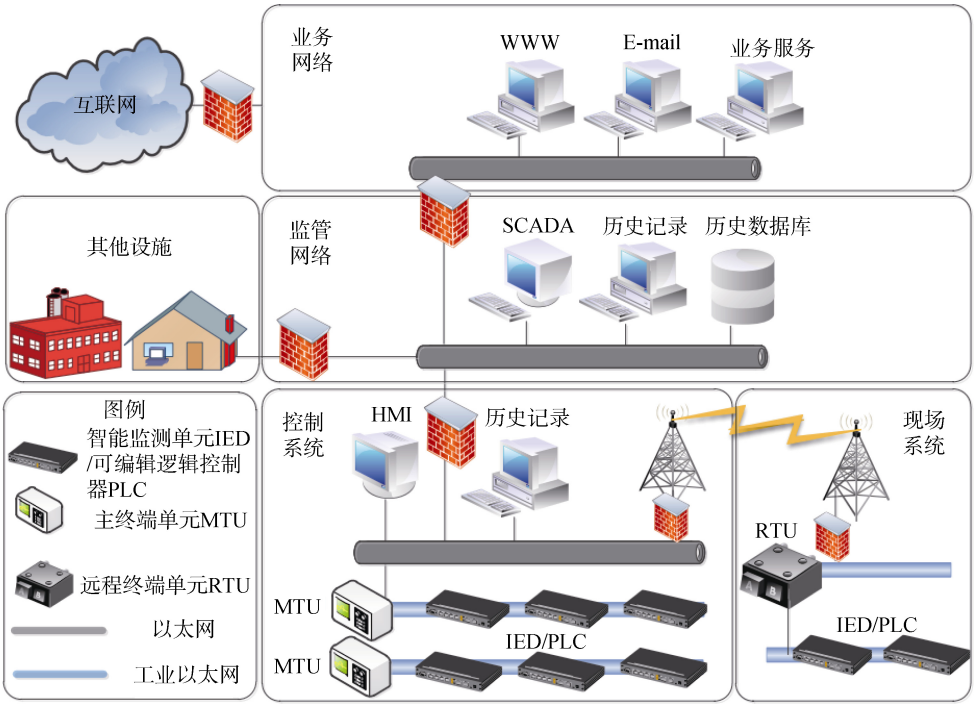


图 1　典型工业控制系统结构图

Figure 1　Typical Architecture of Industrial Control System

工业控制系统包含的组件主要有四个部分：业务网络、监管网络、控制系统和现场系统。其中业务网络主要包括网络服务器、邮件服务器等业务服务；监控网络主要包括 SCADA(Supervisory Control And Data Acquisition)系统、历史记录、历史数据库等；控制系统主要包括 HMI(Human Machine Interface) 、 PLC(Programmable Logic Controller) 、 RTU(Remote Terminal Unit)、IED(Intelligent Electronic Device)等，HMI 是 PLC/RTU/IED 的操作控制面板，功能是用来给操作员/工程师提供视图，实时显示现场系统状态。现场系统是整个工业控制系统的最底层系统，主要用来执行操作系统下发的控制指令等，包含的执行设备主要有 PLC/RTU/IED 等。而且，PLC 经常作为 RTU 和 IED 的替代设备而广泛的应用在工业控制系统之中。

据匡恩网络工业控制威胁情报中心统计，截至 2016 年底，中国境内暴露在互联网的工控设备高达 1143 个。从工控设备类型的角度看，此次在线监测采集到的数据中，PLC 为接入互联网数量最多的设备，这些设备的来源多为国外厂商，其安全性变得更加不可控。

根据匡恩网络的《2016 年工业控制网络安全态势报告》显示[5]，2000—2016 年公开工控漏洞影响产品统计数据，其中上位机、SCADA、PLC 历年累计已公开的工控系统漏洞数量分别达到了 470、214、141，成为了工控系统产品中最"脆弱"的产品组件。而这三类产品在我国电力、水利、污水处理、石油化工等国家关键基础设施和冶金、汽车、航空航天等制造业工业领域应用十分广泛，属于无法替代的关键角色。PLC 作为工业生产的重要基础性设备，面

临的安全问题更加突出, 也日益成为遭受攻击的目标。很多攻击都以 PLC 作为攻击的跳板, 然后侵入工业控制系统。比如 2010 年震网病毒攻击事件[1], 利用了两个 Siemens(西门子)PLC 编程软件 WinCC 的漏洞实施攻击; 2016 年出现了首个专门针对 PLC 的蠕虫病毒 PLC-Blaster[30], 该病毒可以在西门子 S7-1200 系列 PLC 之间进行扩散, 改编之后还可作用于其他系统, 而且还可以在代理链接中使用, 作为立足点以进入基础设施的网络系统。也就是说, 工控系统中一台 PLC 被感染, 就能很快扩散到整个系统。Abbasi 等[31]设计了一款专门针对 PLC I/O 针脚控制的 Rootkit, 将 PLC 的攻击推向了新的高度。黑帽大会(BlackHat)也多次展示了专门针对 PLC 的攻击[16, 20, 27, 30]。

本文从攻防两个角度, 对 PLC 的安全问题进行了深入剖析。第 2 节深入分析了 PLC 的工作原理, 并分析其脆弱性; 第 3 节对 PLC 攻击技术进行了分类整理, 并详细分析各类攻击技术的原理; 第 4 节综合分析国内外针对 PLC 安全防护的一些方法; 第 5 节辨析了 PLC 的功能安全(safety)和信息安全(security)的关系; 第 6 节给出了 PLC 信息安全的未来研究趋势及展望。

2 PLC 的基本结构和工作原理

PLC(Programmable Logic Controller)又称为可编程逻辑控制器, 是一种数字运算操作的电子系统, 专为在工业环境应用而设计的。它采用一类可编程的存储器, 用于其内部存储程序, 执行逻辑运算, 顺序控制, 定时, 计数与算术操作等面向用户的指令, 并通过数字或模拟式输入/输出控制各种类型的机械或生产过程, 是工业控制的核心部分。PLC 的产生是为了替代传统的继电器控制装置, 如今 PLC 不再局限于逻辑控制, 在运动控制、过程控制等领域也发挥着十分重要的作用[8-12]。

2.1 PLC 的基本结构

编程逻辑控制器实质是一种专用于工业控制的计算机, 其硬件结构基本上与微型计算机相同, 软件结构与传统的微型计算机存在较大差别。PLC 可靠性高、抗干扰能力强、编程简单、使用方便。

2.1.1 硬件结构

不同厂商生产的 PLC 在硬件组成上基本相同, 核心部件主要包括 CPU 模块、I/O 模块、编程器、电源等[11]。PLC 硬件结构如图 2 所示。下面依次对各部分进行介绍。

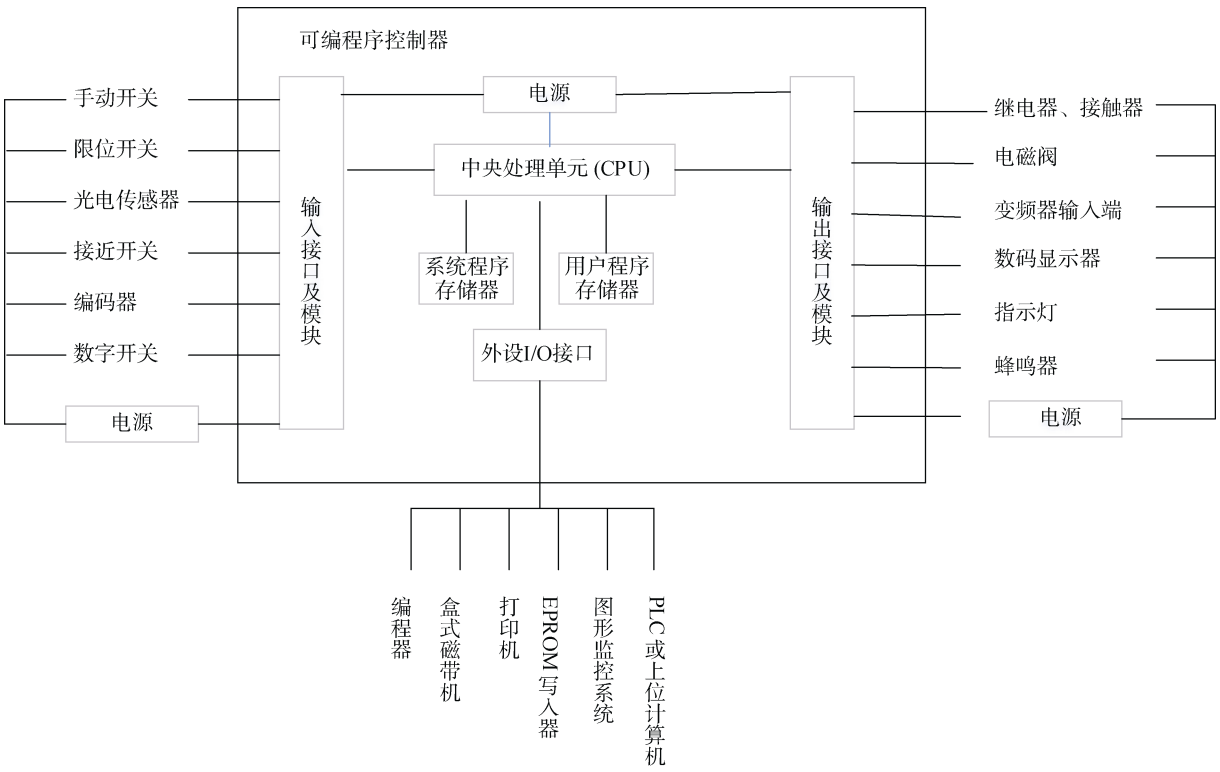


图 2 PLC 硬件结构

Figure 2 PLC Hardware Architecture

1. CPU 模块。包括两个部分: CPU(中央处理单元)和存储器。CPU 是 PLC 的核心部件, 由运算器和控制器组成。主要用于: 接收并存储从编程器输入的用户程序; 检查编程过程是否出错; 进行系统诊断; 解释并执行用户程序; 完成通信及外设的某些功能。

存储器有三种。(1)系统程序存储器。用于存放系统程序, 这些程序在 PLC 出厂前就已经固化到只读存储器 ROM 中。第一部分为系统管理程序; 第二部分为用户指令解释程序; 第三部分为标准程序模块与系统调用程序。(2)用户程序存储器。用于存储 PLC 用户的应用程序, 在调试阶段, 用户程序存放在读写存储器 RAM 中, 可由备用电池(一般为锂电池)保存 2~3 年。(3)工作数据存储器 。工作数据存储器用来存储工作数据, 即用户程序中使用的 ON/OFF 状态、数值数据等。

2. I/O 模块。输入/输出接口是 PLC 与外界连接的接口。输入接口用来接收和采集两种类型的输入信号, 一类是由按钮、选择开关、行程开关、继电器触点、接近开关、光电开关、数字拨码开关等的开关量输入信号。另一类是由电位器、测速发电机和各种变送器等来的模拟量输入信号。 输出接口用来连接被控对象中各种执行元件, 如接触器、电磁阀、指示灯、调节阀(模拟量)、调速装置(模拟量)等。

3. 编程器是 PLC 最重要的外围设备, 是 PLC 不可缺少的部分。编程器的作用是输入和编辑用户程序、调试程序和监控程序的执行过程。

编程器一般有两种类型: 简易编程器和图形编程器。简易编程器体积小, 便宜, 使用方便, 适合小型 PLC, 缺点是需联机编程; 图形编程器是指带有显示屏的编程器, 有液晶显示(LCD)和阴极射线式(CRT)两种, 可用指令语句编程, 也可用梯形图编程, 可联机编程也可脱机编程, 操作方便, 功能强大, 还可与打印机、绘图仪等设备相连, 但价格较高, 适用于大型 PLC。

随着 PLC 联网功能增强, 出现了第三种编程方式, 即计算机辅助编程。由于计算机的参与, 用 PLC 编程软件编程的工作效率和编程量远非前两种编程器可比, 因此, 越来越多的用户更愿意采用这种编程方式。

4. 电源模块。PLC 内部配有开关式稳压电源的电源模块, 用来将外部供电电源转变成供 PLC 内部的 CPU、存储器和 I/O 接口等电路工作所需要的直流电源。另外, 为防止在外部电源发生故障的情况下, PLC 内部程序和数据等重要信息的丢失, PLC 还带有锂电池作为后备电源。

2.1.2 软件结构

在可编程控制器中, PLC 的软件分为两大部分:

1. 系统监控程序: 用于控制可编程控制器本身的运行。主要由管理程序、用户指令解释程序和标准程序模块, 系统调用。

2. 用户程序: 它是由可编程控制器的使用者编制的, 用于控制被控装置的运行。

2.2 PLC 的工作原理

PLC 有两种工作状态, 即运行(RUN)状态和停止(STOP)状态。在运行状态, PLC 通过执行反映控制要求的用户程序来实现控制功能。为了使 PLC 的输出及时地响应随时可能变化的输入信号, 用户程序不是只执行一次, 而是反复不断地重复执行, 直到 PLC 停机或切换到 STOP 工作状态[14, 15]。

除了执行用户程序外, 每次循环过程中, PLC 不还要完成内部处理、通信处理等工作, 一次循环可分为 5 个阶段, 如图 3 所示。

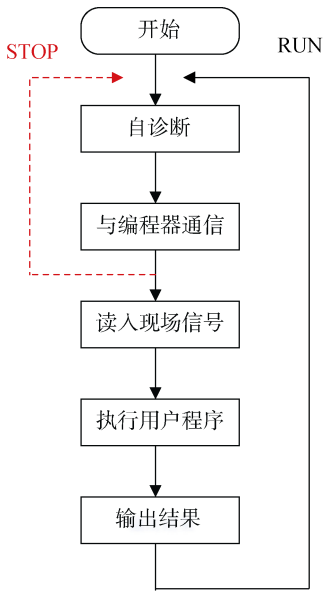


图 3 PLC 循环执行示意图

Figure 3 PLC Execution Loop

PLC 的这种周而复始的循环工作方式称为扫描工作方式。在工作状态下, 执行一次上图所示的扫描操作所需的时间称为扫描周期。其典型值为 1~100ms。PLC 扫描周期如图 4 所示。

当 PLC 投入运行后, 其工作过程一般分为三个阶段, 即输入采样、用户程序执行和输出刷新三个阶段。完成上述三个阶段称作一个扫描周期。在整个运行期间, 可编程逻辑控制器的 CPU 以一定的扫描速度重复执行上述三个阶段。

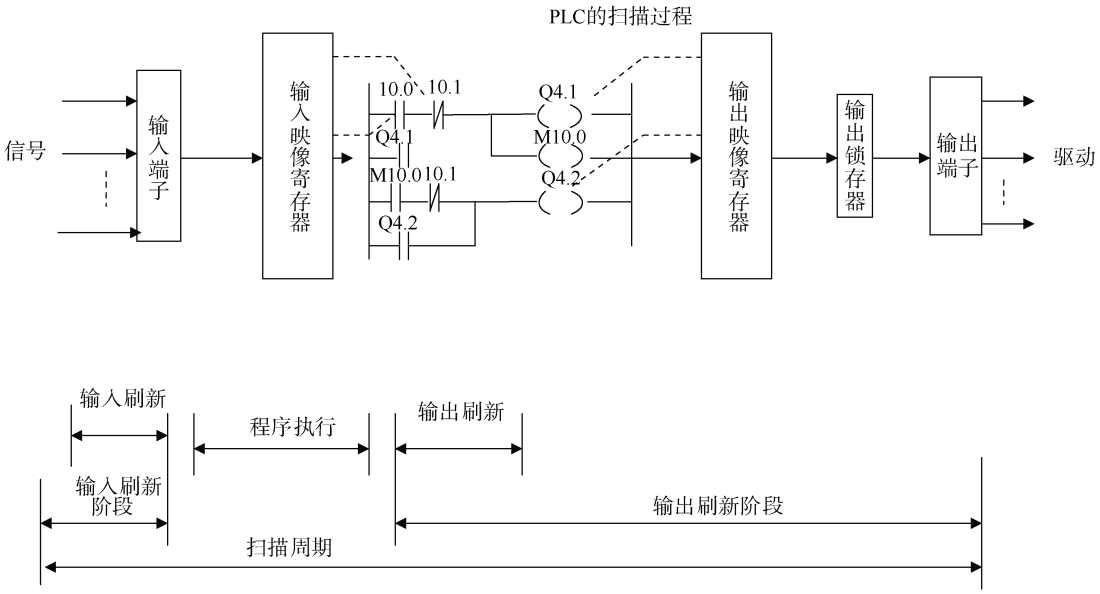


图 4　PLC 扫描周期

Figure 4　PLC Scan Cycle

1. 输入采样阶段

在输入采样阶段, 可编程逻辑控制器以扫描方式依次地读入所有输入状态和数据, 并将它们存入 I/O 映象区中的相应的单元内。输入采样结束后, 转入用户程序执行和输出刷新阶段。在这两个阶段中, 即使输入状态和数据发生变化, I/O 映象区中的相应单元的状态和数据也不会改变。因此, 如果输入是脉冲信号, 则该脉冲信号的宽度必须大于一个扫描周期, 才能保证在任何情况下, 该输入均能被读入。

2. 用户程序执行阶段

在用户程序执行阶段, 可编程逻辑控制器总是按由上而下的顺序依次地扫描用户程序(梯形图)。在扫描每一条梯形图时, 又总是先扫描梯形图左边的由各触点构成的控制线路, 并按先左后右、先上后下的顺序对由触点构成的控制线路进行逻辑运算, 然后根据逻辑运算的结果, 刷新该逻辑线圈在系统 RAM 存储区中对应位的状态; 或者刷新该输出线圈在 I/O 映象区中对应位的状态; 或者确定是否要执行该梯形图所规定的特殊功能指令。

即, 在用户程序执行过程中, 只有输入点在 I/O 映象区内的状态和数据不会发生变化, 而其他输出点和软设备在 I/O 映象区或系统 RAM 存储区内的状态和数据都有可能发生变化, 而且排在上面的梯形图, 其程序执行结果会对排在下面的凡是用到这些线圈或数据的梯形图起作用; 相反, 排在下面的梯形图, 其被刷新的逻辑线圈的状态或数据只能到下一个扫描周期才能对排在其上面的程序起作用。

在程序执行的过程中如果使用立即 I/O 指令则可以直接存取 I/O 点。即使用 I/O 指令的话, 输入过程影像寄存器的值不会被更新, 程序直接从 I/O 模块取值, 输出过程影像寄存器会被立即更新, 这跟立即输入有些区别。

3. 输出刷新阶段

当扫描用户程序结束后, 可编程逻辑控制器就进入输出刷新阶段。在此期间, CPU 按照 I/O 映象区内对应的状态和数据刷新所有的输出锁存电路, 再经输出电路驱动相应的外设。这时, 才是可编程逻辑控制器的真正输出。

2.3　PLC 脆弱性分析

PLC 可以看成是为工业控制环境而设计的专用计算机, 但是, 从其自身设计以及运行上都存在诸多安全问题。

首先, PLC 自身设计上存在缺陷。主要体现在以下 4 个方面:

1. 扫描式的工作方式。前文已经介绍过, PLC 的扫描周期为 1~100ms。也就是说, 在扫描周期结束之前无法进行数据更新。PLC 输入信号时间若小于反应时间, 则有误读的可能性。每次程序执行后与下一次程序执行前, 输出与输入状态会被更新一次, 因此称此种运作方式为输出输入端“程序结束再生”。但是这就给攻击者留下了足够的时间实施攻击, 后文在攻击技术进行详细介绍。

2. 内存容量小。PLC 内存有用户及系统两大部分。用户内存主要用以存储用户程序, 个别的还将其中的一部分划为系统所用。系统内存是与 CPU 配置在一起的。CPU 既要具备访问这些内存的能力, 还应

提供相应的存储介质。用户内存大小与可存储的用户程序量有关。内存大, 可存储的程序量大, 也就可进行更为复杂的控制。从发展趋势看, 内存容量总是在不断增大。大型 PLC 的内存容量可达几十 k, 以至于一百多 k。内存较小由多方面的原因造成的。(1)PLC 的控制逻辑相对简单, 不需要较大的存储空间。(2)PLC 特有的扫描式工作方式决定了不可能允许较大程序的存在, 也就不需要较大的内存。(3)PLC 使用的内存是工业环境专用的, 有些甚至需要在高温等严酷环境下工作, 造价较高。

3. 使用的操作系统存在较大安全隐患。由于工业控制现场实时性的要求, PLC 采用的操作系统大多是经过裁剪的实时操作系统(RTOS), 比如 Linux RT、QNX、Lynx、VxWorks 等。嵌入式系统历来都会使用大量专有组件, 并没有与其他系统共享太多共同电路, 这意味着当发现漏洞时, 由于成本或资源限制, 漏洞修复难度较大。而且, PLC 的系统更新一般只能由生产厂商完成, 因此其打补丁的过程就更加缓慢。

4. 采用的通信协议缺乏安全机制。一般的工业协议都经历长时间的演变与积累, 协议在设计之初就没有考虑加密、认证等在当今看来保障用户安全的必要认证条件。其次, 工控协议的特性第面向命令、面向功能、轮询应答式, 攻击者只要掌握了协议构造方式, 并接入到工控网络之中, 并可以对 PLC 的任意数据进行篡改。最后, 工控协议中包含了大量的命令字, 如读取数据、写入指令等, 其中一部分高级或协议约定的自定义功能往往会给用户安全带来更多的威胁, 如 Modbus 协议的从机诊断命令将会造成从机 PLC 切换到侦听模式; CIP 协议的某些命令字会造成从机 PLC 直接重启; S7 协议的 Stop CPU 工鞲将会导致 PLC 程序运行停止。后文我们将会详细介绍如何利用协议漏洞来对 PLC 实施攻击。

其次, 在 PLC 的实际运行中, 也存在一些制约 PLC 安全的问题。主要体现在以下 4 个方面:

1. PLC 使用专用的软硬件。PLC 的生产厂商往往采用专用的软硬件, 这样就给安全加固带来了极大的问题。外部的安全人员需要首先对 PLC 的软硬件进行充分的了解之后才能有的放矢, 但是这些资料的获取极其困难。也就是说, PLC 的安全只能由 PLC 生产厂商实施, 无法积聚信息安全领域的专家知识。

2. 实际的工业控制过程实时性较高。在工业控制系统中, PLC 的命令需要实时传输, 尤其是在关键控制过程, 命令的传输需要毫秒级甚至是微秒级。PLC 较高的实时性和较低的 CPU 运算能力决定了其无法使用强加密技术, 这就导致了 PLC 通信过程明文进行传输又无法进行有效防护的窘境。

3. PLC 的使用寿命长。目前 PLC 的使用寿命一般都在 10 年以上甚至更久。因此在一个工业控制生产环境之中, 存在大量的遗留设备, 给 PLC 的安全措施带了极大的难题。同时, PLC 由于其使用寿命长, 软件更新缓慢, 而且大量存在新老型号混合使用的情况。

4. PLC 之间以及 PLC 与业务网络之间的互连性正不断增强。传统的工业环境是采用物理隔离的方式进行运行, 也就是说, 工业生产环境不与外部环境进行交互。这也就给工业界造成了一种错觉, 认为只要不连接互联网就是安全的。但是震网病毒打破了这一传统的观念。同时, 随着工业 4.0 的不断推进, 工业生产系统与外界的交互不断增强, 给生产设备的安全带来了极大的安全挑战。

最后, 目前 PLC 安全还存在缺乏完善的信息安全标准体系、急需大量既懂得信息安全有熟悉自动控制的综合性人才。

3 PLC 攻击关键技术分类

PLC 是关键基础设施中的基础性设备, 其安全涉及到整个系统的安全稳定运行。但是, 随着两化融合的不断加深以及工业 4.0 的推进, 工业控制系统在提高信息化水平的同时, 其信息安全问题也日益突出。PLC 遭受黑客攻击的途径也日益翻新, 各种木马和病毒变体数量不断攀升, 威胁工业控制系统的稳定运行和人员生命财产安全。

3.1 PLC 遭受攻击的途径

随着技术的不断进步, PLC 在向着智能化的方向发展, 接口数量和类型越来越多, 功能也日益丰富。目前的 PLC 一般都是基于裁剪后的嵌入式系统, 同时将原来位于串行链路上的通信协议转移到 TCP/IP 之上, 为黑客实施攻击提供了便捷的途径。

3.1.1 通过嵌入式系统漏洞实施攻击

根据 2.3 章节 PLC 脆弱性分析可知, PLC 采用的大多是经过裁剪的实时操作系统(RTOS), 比如 Linux RT、QNX、Lynx、VxWorks 等。这些操作系统广泛的应用于通信、军事、航空、航天等高精尖技术及实时性要求较高的领域中。但是其安全问题不容忽视。常见的 PLC 使用的操作系统如表 1 所示。

表 1 常用 PLC 的操作系统

Table 1 Common PLC operating system

PLC	*Operating System*
Allen-Bradley PLC5	Microware OS-9
Allen-Bradley ControlLogix	VxWorks
Emerson DeltaV	VxWorks
Schneider Modicon Quantum	VxWorks
Yokogawa FA-M3	Linux
Wago 750	Linux
PLC reference platform	QNX Neutrino
Siemens SIMATIC WinAC RTX	Microsoft Windows

以 VxWorks 为例, CVE 公布的关于 VxWorks 相关漏洞统计如表 2 所示。

表 2 VxWorks 系统漏洞统计表

Table 2 VxWorks System Vulnerability Statistics

年份	漏洞数量	DoS 类型漏洞	代码执行
2008	1	1	
2010	4		
2013	6	6	1
2015	1		
漏洞数合计	12	7	1
百分比(%)		58.3	8.3

Beresford 在 Black Hat2011 上指出[16], Simatic PLC 运行在 x86 Linux 系统之上, 那就意味着如果插入一段载荷, 就可以对 shell 进行爆破并连接到该设备。尤其需要注意的是 PLC 上运行的所有程序都是以 root 权限运行的, 一旦被攻击者攻入, 后果非常严重。如图 5 所示:

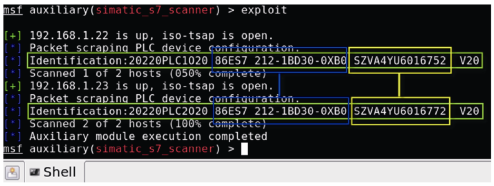


图 5 对 shell 系统的爆破

Figure 5 Attack on Shell System

西门子、施耐德的多款 PLC 设备软件搭载在 VxWorks 系统上运行, wdbrpc 是 VxWorks 的远程调试端口, 以 UDP 方式进行通信, 端口号为 17185。该协议基于 sun-rpc, 提供的服务主要用于支持系统远程通过集成开发环境 Tornado 交互(如图 6)。根据灯塔实验室公布的资料, 黑客可以通过 wdbrpc 协议 dump 全部内存空间数据, 找到内存中的所有 ftp、telnet 登录密码, 进一步可以实现的攻击有: 篡改 bootline 绕过登录验证、dump 内存数据从中抓取登录密码等。通过攻击嵌入式实时操作系统进而控制 PLC 的正常运行。

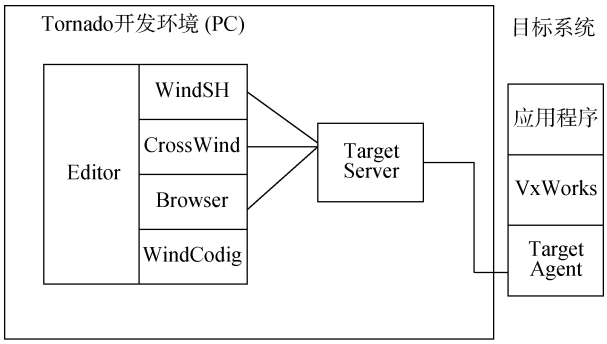


图 6 Tornado 开发环境与 VxWorks 系统图

Figure 6 Tornado Development Environment and VxWorks System

3.1.2 通过 PLC 通信协议漏洞实施攻击

分析任何协议时, 区分安全问题的种类是非常有用的: 一类是协议自身的设计和描述引起的, 另一类是协议的不正确实现引起的。发现协议设计和描述中的安全问题较协议实现中的安全问题可能要容易的多, 也可能要难得多, 但修复源于不正确的协议实现的安全问题相对要容易一些。

随着时代的发展, 厂级监控的实时性、可靠性需求增高, 工业通信总线通讯速率的不断提升, 从 RS232/485 到工业以太网再到工业实时以太网, 工控网络中大量引入了以太网, 并且使用 TCP/IP 或 ISO 标准封装后进行传输, 因为一般的工控协议都经历了长时间的演变与积累, 协议在设计之初都没有考虑加密、认证等在当今看来保障用户安全的必要认证条件, 如第一个现场总线协议 Modbus 由莫迪康与 1979 年提出, 所以我们常见的工控网络协议的安全性一直都不高。加上工控协议的特性是面向命令、面向功能、轮询应答式, 攻击者只需要掌握协议构造方式, 并接入到了工控网络中, 便可以通过协议对目标设备的任意数据进行篡改。

一般常见工控协议中包含了大量的命令字, 如读取、写入数据等, 然而其中一部分高级或协议约定的自定义功能往往会给用户安全带来更多的威胁, 如 Modbus 协议的从机诊断命令将会造成从机设备切换到侦听模式、CIP 协议某些命令字还能导致设备直接重启、S7 协议的 STOP CPU 功能将会导致 PLC 程序运行停止, 在大多数的情况下用户在上位机进行组态时仅会使用协议的某些读取数据功能和固定范围、固定地址的写数据功能, 而协议栈上更多的功能则不会应用于系统集成中。

Langner 指出[19]不需要控制系统内部知识, 不需要编程技巧就可以实现一次攻击。利用 S7 协议漏洞, 注入代码到组态 OB1(相当于 main 函数)之前, 这样

PLC 在每次扫描之前都会首先执行恶意代码, 并可以通过调用 BEC(block end condition)指令, 随时终止合法代码的运行。就 Stuxnet 而言, 终止条件是基于时间和工业过程。Beresford[16]详细介绍了如何利用 Siemens 的 S7 协议漏洞实施攻击。首先抓取工程师站和 PLC 之间通信的数据包, 然后分离出 Client 端的 TCP 数据流, 基于这些数据包, 重新构建新的数据包, 最后将这些数据包重放给 PLC, 从而获取 S7-1200 型号 PLC 的内存读写权限, 找到代码路径(code path)、源代码(source code)及新的漏洞, 从而可以爆破获取 S7 密码(过程如图 7 所示),

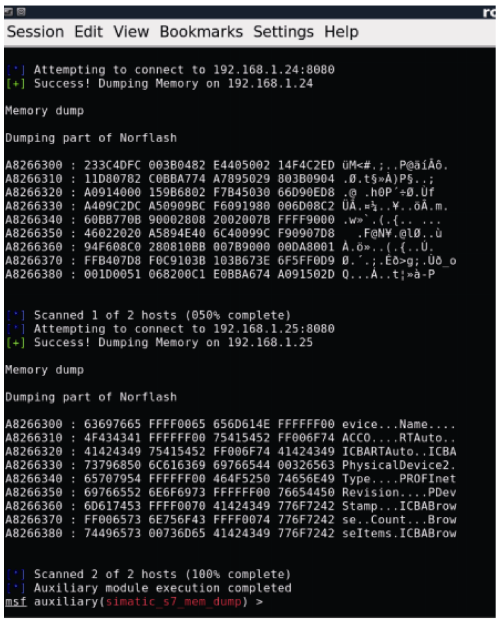


图 7 S7-1200 PLC 内存分析过程

Figure 7 S7-1200 PLC Memory Analysis Process

Meixell[20]在 2013 年 USA BlackHat 指出, 简单的串口协议(比如 Modbus 和 DNP3)已经被包含在 IP 数据报内, 攻击者仅仅构造一个基于 IP 的控制数据包并发送给 PLC 就可以造成严重的后果。以 Modbus 协议为例, 其数据包结构及常用主要功能码如下表 3 所示。利用功能码 0x05 就可以将所有的寄存器置 1, 打开所有的阀门。

Tzokatziou[21]指出, 由于 PLC 通信协议是明文传输, 而且对于通信对象没有认证过程。

因此攻击者可以利用 CoDeSys 系统, 直接和 PLC 进行连接, 捕获两者之间的通信的数据包, 然后直接给 PLC 发送篡改后的控制指令, 达到任意启停 PLC 的操作。

3.1.3 通过 PLC 软件漏洞实施攻击

PLC 的软件系统包括系统监控软件和用户组态软件, 前者用于监视控制器本身的运行, 后者用于编写用户程序。以西门子(Siemens)PLC 为例, STEP 7 编程软件用于 PLC 的编程、参数设置和在线调试, 而 WinCC 则主要用于过程监视。

表 3 Modbus 数据包结构及主要功能码

Table 3 Modbus Packet Structure and Main Function Code

Transaction ID		Protocol ID	Data Length	Unit ID	Function Code	Data
MSB	LSB	0x00 0x00	# of bytes	0xFF (Typ)	See Table	DATA

Function Code	Function
0x01	Read Coil
0x02	Read Discrete Input
0x03	Read Holding Register
0x04	Read Input Register
0x05	Write Single Coil
0x06	Write Single Register

典型的攻击案例是 2010 年的“震网”病毒攻击伊朗核电站事件[1, 2]。“震网”病毒除了利用 windows 操作系统的 4 个 0-day 漏洞, 还利用了西门子 WinCC 中的两个漏洞(1)WinCC 系统中存在一个硬编码漏洞, 保存了对访问数据的默认账户名和密码, Stuxnet 利用这一漏洞尝试访问该系统的 SQL 数据库; (2)在 WinCC 需要使用的 Step7 工程中, 打开工程文件时, 存在 DLL 加载策略上的缺陷, 从而导致一种类似于“DLL 预加载攻击”的利用方式。然后 Stuxnet 通过使用自身的 s7otbxsx.dll 替换 Step7 软件中的 s7otbxsx.dll, 实现对一些查询、读取函数的额 Hook。

其他的攻击途径包括攻击人机交互界面(HMI), 使得操作员失去视图。典型的攻击案例是 2015 年的乌克兰电网攻击事件[4]。攻击者取得工作站节点的控制权, 获取与操作员一致的操作界面和操作权限, 通过远程控制对 PLC 进行开关控制或改变运行参数, 从而引起电网故障或者断点。Meixell[20] 指出, EWS(*Engineering Workstations*)上面的 PLC 编程和控制软件一旦遭受攻击, 攻击者就可以修改 PLC 上的逻辑控制, 移除组态中内嵌的安全方案, 强制 I/O 状态, 攻击过程如图 8 所示。

灯塔实验室指出, Unity Pro 是施耐德系列 PLC 的编程软件, Unity Pro 附带的 OSLoader 软件可以完成 PLC 的操作系统固件升级。OSLoader 登录设备后会尝试远程读取文件系统, 这样即可实现远程上传下载, 攻击者可以通过替换固件的方式轻松让 PLC 宕机。

3.1.4 通过 PLC 互连实施攻击

当前的工业控制网络朝着“一网到底”的方向发展，工业控制系统横向和纵向连接更加紧密：业务层可以直接访问控制层的数据，甚至对设备进行控制；同一层次内的设备由于相互间的协作工作而联结在一起。对于 PLC 而言，由于 PLC 主要用于过程控制，而生产流程往往由诸多控制过程组成，因此需要多个 PLC 协同工作，共同完成某项生产任务。

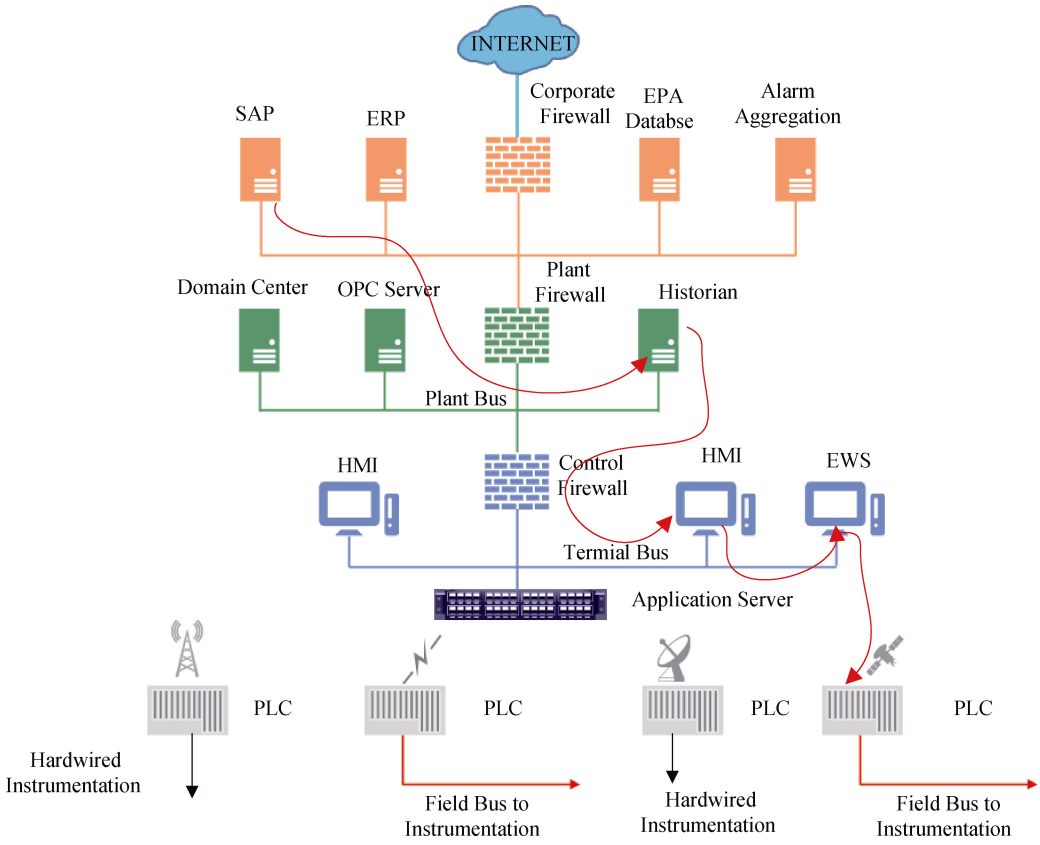


图 8 利用人机交互界面攻击 PLC 过程

Figure 8 Attack on PLC by Utilizing HMI

Radvanovsky 指出[24]，美国启动的 SHINE (SHodan Intelligence Extraction)项目是为了提取关于可以从互联网访问的 SCADA 和 ICS 设备的信息(尤其是 PLC 和 RTU)。SHODAN 搜索引擎是通过搜索常用的 TCP/UDP 端口来工作的，如表 4 所示。

Newman[26]指出，监狱中控制室值班通过在线浏览图片和电影引入病毒和蠕虫。一些监狱为犯人提供上网服务，虽然不和监狱控制和监视系统直接连接，但是也是一个可攻入的脆弱点。还要监狱的巡逻车，使用的是无线信号，需要连接监狱网络上传数据，也是一个攻击入口。

Klick[27]在 2015 USABlackHat 上指出，PLC 缺乏安全机制，通常可以上传代码到这些面向互联网的 PLC，利用这些 PLC 作为网关，渗透生产网络，甚至是公司 IT 网络。Kclik 利用 PLC 编程语言 STL 编写端口扫描器和 SOCKS 代理，然后利用感染的 PLC 去扫描本地网络，并将其作为网关，从而连接与其相连的其他 PLC 甚至渗透到公司业务网络，如图 9 所示。

表 4 常用协议及端口对照表

Table 4 Common Protocol and Port Pairs

协议	端口
Siemens S7	102
Modbus/Tcp	502
IEC 60870-5-104	2404
DNP3	20000
EtherNet/IP	44818
CodeSYs	2455
BACnet	47808
SSH	22
Telnet	23
FTP	21
SNMP	161

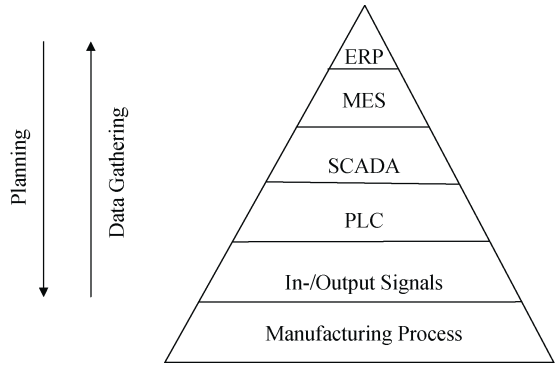


图 9　公司内系统层级图

Figure 9　System layers in Corporation

McLaughlin[28]设计了一款针对 PLC 的恶意软件, 能够生成动态数据包载荷, 攻击者使用这个工具, 可以不用提前对控制系统有先验知识就可以实施攻击, 极大的降低了攻击 PLC 的门槛。首先利用生成的载荷感染一个到多个主机, 然后进行工业过程分析, 接着对二进制文件进行解码, 最后对生成的载荷进行裁剪, 上传到 PLC 并运行, 具体过程如图 10 所示。

McLaughlin 在另一篇文章[29]中开发了一种自动生成 PLC 载荷的工具—Sabot, 自动识别 PLC 逻辑控制, 并生成恶意的 PLC 代码, 具体过程如图 11 所示。

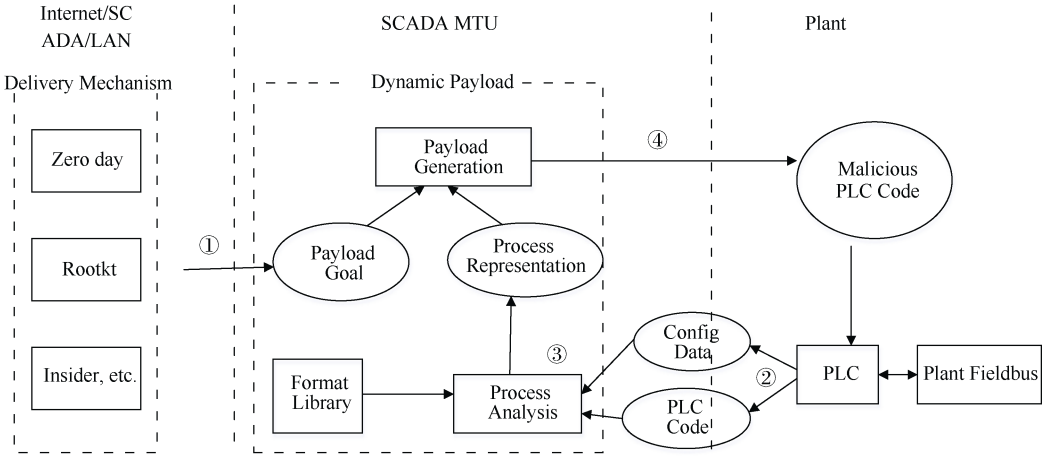


图 10　动态生成恶意载荷过程

Figure 10　Processes of Dynamically Generating Malicious Payloads

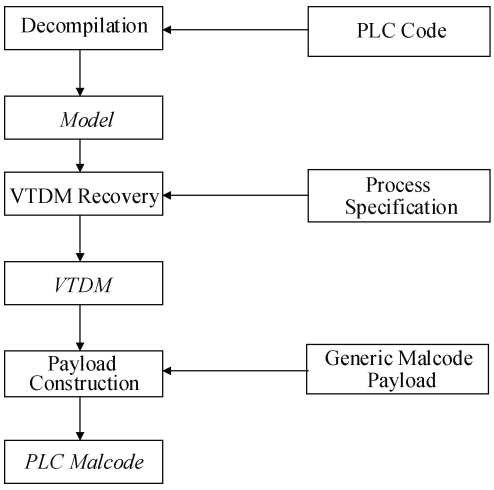


图 11　Sabot 攻击过程示意图

Figure 11　Attack Process of Sabot

Spenneberg[30]等在 2016 Asia BlackHat 上西门子 SIMATICA S7-1200 为例, 展示了一款专门在 PLC 上存活的蠕虫。此蠕虫不需要依赖于 PC 电脑去扩散, 仅仅活跃并运行于 PLC 中, 通过网络扫描来发现新的目标(PLC), 然后攻击这些目标并将复制自身到新的 PLC 中, 而且受感染的 PLC 主程序不会发生任何改变。从而可以做到目标发现, 携带恶意载荷等攻击手段。而且, 清除这些蠕虫非常困难, 目前只能通过恢复出厂设置或者复写蠕虫所在的功能块(Function Block)。感染过程如图 12 所示。

代码的执行过程如图 13 所示。

3.2　PLC 遭受攻击的种类

PLC 遭受攻击的种类按照攻击的难易程度可以分为干扰性攻击、组态攻击和固件攻击。

三类攻击的描述如表 5 所示。

3.2.1　干扰性攻击

干扰性攻击主要用于耗尽 PLC 的资源, 比如网络带宽、CPU 计算资源等, 从而使得 PLC 对正常的请求无法及时作出回应。华北电力大学举办的工控大赛上, 初级攻击层次使用的就是 DDoS 的方式, 使得 PLC 的通信模块出现拒绝服务, 导致用户界面显

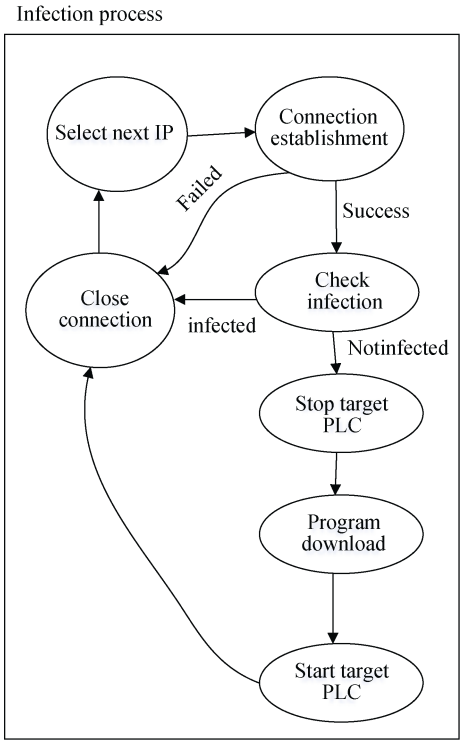


图 12　蠕虫感染 PLC 过程

Figure 12　Worm Infection Process on PLC

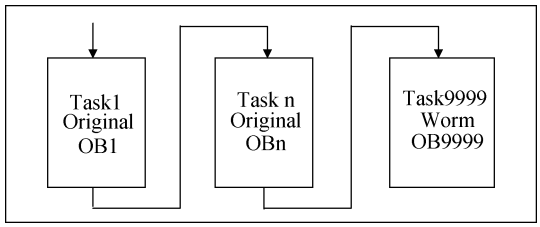


图 13　蠕虫恶意代码执行过程

Figure 13　Malicious Code Execution Process

表 5　攻击的分类、描述及危害

Table 5　Classification, Description and Effects of Attacks on PLC

攻击分类	描述	可能产生的后果
干扰性攻击	对 PLC 的正常运行造成干扰, 但不改变 PLC 的组态。	扰乱 PLC 和受控设备的正常通信, 导致通讯中断或异常, 使得 PLC 控制指令出现短时异常。
组态攻击	控制 PLC 上位机, 并修改 PLC 的组态。	对 PLC 实施精准攻击, 对整个控制系统造成危害极大。
固件攻击	对 PLC 的固件进行修改, 同时在上位机上显示正常, 实施隐蔽攻击。	对 PLC 的危害最大, 而且不易察觉。

示异常。另外也可以利用 PLC 通信缺乏认证和加密的缺陷, 直接修改 PLC 的控制点位的数据, 但是由于 PLC 自身的扫描周期比较短, 因此, 达到此种攻击需要攻击者使用的攻击机性能比较强大, 发包速度比较快。而且这种攻击在停止后, 一般 PLC 能恢复正常的运转。

Newman 等[26]指出, 在监狱中用来控制牢房和监狱大门的 PLC 可以远程打开或者锁死。一旦 PLC 被攻陷, 就可以操作 PLC 控制的所有设施的物理状态: 抑制发送给 PLC 或者由 PLC 发出的告警信息。McLaughlin 等[28]实现了一款 PLC 攻击软件, 对 PLC 的通信和控制过程进行分析, 并生成动态的 PLC 通信载荷。不去修改 PLC 的组态, 但是会对 PLC 的正常运行造成极大的危害, 比如: 打开变电站中的所有断路器。Abbasi 等[31]开发出一款无法被检测到的 PLC Rootkit, 可能会比 Stuxnet 更加危险, 因为 Stuxnet 的设计目标在于指向 Windows 架构之上的应用软件, 而 PLC Rootkit 立足于更加底层的系统。该恶意软件会干扰 PLC 运行时与逻辑同 I/O 外设之间的连接(如图 14), 驻留在 PLC 组件的动态内存之中, 且操纵相关的 I/O 及 PLC 流程(如图 15), 同时影响 PLC 进行通信交互以及处理物理流程控制的 I/O 模块。

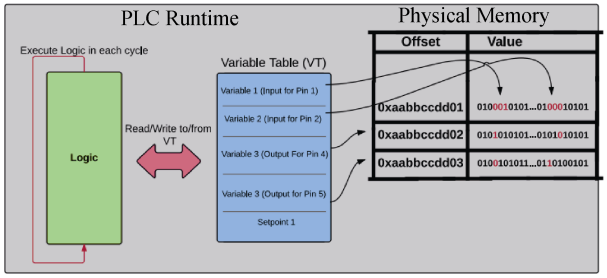


图 14　PLC RootKit 攻击过程

Figure 14　Attack Process of PLC RootKit

Tzokatziou 等[21]指出, 可以通过 HID(human interface devices)对 PLC 的正常运行实施干扰。利用 ABB PM564 PLC 通信协议无加密、无认证的特性, 使用 Codesys 系统和 PLC 建立通信并分析数据包指令, 人为构造数据包, 达到任意启停 PLC 的目的。

目前出现了专门针对 PLC 的勒索软件[78]。也就是对工业控制系统进行加密, 只有在支付一定数额的赎金之后才能获取解密密钥, 这种攻击对 PLC 的组态并未破坏, 但是在遭受攻击期间无法正常工作。

3.2.2　组态攻击

组态攻击是对 PLC 的组态进行攻击, 破坏 PLC 的控制逻辑, 从而达到精准实施攻击的目标。Kclik 等[27]开发了 PLCinject 工具, 可以对 PLC 注入篡改后

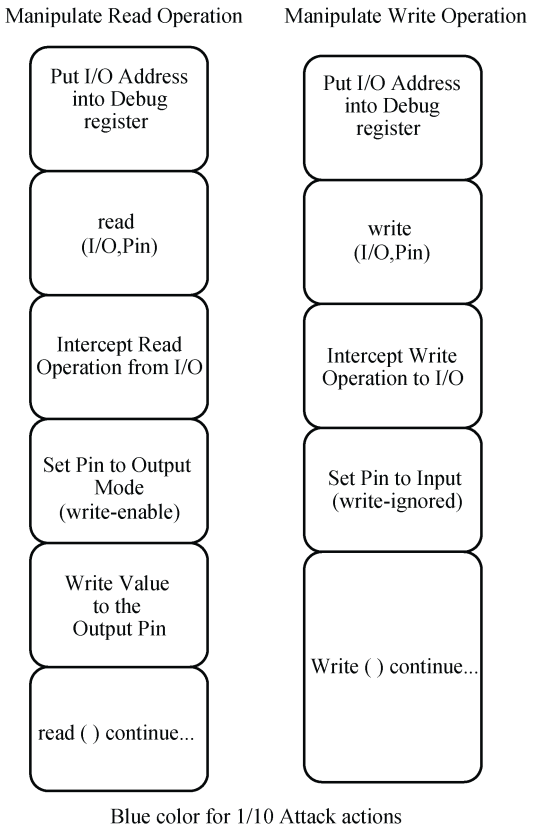


图 15　I/O 针脚复用攻击示意图

Figure 15　I/O Pin Multiplexing Attack Example

的组态，使得 PLC 的逻辑执行跳转到攻击者指定的功能块上，它利用 PLC 编程语言 STL 编写端口扫描器和 SOCKS 代理，首先下载 PLC 组态 OB1，在组态开始执行前加入 CALL 指令，调用恶意功能块 FC666，启动 SNMP 扫描器，然后运行以下 7 步骤:

1. 获取本地 IP 地址和子网
2. 计算 IP 地址
3. 建立 UDP 连接
4. 发送 SNMP 请求数据包
5. 接收 SNMP 回复数据包
6. 将回复数据包保存在 DB 中
7. 停止扫描，断开 UDP 连接

组态注入过程如图 16 所示。

Langner 等[19] 利用编译后的十四字节序列，注入到原来 PLC 合法的组态 OB1 之前，设置终止条件，随时都可以终止 PLC 的运行。14 字节的代码如图 11 所示。

7E 63 00 0C 38 07 11 12 25 00 39 A0 05 00

McLaughlin 等[29]通过连接到 Internet 的 PLC，分析 PLC 的逻辑控制，并自动生成恶意代码，编译后注入 PLC，从而更改 PLC 的组态。

但是组态攻击要达到比较准确的目标，往往需要对 PLC 的组态进行分析，找到攻击点的准确定位。典型的如 Stuxnet[1]。

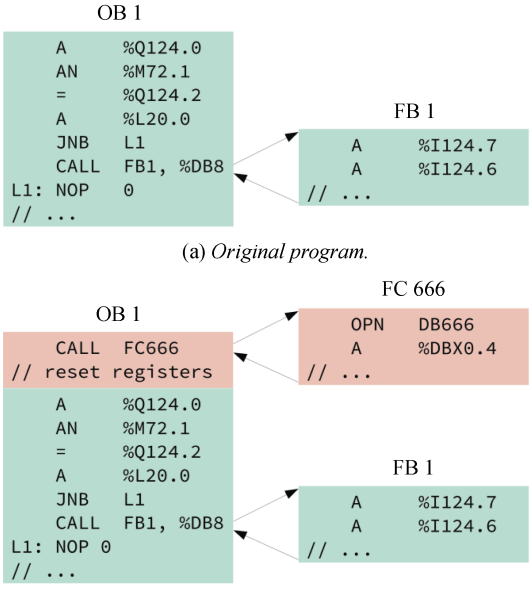


图 16　PLC 组态注入过程

Figure 16　PLC Configuration Injection Process

3.2.3　固件攻击

原来的攻击都针对于 PLC 的上层系统，比如针对人机交互界面(HMI)和网络通信协议(比如 Modbus 协议)。即使是攻击手段较为复杂的 Stuxnet，也是针对 Siemens PLC 的编程软件，而不是底层的现场设备代码。

在 PLC 架构中，固件充当了操作系统的角色，提供了诸多服务，比如通过 web 服务器远程访问、远程固件升级等。这些功能给终端用户的操作提供了极大的便利，但是同时也给攻击者以可乘之机，如图 17 所示[17]。

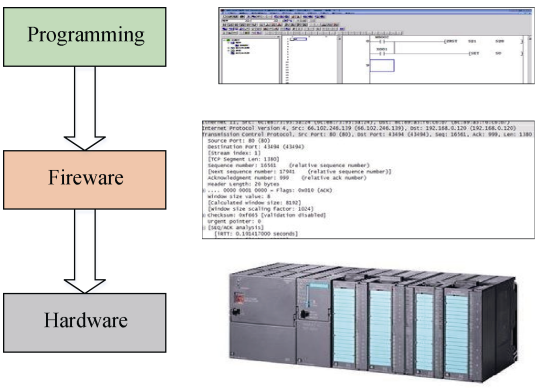


图 17　PLC 组态、固件、硬件模型

Figure 17　PLC Configuration, Firmware, Hardware Model

针对 PLC 的固件攻击是目前最为隐蔽的攻击方式, 实现起来比较复杂, 一般分三步进行:

(1)对 PLC 的固件进行反汇编, 确定各个参数的寄存器地址, 匹配已知的设备功能;

(2)注入恶意指令、修改跳转指令或者修改寄存器地址, 且一般不影响上层组态软件的稳定性;

(3)对固件进行重新打包并重新下载到 PLC 中。

Schuett 等[17]首先通过逆向工程, 分析 PLC 的固件映像的指令集, 找到固件的存储区域和执行路径, 然后修改固件中的服务和控制指令, 从而实施远程攻击: 设定时间终止 PLC 运行、收到控制信号终止 PLC 运行、对固件做永久修改使得操作员无法重新使用 PLC。PLC 固件的一个缺陷就是信任其内在的固件验证过程, 这个过程是依赖于 CRC 校验机制。CRC 校验可以用来验证固件是否受到破坏, 但是无法检测到恶意篡改, 如图 18 显示了对固件诊断例程的修改。

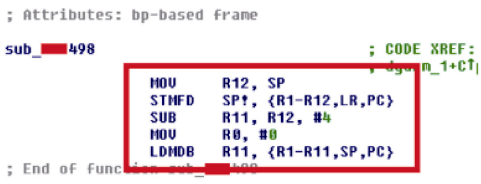


图 18 对 PLC 固件诊断例程修改过程

Figure 18 Modification Process for PLC Firmware Diagnostic Routine

Beresford 等[16]实现了针对 Siemens Simatic S7 PLC 的固件攻击, Basnight 等[33]介绍了 PLC 固件逆向分析的过程, 如图 19 所示。

Garcia 等[34]详细介绍了针对 PLC 的固件攻击全过程, 利用内嵌的固件升级机制、在线代码注入等达到固件攻击的目的, 具体的攻击过程如图 20 所示。

通过双向修改实现攻击过程, 修改 PLC 的控制指令, 从而摧毁物理世界; 同时修改传感器的测量值, 使得操作员看到“合理”的数值, 从而达到隐藏攻击, 避开检测的目标。

Abbasi 等[31]在 2016 年黑帽大会上开发了一套隐蔽攻击的 PLC Rootkit。它会干扰 PLC 运行时与逻辑同 I/O 外设间连接, 操纵相关的 I/O 及 PLC 流程, 同时影响 PLC 进行通信交互作用以处理物理控制的 I/O 数据块。这是一种更为底层的攻击, 有很好的隐蔽性, 从而可以逃避检测。

4 PLC 安全防护机制

大量的遗留设备、专用的软硬件、有限的处理能力以及地域上分布的广泛性, 极大的阻碍了传统计算机体系中低廉且高效的安全防护方案的推广应用。《GB/T 33008.1-2016 工业自动化和控制系统网络安全 可编程序控制器(plc) 第 1 部分: 系统要求》[32]给出了系统能力等级(CL)如下:

(1) CL1: 提供机制保护控制系统防范偶然的、轻度的攻击;

(2) CL2: 提供机制保护控制系统防范有意的、利用较少资源和一般技术的简单手段可以能达到的较小破坏后果的攻击;

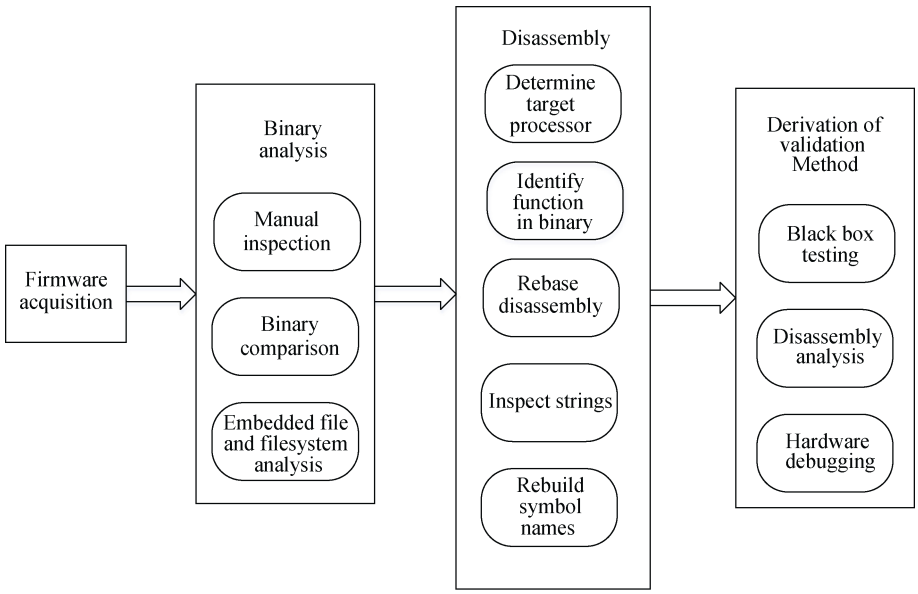


图 19 PLC 固件逆向分析过程

Figure 19 PLC Firmware Reverse Analysis Process

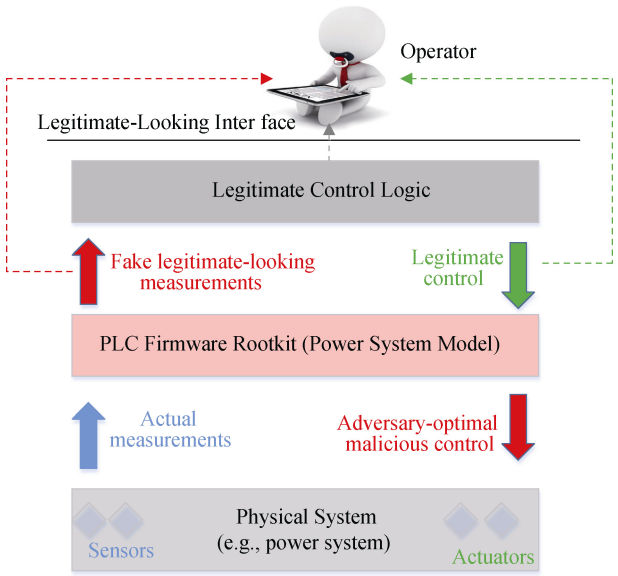


图 20 PLC 固件双向攻击过程

Figure 20 PLC Firmware Two-Way Attack Process

(3) CL3: 提供机制保护控制系统防范恶意的、利用中等资源、PLC 特殊技术的复杂手段可能达到的较大破坏后果的攻击;

(4) CL4: 提供机制保护控制系统防范恶意的、使用扩展资源、PLC 特殊技术的复杂手段与工具可能达到重大破坏后果的攻击。

基于《GB/T 33008.1-2016》的安全要求, 我们从硬件设备、通信及协议、执行行为、网络隔离、控制组态等方面对 PLC 的安全防御进行了归类整理。

4.1 基于硬件设备的 PLC 安全防护

对于 PLC 设备的安全防护主要包括:

1) 验证固件的完整性, 对固件的修改进行及时感知。Adelstein 等[33]引入基于固件签名的检测方法, 包含一个固件“验证编译器”, 允许固件在运行时接受检验, 通过检测确定的执行流特性、内存及队长完整性。Zhang 等[34]提出 IOCheck 框架, 在运行时验证固件完整性和 IO 配置, 在假设可信的 BIOS 启动之后, IOCheck 利用 X86 CPU 架构下的系统管理模式来运行完整性校验。Duflot 等[35]提出 NAVIS 框架, 来检测网卡固件的完整性。

2) 采用可信计算等技术, 提高底层数据传输的可信性。Stephen McLaughlin[35]提出应该在 PLC 中配备可信计算基(TCB, Trusted Computing Base)。可信计算基由一组可信软硬件组成, 用来保证 PLC 对安全策略的执行。图 21 是一个基于 TCB 的安全策略的执行过程。

```
# Only administrators with local access can upload guard blocks and recovery blocks.
admin:local ul-write *:*:G, *:*:R

# Only adminstrators and engineers can upload organization blocks.
admin:*, eng:* ul-write *:*:O

# Any user can upload basic function blocks.
*:* ul-write *:*:F

# Recovery blocks can only call guard blocks and other recovery blocks.
*:*:R rt-call *:*:G, *:*:R

# Guard blocks can only call recovery blocks and other guard blocks.
*:*:G rt-call *:*:R, *:*:G

# Organization blocks can only call function blocks and guard blocks.
*:*:O rt-call *:*:F, *:*:G

# Function blocks can only call function blocks and guard blocks.
*:*:F rt-call *:*:F, *:*:G

# Guard blocks can write to any memory region or data block.
*:*:G rt-write OR, MR, TR, *:*:D

# Only function blocks from administrators and process engineers may write to other memory regions.
admin:*:*, eng:*:* rt-write MR, TR

# Function blocks from local users can write to data blocks from local or remote users.
*:local:* rt-write *:local:D, *:remote:D

# Function blocks from remote users can only write to data blocks from remote users.
*:remote:* rt-write *:remote:D

# Any operations not matching rules above is denied.
default deny
```

图 21 基于 TCB 的安全策略的执行过程

Figure 21 Implementation Process of TCB-Based Security Policy

乔全胜等[36]采用 Xilinx-7000 工业级芯片搭建硬件环境, 并通过嵌入式系统移植, 在可信计算技术基础上, 以协同处理的方式实现了快速加解密验证, 用哈希(Hash)算法对 PLC 系统启动文件进行了完整性验证, 保证了 PLC 系统的可信启动。李孟君等[37]分析了可信计算与 PLC 系统结合面临的问题和挑战, 从上位机和下位机提出了基于 TPM(可信平台模块)的可信 PLC 系统构建方案, 运用可信计算技术对上位机进行了安全增强, 确保上位机运行环境的安全可控; 运用身份认证机制, 实现对上位机组态软件进行权限管理, 防止攻击者恶意篡改和替换; 运用数字签名技术, 实现对逻辑组态和监控组态的可信

软件分发管理。

4.2 基于通信及协议的 PLC 安全防护

由于原来的 PLC 控制协议没有加密、认证等机制, 导致攻击者只要能和 PLC 通信, 简单的构造畸形数据包就可以达到实施破坏的目的, 因此可以从协议及通信的角度对 PLC 进行安全防护。Spenneberg 等[38]指出, 他们设计出的专门针对 PLC 的蠕虫非常有效, 但是在遭受感染期的过程中, PLC 大概有 10 秒钟是不工作的, 在此期间原始用户的程序也不会运行, 而且会产生大量的不正常流量。而且在扫描和感染阶段, 会发送很多可疑的数据包。这些都是可以被基于流量的检测机制发现。Malchow 等[39]引入一款 PLC Guard 安全防护设备, 分析 PLC 和工程师站之间的通信流量。当工程师站向 PLC 传输代码时, PLC Guard 分析代码的传递, 并和以前版本进行比较(如图 22 所示), 包括不同层次上的图形抽象和概括。操作员可以选择接受或拒绝代码传输, 而且此设备很难被攻破。Nelson 等[40]提出使用 NAC 地址绑定来保证物理上连接的端口安全, 对 PLC 和工程师站之间的通信进行加密, 从而增加逆向分析控制协议的难度。Bestak 等[41]提出对 PLC 等设别使用一种加密算法来对测量的数据进行加密, 保证通信过程中不被攻击者恶意篡改, 从而达到数据完整性的目标。Heo 等[42]提出, 可以对自动化控制中的 PLC 通信网络进行加密, 从而保证数据真实性。Bestak 等[43]指出可以尝试在 PLC 网络中使用加密算法, 从而保证通信安全性。Andrew Clark 等[44]提出了一种新型的防御框架, 利用一组随机化的加密密钥对系统操作员发送给 PLC 的控制命令进行认证。框架利用加密分析、控制理论、博弈论方法来对恶意控制指令造成的影响的进行量化分析, 如图 23 所示, 在过程控制网络(PCN, Porocess Control Network)与控制系统网络(CSN, Control System Network)交互过程加入了主动加密机制, 用来对发送的控制命令进行认证。

Haroon Wardak 等[45]指出目前报道的大多数针对 PLC 的攻击都是基于 PLC 以一种未授权的方式建立通信, 并对 PLC 访问控制问题进行了研究, 分析了通常采用的基于密码的访问控制机制的脆弱性, 指出可以在 PLC 和其他设备之间配置工业自动化环境下的数据安全模块(Scalance S)得到解决。Stanislav Ponomarev 等[46]提出了一种通过测量和验证通过网络传输的数据来检测入侵网络的 ICS 的方法。

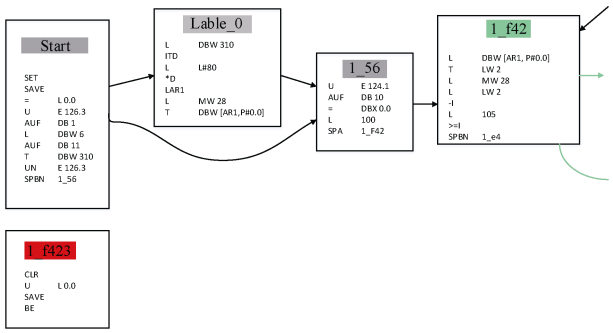


图 22 PLC Gurad 防护过程示意图

Figure 22 PLC Gurad Protection Process

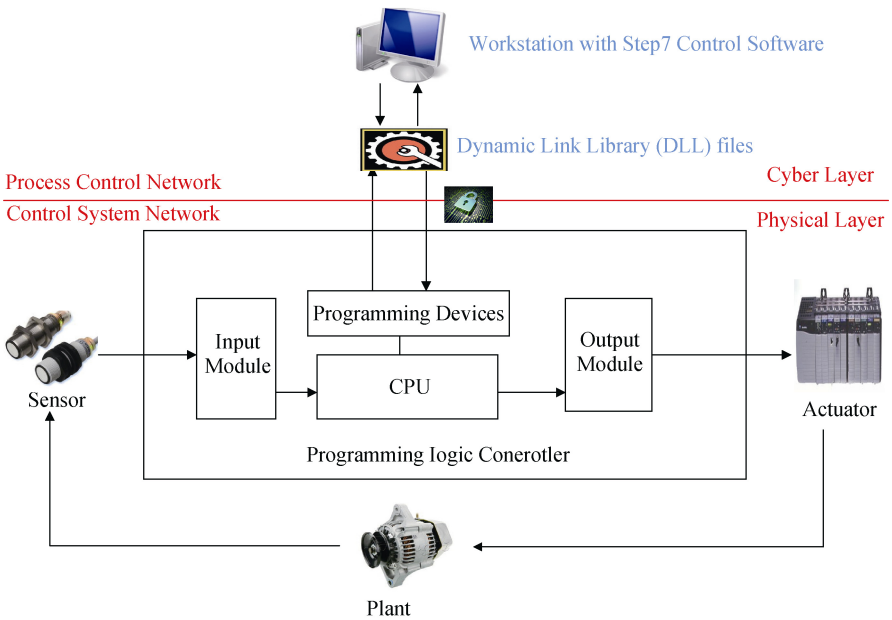


图 23 PLC 主动加密机制

Figure 23 PLC Active Encryption Mechanism

4.3 基于执行行为的 PLC 安全防护

包括对 PLC 的内存行为、发包的时间、动作执行等进行建模, 从而能够对 PLC 的执行行为进行监控。Spenneberg 等[38]指出蠕虫在感染 PLC 的过程中会自动出重启, 这是属于异常执行动作。而 Langner 等[19]指出在攻击过程中 OB1 需要首先调用恶意代码, 执行终止条件等, 这些都是可以被检测到的异常行为, 从而可以采取相应的安全措施。Chih-TaLin 等[47]对 Modbus 协议的深入分析, 提出了一种基于自动学习的恶意入侵检测方法, 并在开发的测试平台上进行了各种测试, 入侵检测框架如图 24 所示。

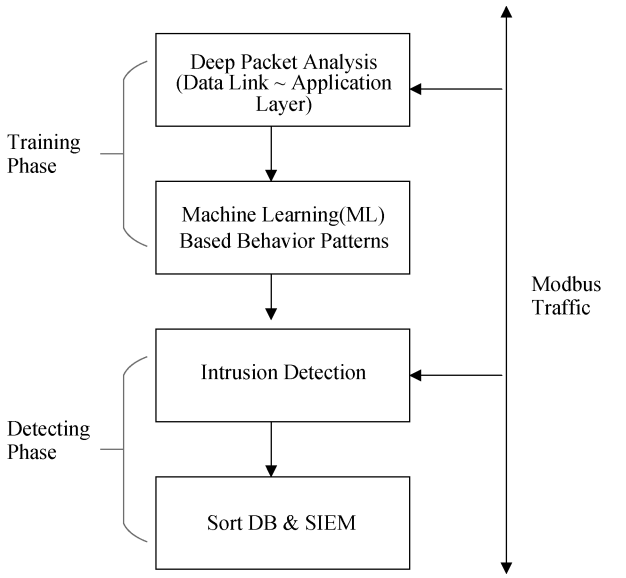


图 24 基于机器学习的入侵检测机制

Figure 24 Intrusion Detection Mechanism Based on Machine Learning

Ken Yau 等[48]提出使用半监督机器学习, 即单类支持向量机(OCSVM, One-class Support Vector Machine), 根据捕获的 PLC 存储器地址值来检测 PLC 异常行为, 主要步骤如图 25 所示。

Saman Zonouz 等[49]提出了一种利用 PLC 代码符号执行的方法来检测工业控制恶意软件。具体过程如图 26 所示, 首先对安全要求取反, 生成对应的不安全要求(UR), 然后寻找满足条件的路径 P。其中 P 为 TEG 和 UR 的笛卡尔积。如果不存在满足条件的路径, 则说明代码符合安全要求, 可以安全执行; 否则, 生成连接路径条件并计算样本输入向量, 该向量可用来进行调试。Henry Senyondo 等[50]提出了 PLCloud, 一种基于云的安全防护架构, 通过最小的基于云计算的可信安全验证模块来最小化基础设施可信计算基。PLCloud 包含两个部分: PLCloud Agent 和虚拟的 PLC 模拟器。PLCloud Agent 负责搜集 PLC 的控制指令和传感器输入, 同时接收模拟器的通知并执行响应的操作。PLC 模拟器功能比较强大, 实时与物理 PLC 设备状态保持同步, 周期性的保存快照以备灾备恢复时使用, 并可以执行模型检验、控制流图符号执行、安全要求校验、线性时序模型检测等功能, 从而提供最小化基础设施可信计算基。

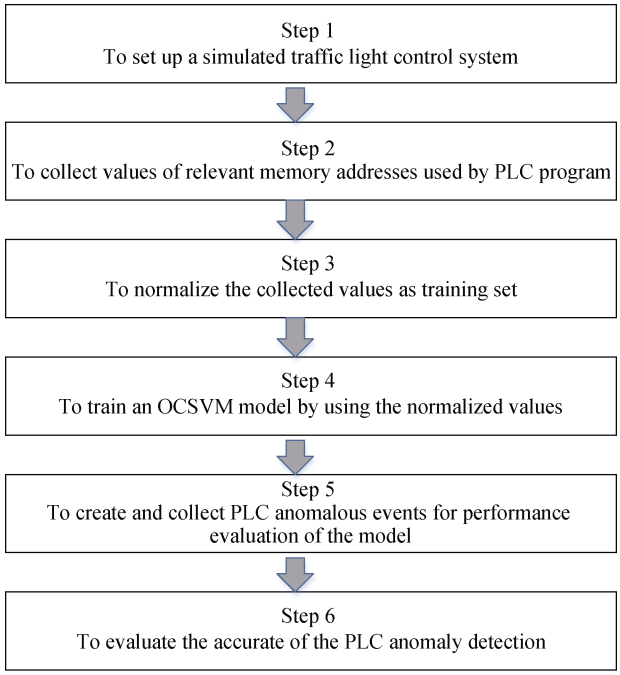


图 25 利用 OCSVM 的异常检测步骤

Figure 25 Anomaly Detection Steps Using OCSVM

4.4 基于网络隔离的 PLC 安全防护

使用工业防火墙[51], 对 PLC 的通信进行过滤和防护, 这是目前最常见也是最常用的 PLC 安全防护手段, 相对比较保守, 在无法确保 100%准确率的情况下, 一般采取的策略是只进行告警而不进行阻断。Stevan 等[52]通过对比多种不同厂家不同型号的 PLC, 指出, 不一定要利用 0-day 漏洞就可以对 PLC 实施攻击, 只需要和 PLC 建立通信即可。因此使用防火墙和 VPN 对 PLC 设备进行隔离是安全防护的第一步。Be´la Genge 等[53]提出了利用防火墙和深度防御策略来进行安全防护的方法。Sandaruwan 等[54]提出 IDS 应与防火墙一起部署, 以防止感染的访问。David Kuipers 等[55]提出了纵深防御策略, 对 PLC 等进行隔离, 首先对整个工业生产系统进行安全分区, 然后在不同过的区域之间设置防火墙进行隔离, 从而保证跨越不同的安全区域进行访问能够受到限制。

4.5 基于控制组态的 PLC 安全防护

控制组态包含了 PLC 的执行逻辑, 是直接作用于生产过程的程序, 因此对控制组态的安全性验证

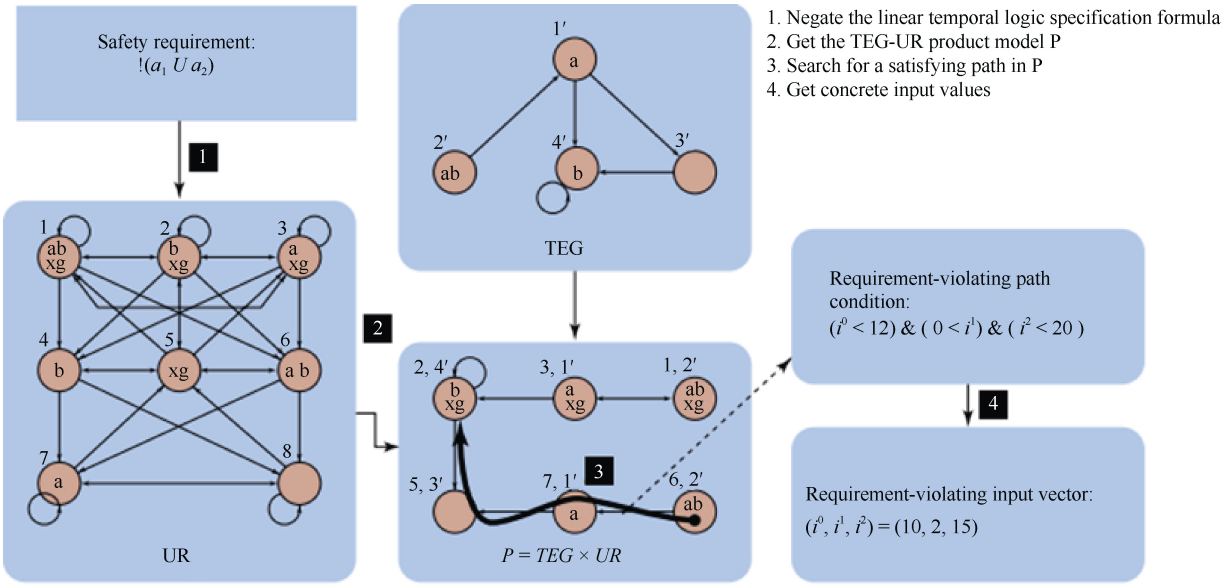


图 26　PLC 代码符号执行安全检测机制

Figure 26　PLC Configuration Symbol Execution Security Detection Mechanism

也是 PLC 安全防护的重要组成部分。Litian Xiao 等[56]提出了一种对 PLC 程序(组态)进行联合验证的层次化框架, 主要包括代码层面、模型层面和合规性层面。在代码层面上, 对应于 PLC 软件测试, 分析了静态测试、真实环境测试、硬件检测器测试、仪器测试和模拟测试的适用性。在模型层面上, 根据 PLC 程序体系结构, 形式化描述和语义定义, 采用线性时态逻辑语法和语义的方法引入算术符号转换系统。在合规性层面上, 提出了一种基于 PLC 程序定理验证技术的正确性验证框架。具体过程如图 27 所示。

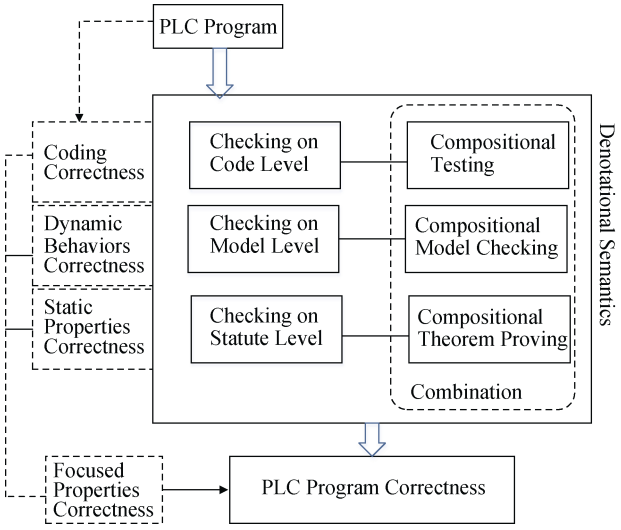


图 27　PLC 组态联合验证框架

Figure 27　PLC Configuration Joint Verification Framework

Yu Jiang 等[57]提出了混合关系模型(HRM, hybrid relation model), 对 PLC 的梯形图组态进行了分析, 并证明构建的用来捕获组态执行逻辑的 HRM 是一个贝叶斯网络(BN, Bayesian Network), 利用 BN 的运算机制, 来处理组态执行过程中的硬件故障概率。Mo Xia 等[58]指出, 通过传统的测试难以发现由 PLC 构成的复杂系统的逻辑漏洞, 形式化证明引入了严格的数学分析来枚举所有的状态空间, 但是尚无对 PLC 能进行验证的有效工具, 而通用的形式化验证工具需要大量的相关背景知识, 因此提出了一个针对 PLC 系统的自动化建模和模型检测工具, 称为 FMMC。包括图形化建模、语法检验、代码生成、代码优化和违规代码的呈现, 用户可以方便地找到错误的来源。Cleber A. Sarmento 等[59]将 LD 编写的 PLC 控制程序被建模为扩展的有限状态机, 并且随后被正式验证。从这个验证过程中, 可以识别这些机器中的功能错误, 并因此识别控制程序中的相关错误。Sridhar Adepu 等[60]提出了一种方法来推导基于状态的不变量, 从 ICS 设计开始, 并使用从中导出不变量的扩展混合自动机对其过程动态进行建模。每个不变量都被编程并插入到适当的 PLC 中伴随着控制代码。不变量在 ICS 操作期间处于活动状态, 并根据系统设计检查系统状态的有效性。Akinori Mochizuki 等[61]介绍了一种白名单设计技术, 通过 Petri 网对现场设备的正常行为进行建模, 并将白名单模型转换为梯形图, 能够辅助 PLC 检测到网络攻击。

Luis Garcia 等[62]从可编程逻辑控制器程序扫描

周期内直接执行高级安全和验证解决方案, 操作员或程序客户端向临时缓冲区写入值, 调用验证库中的功能对该值进行验证, 如果值没有违反安全约束, 那么在临时缓冲区中的值就会被传送到目的缓冲区, 具体流程如图 28 所示。

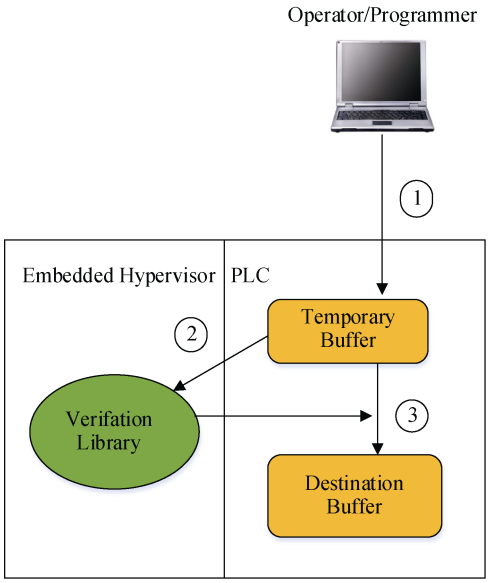


图 28 PLC 组态验证过程

Figure 28 PLC Configuration Verification Process

Huayang Cao 等[63]指出过程控制系统存在的两个弱点是, PLC 设备无法验证控制代码的有效性, 无法监控控制设备本身的状态。针对这两个漏洞, 设计了一种基于 HMAC 算法的代码数字签名机制, 提出了监控控制设备状态的主动和被动方案。

4.6 其他防护方法

还有其他的一些安全防护方法, 比如 Buza 等[64]提出 Crysys PLC honeypot (CryPLH)系统(如图 22 所示), 来发现针对 PLC 的攻击。这种 PLC 蜜罐系统可以是整个安全监控系统的一部分。实验验证, 此蜜罐系统对于攻击者来说与真实的 PLC 是不可区分的。Nelson 等[40]提出使用 NAC 地址绑定来保证物理上连接的端口安全, 对 PLC 和工程师站之间的通信进行加密, 从而增加逆向分析控制协议的难度。

5 PLC 功能安全与信息安全的关系

虽然功能安全和信息安全在中文表述中都使用了“安全”一词, 而在英文表述中使用“safety”和“security”进行了区分[71-77]。Piètre-Cambacédès 等[65]专门对“safety”和“security”进行了辨别, 给出了两个特征用于区分: (1)是系统对环境造成影响还是环境对系统造成影响; (2)是恶意的还是无意的。

功能安全(safety)一般指的是风险来自于系统, 可能对环境造成影响, 而且往往是意外性的风险; 而信息安全(security)源于环境的威胁, 会潜在的影响系统, 而且一般是恶意的威胁。

而且, 功能安全(safety)是核心, 信息安全(security)最终目的也是为了保障功能安全。PLC 在诞生之初, 由于当时技术水平的限制和实际的自动化需求, 仅仅关注于功能安全, 而忽略了信息安全, 将系统的可用性放在最高的优先级, 导致信息安全防护的缺失。

PLC 受到来自外部环境的信息安全攻击, 导致 PLC 系统本身受到损害, 而这种损害可能反过来影响人和环境, 导致功能安全问题。除了随机的、意外的系统故障外, 由于信息安全攻击导致的系统功能失效或失控, 都会到环境造成更大的伤害。因此, 我们可以得出这样的结论: 功能安全是根本, 信息安全是手段。

6 PLC 信息安全展望

PLC 作为工业控制系统的重要组成部分, 其安全性关乎整个系统的稳定可靠运行。因此, 加强对 PLC 的安全防护是重中之重。

目前针对 PLC 的安全防护, 提出以下建议与展望:

1. 从 PLC 自身的安全着手。(1)采用安全增强型的嵌入式操作系统, 添加安全防护手段, 及时对操作系统进行安全升级; (2)采用安全性较高的通信协议, 运用加密、认证等已经成熟的技术手段, 增加 PLC 通信的安全性; (3)增强 PLC 的处理能力, 使 PLC 自身能运行一些安全防护措施。

2. 寻找 PLC 的替代品。目前已经有厂家在研究如何利用性能更加强大的 PC 机来取代 PLC。PC 有强大的处理能力, 而且设备成本、安全升级、后期维护等都非常便利[73-80]。

3. 利用虚拟化技术, 将 PLC 硬件作为底层基础设施, 通过设备虚拟化等技术, 将 PLC 作为一种服务提供给用户, 比如 Givehchi 等[81]提出将控制作为一种服务(Control-as-a-service), 从而能够在虚拟化时加入安全措施[82-87]。

4. 采用多级安全审计工具, 从现场设备、传感器、执行器、PLC, 再到上层的 HMI, 业务网络, 层层实施监控, 实时对数据进行比对, 及时发现设备中存在的异常。目前中科院信息工程研究所五室研发的工控系统审计系统已经在实际中有所应用。

5. 利用机器学习的方法, 对 PLC 与被控物理设

备进行建模, 构建设备的物理指纹信息, 这样能够及时发现异常, 并采取相应的安全措施。包括 PLC 的能耗信息, 控制指令的下发和回传时间等[88, 89]。

这些安全防护措施有些是需要 PLC 生产厂商的参与, 有些需要高校、研究院所等的投入, 同时需要政府等主管部门制定相关的安全标准加以引导和规范。总之, PLC 安全贯穿其整个生命周期, 需要行业的上下游共同团结起来, 打造 PLC 的安全“生态系统”。

致 谢 本文得到(1)天基资源网络化服务体系构建与在轨验证 课题 6:天基信息安全共享与服务机制研究的支持, 课题号: ZDRW-KT-2016-02-06; (2)北京市科学技术委员会(Beijing Municipal Science & Technology Commission)的课题 “国家关键基础设施安全监管平台核心技术研究(Research on Core Technologies of national key infrastructure security supervision platform)” 的支持, 课题号: Z161100002616032; (3)国家重点研发计划基金资助项目(China National Key R&D Program)的支持, 课题号: 2016QY06X1205; 在此表示衷心的感谢。

参考文献

[1] T. Chen, and S. Abu-Nimeh, “Lessons from Stuxnet.” *Computer*, vol. 44, no. 4, pp. 91-93,2011.

[2] “Son of Stuxnet: The Digital Hunt for Duqu, a Dangerous and Cunning U.S.-Israeli Spy Virus,” Zetter, and Kim, http://www.connexions.org/CxLibrary/CX16983.htm, Dec. 2014.

[3] Rrushi, Julian, et al., “A quantitative evaluation of the target selection of havex ics malware plugin.” in *Industrial Control System Security Workshop(ICSS’2014)*. Dec. 2014.

[4] Liang, Gaoqi, et al., “The 2015 ukraine blackout: Implications for false data injection attacks.” *IEEE Transactions on Power Systems,* vol. 32, no. 4, pp. 3317-3318, 2017.

[5] [5] “Industrial Control System Network Security Situation Reprot,” https://wenku.ofweek.com/show-34075.html, Janu. 2017. (“2016 工业控制网络安全态势报告”, https://wenku.ofweek.com/show-34075.html, Janu. 2017.)

[6] MM Lashin, “Different applications of programmable logic controller (PLC),” *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT’2014)*, vol. 4, no. 1, pp. 27-32, 2014.

[7] A. Shahzad, S. Musa, and A. Aborujilah, et al., “The SCADA review: system components, architecture, protocols and future security trends,” *American Journal of Applied Sciences*, vol. 11, no. 8, pp.1418, 2014.

[8] Peng, and Daogang, et al. ,“Design and development of Modbus/RTU master monitoring system based on embedded PowerPC platform,” *IEEE* pp. 2148-2152. 2009.

[9] [9] Ren, and L. Sheng, et al. ,“Development of PLC-based Tension Control System,” *Chinese Journal of Aeronautics*, vol. 20, no. 3, pp. 266-271, 2007.

[10] Ioannides, and M. G., “Design and implementation of PLC-based monitoring control system for induction motor,” *IEEE Transactions on Energy Conversion*, vol. 19, no. 3, pp. 469-476, 2004.

[11] Han, and Jinsoo, et al., “Smart home energy management system including renewable energy based on ZigBee and PLC,” *IEEE Transactions on Consumer Electronics*, vol. 60, no. 2, pp. 198-202, 2014.

[12] Lampe, Lutz, and A. J. H. Vinck, “On cooperative coding for narrow band PLC networks,” *AEU - International Journal of Electronics and Communications*, vol. 65, no. 8, pp. 681-687, 2011.

[13] Knapp, Eric , and J. Broad, “Industrial network security : securing critical infrastructure networks for Smart Grid, SCADA, and other industrial control systems,” Syngress, 2014.

[14] Yuan, and Yunlong, “Realization of PLC Control System Based on Configuration Software,” *Process Automation Instrumentation* , vol. 18, no. 9, pp. 214-220, 2006.

[15] Dai, W. William, and V. Vyatkin, “Distributed PLC Control Systems Using IEC 61499 Function Blocks,” *IEEE Transactions on Automation Science & Engineering*, vol. 9, no. 2, pp. 390-401, 2012.

[16] D. Beresford, “Exploiting siemens simatic s7 plcs,” *Black Hat USA*, vol. 16, no. 2, pp. 723-733, 2011.

[17] C. Schuett, J. Butts, and S. Dunlap, “An evaluation of modification attacks on programmable logic controllers,” *International Journal of Critical Infrastructure Protection*, vol. 7, no. 1, pp. 61-68, 2014.

[18] B. Zhu, A. Joseph, and S. Sastry, “A taxonomy of cyber attacks on SCADA systems,” *IEEE Internet of things(IoT’2011)*, pp. 380-388, 2011.

[19] “A time bomb with fourteen bytes,” R. Langner, http://www. langner. comlen, Sept. 2011.

[20] B. Meixell, and E. Forner, “Out of control: Demonstrating scada exploitation,” *Black Hat USA*, 2013.

[21] G. Tzokatziou, L. Maglaras, and H. Janicke, “Insecure by design: using human interface devices to exploit SCADA systems,” *Proceedings of the 3rd International Symposium for ICS & SCADA Cyber Security Research*, pp. 103-106, 2015.

[22] “PLC helpful during the creation, configuration and development of UNICOS applications, generating code for the control and supervision,” SIEMENS, http://accelconf.web.cern.ch, 2013.

[23] M.Adambaev, and A. Auezova, “Programming controllers and visualization in the software environment Unity Pro,” *The 11th International Scientific Conference “Information Technologies and*

Management. pp. 18-19, 2013.

[24] B. Radvanovsky, "Project shine: 1,000,000 internetconnected scada and ics systems and counting," *Tofino Security*, pp.19-20, 2013.

[25] E. Sohl, C. Fielding, and T. Hanlon, et al., "A Field Study of Digital Forensics of Intrusions in the Electrical Power Grid," *Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy*, pp.113-122, 2015.

[26] T. Newman, T. Rad, and LL. ELCnetworks, et al., "SCADA & PLC vulnerabilities in correctional facilities," *Core Security*, 2011.

[27] J. Klick J, S. Lau, and D. Marzin, et al., "Internet-facing PLCs-a new back orifice," *Blackhat USA*, 2015.

[28] S.E. McLaughlin, "On Dynamic Malware Payloads Aimed at Programmable Logic Controllers", *HotSec*. 2011.

[29] S.McLaughlin S, and P. McDaniel, "SABOT: specification-based payload generation for programmable logic controllers", *Proceedings of the 2012 ACM conference on Computer and communications security*. pp. 439-449, 2012.

[30] R. Spenneberg, M. Brüggemann, and H. Schwartke, "Plc-blaster: A worm living solely in the plc," *Black Hat Asia*, *Marina Bay Sands, Singapore,* 2016.

[31] A. Abbasi, and M. Hashemi, "Ghost in the PLC: Designing an Undetectable Programmable Logic Controller Rootkit via Pin Control Attack," 2016.

[32] "Industrial automation and control system security.Programmable logic controller (PLC). Part 1: System requirements," SAC TC 124 and SAC TC 260, http://www.zbgb.org/2/ StandardDetail3630706.htm, Octo, 2016.
("工业自动化和控制系统网络安全 可编程序控制器(PLC) 第 1 部分:系统要求", SAC TC 124 and SAC TC 260, http://www.zbgb.org/2/StandardDetail3630706.htm, Octo, 2016.)

[33] F. Adelstein, M. Stillerman, and D. Kozen, "Malicious code detection for open firmware," *Proceedings IEEE 18th An-nual Computer Security Applications Conference(CSAC'02)*, pp. 403-412, 2002.

[34] F. Zhang F, H. Wang, and K. Leach, et al., "A framework to secure peripherals at runtime," *European Symposium on Research in Computer Securit*y, pp. 219-238, 2014.

[35] L. Duflot, Y. A. Perez, and B. Morin, "What if you can't trust your network card?" *International Workshop on Recent Advances in Intrusion Detection*, pp. 378-397, 2011.

[36] Quansheng, and Qiao , et al., "Design and Implementation of the Trusted PLC," *Process Automation Instrumentation* , 2016.

[37] Meng-Jun, and L. I. , et al., "Research on Trusted Computing Constructing Technology for PLC System," *Software Guide*, vol. 16, no. 11, pp. 168-171, 2017.

[38] R. Spenneberg, M. Brüggemann, and H.Schwartke, "Plc-blaster: A worm living solely in the plc," *Black Hat Asia*, *Marina Bay Sands, Singapore,* 2016.

[39] J. O. Malchow, D. Marzin, and J. Klick J, et al., "Plc guard: A practical defense against attacks on cyber-physical sys-tems," *2015 IEEE Conference on Communications and Network Security (CNS'15)*, pp. 326-334, 2015.

[40] T. Nelson, "Common control system vulnerability," Idaho National Laboratory (INL), 2005.

[41] I. Bestak, and M. Orgon, "Performance measurement of encryption algorithms used in PLC devices," *Interna-tional Journal of Research and Reviews in Computer Science (IJRRCS)*, vol. 2, no. 5, 2011.

[42] J. Heo, C. S. Hong, and S. H. Ju, et al., "A security mecha-nism for automation control in PLC-based net-works," *2007 IEEE International Symposium on Power Line Communications and Its Appli-cations(ISPLC'07)*, pp. 466-470, 2007.

[43] I. Bestak, M. Orgon, "The use of encryption algo-rithms in PLC networks," *Simulation*, vol. 3, no.64, pp. 168-169, 2012.

[44] A. Clark, Q. Zhu, and R. Poovendran, et al., "An impact-aware defense against Stuxnet," *IEEE American Control Conference*, m pp. 4140-4147, 2013.

[45] H. Wardak, S. Zhioua, and A. Almulhem, "PLC access control: a security analysis," *IEEE Industrial Control Systems Security*, pp. 1-6, 2017.

[46] S. Ponomarev, "Intrusion Detection System of industrial control networks using network telemetry," *Dissertations & Theses - Gradworks*, 2015.

[47] C. T. Lin, S. L.Wu, and M. L. Lee, "Cyber attack and defense on industry control systems," *IEEE 2017 Conference on Dependable and Secure Computing*, pp. 524-526, 2017.

[48] K. Yau, K. P. Chow, and S. Yiu, et al., "Detecting anomalous behavior of PLC using semi-supervised machine learning," *2017 IEEE Conference on Communications and Network Security*, pp. 580-585, 2017.

[49] S. Zonouz, J. Rrushi, and S. Mclaughlin, "Detecting Industrial Control Malware Using Automated PLC Code Analytics," *IEEE Security & Privacy*(S&P'14), vol. 12, no. 6, pp. 40-47, 2014.

[50] H. Senyondo, P. Sun, and R. Berthier, et al., "PLCloud: Comprehensive power grid PLC security monitoring with zero safety disruption," *IEEE International Conference on Smart Grid Communications*, pp. 809-816, 2015.

[51] W. Shang, Q. Qiao, and M. Wan, et al., "Design and Im-plementation of Industrial Firewall for Modbus/TCP," *2015 2nd Conference on Information and Network Security for Security & Protection(ICINS2015)*, vol. 1, no. 5, pp. 432-438, 2015.

[52] A. Stevan, Milinković, and R. Ljubomir, "Industrial PLC security issues," *IEEE Telecommunications Forum*, pp. 1536-1539, 2012.

[53] F. Graur, and P. Haller, "Experimental assessment of network design approaches for protecting industrial control systes," *Elsevier Science Publishers B. V.* , 2015.

[54] G. P. H. Sandaruwan, P. S. Ranaweera, and V. A. Oleshchuk, "PLC security and critical infrastructure protection," *IEEE International Conference on Industrial and Information Systems*, pp. 81-85, 2014.

[55] M.Fabro, "Control Systems Cyber Security: Defense-in-Depth Strategies," 2006.

[56] L. Xiao, M. Li, and M. Gu, et al., "A hierarchy framework on compositional verification for PLC software," *IEEE International Conference on Software Engineering and Service Science*, pp. 204-207, 2014.

[57] Y. Jiang, H. Zhang, and X. Song, et al., "Bayesian-Network-Based Reliability Analysis of PLC Systems," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 11, pp. 5325-5336, 2013.

[58] M. Xia, M. Sun, and G. Luo, et al., "Design and implementation of automatic verification for PLC systems," *IEEE Computer Society IEEE International Conference on Cognitive Informatics & Cognitive Computing*, pp. 374-379, 2013.

[59] C. A. Sarmento, D. J. S. Filho, and P. E. Miyagi, "Extending the verification coverage for PLC control programs: A functional safety approach," *38th Annual Conference on IEEE Industrial Electronics Society(IECON2012)*, pp.2833-2838, 2012.

[60] S. depu, and A. Mathur, "From Design to Invariants: Detecting Attacks on Cyber Physical Systems," *IEEE International Conference on Software Quality, Reliability and Security Companion*. pp. 533-540, 2017.

[61] A. Mochizuki, K. Sawada, and S. Shin, et al., "On experimental verification of model based white list for PLC anomaly detection," *Asian Control Conference*, pp. 1766-1771, 2017.

[62] L. Garcia, S. Zonouz, and W. Dong, et al., "Detecting PLC control corruption via on-device runtime verification," *IEEE Resilience Week*, pp. 67-72, 2016.

[63] H. Cao, K. Chen, and P. Zhu, "Secure process control system of industrial networks," 2013.

[64] D. I. Buza, F. Juhász, and G. Miru, et al., "CryPLH: Pro-tecting smart energy systems from targeted attacks with a PLC honeypot," *International Workshop on Smart Grid Security*, Springer, Cham, pp. 181-192, 2014.

[65] L. Piètre-Cambacédès, and C. Chaudet, "The SEMA ref-erential framework: Avoiding ambiguities in the terms "security" and "safety"," *International Journal of Critical Infrastructure Protection*, vol. 3, no. 2, pp. 55-66, 2010.

[66] A. Burns, J. McDermid, and J. Dobso, "On the meaning of safety and security," *The Computer Journal*, vol. 35, no. 1, pp. 3-15, 1995.

[67] M. B. Line, O. Nordland, and L. Røstad, et al., "Safety vs security?" *PSAM conference, New Orleans, USA*, 2006.

[68] J. Rushby, "Critical system properties: Survey and taxonomy," *Reliability Engineering & System Safety*, vol. 43, no. 2, pp. 189-219, 1994.

[69] Firesmith, and G. Donald, "Common concepts underlying safety security and survivability engineering," *No. CMU/SEI-2003-TN-033. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST*, 2003.

[70] D. M. Nicol, W. H. Sanders, and K. S. Trivedi, "Model-based evaluation: from dependability to security," *IEEE Transactions on dependable and secure computing*, vol. 1, no. 1, pp. 48-65, 2004.

[71] A. Avizienis, J. C. Laprie, and B. Randell, et al., "Basic concepts and taxonomy of dependable and secure computing," *IEEE transactions on dependable and secure computing*, vol. 1, no. 1, pp. 11-33, 2004.

[72] M. Al-Kuwaiti, N. Kyriakopoulos, and S.Hussein, "A comparative analysis of network dependability, fault-tolerance, reliability, security, and survivability," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 2, 2009.

[73] A. Rullán, "Programmable logic controllers versus personal computers for process contro," *Computers & industrial engineering*, vol. 33, no. 1, pp. 421-424, 1997.

[74] Ogawa, Masao, and Y. Henmi, "Recent Developments on PC+PLC based Control Systems for Beer Brewery Process Automation Applications," *International Joint International on Sice-icase*, 2006.

[75] Zhou, Min, L. I. Feng-Ting, and W. U. Wei-Min, "Research on Communication Between PC and Multi-PLC Based on VB," *Computer Engineering*, vol. 35, no. 4, pp. 103-105, 2009.

[76] Zhou, and Xinmin, "Real Time Communication between PC and S7-200 PLC Based on OPC," *Journal of Wuhan University of Technology*, 2008.

[77] Li, Pengfei, and J. Li, "Application of Communication and Remote Control in PLC Based on ZigBee," *International Conference on Computational Intelligence & Security*, 2009.

[78] Wang, and Jia Qiang, "PC Based PLC Technology," *Process Automation Instrumentation*, 2003.

[79] Kulisz, and Jozef, et al., "A PC-BASED OBJECT SIMULATOR FOR SUPPORTING PLC SOFTWARE DEVELOPMENT," *Ifac Proceedings*, vol. 43, no. 24, pp. 215-220, 2010.

[80] O. Givehchi, J. Imtiaz, and H. Trsek, et al., "Con-trol-as-a-service from the cloud: A case study for using virtualized PLCs," *2014 10th IEEE Workshop on Factory Communication Systems (WFCS'14)*, pp. 1-4, 2014.

[81] P. Gaj, M. Skrzewski, and J. Stój, et al., "Virtualization as a way to distribute pc-based functionalities," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 763-770, 2015.

[82] T. Wu, and J. R. C. “Nurse, Exploring The Use Of PLC Debugging Tools For Digital Forensic Investigations On SCADA Systems,” *The Journal of Digital Forensics, Security and Law(JDFSL2015)*, vol. 10, no. 4, pp. 79-80, 2015.

[83] C. Y. Liu, X. Y. Zheng, and C. H. Wang, et al., “A novel embedded system-based backbone communication network for smart grid,” *IEEE 2015 9th International Conference on Sensing Technology (ICST’15)*, pp. 474-481, 2015.

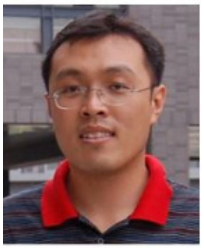
[84] T. Holczer, M. Félegyházi, and L. Buttyán, “The design and implementation of a PLC honeypot for detecting cyber attacks against industrial control systems,” 2015.

[85] T. Goldschmidt, M. K. Murugaiah, and C. Sonntag, et al., “Cloud-based control: A multi-tenant, horizontally scal-able soft-PLC,” *2015 IEEE 8th International Conference on Cloud Computing (CLOUD’15)*, pp. 909-916, 2015.

[86] G. Yang, and J. Yang, “Executing Strategy and Visualiza-tion Design for Instruction of Soft PLC System,” *Journal of Residuals Science & Technology*, vol. 14, no. 3, 2017.

[87] D. Formby, P. Srinivasan, and A. Leonard, et al., “Who's in Control of Your Control System? Device Fingerprinting for Cyber-Physical Systems,” *Network and Distrib-uted System Security Symposium(NDSS’16)*, 2016.

[88] P. Yong, X. tong and Z. miao, et al., “Scenario fingerprint of an industrial control system and abnormally detection,” *Journal of Tsinghua University (Science and Technology),* vol. 56, no. 1, pp. 14-21, 2016.
(彭勇, 向憧, 张淼,等, “工业控制系统场景指纹及异常检测”, *清华大学学报(自然科学版)*, 2016(1):14-21.)



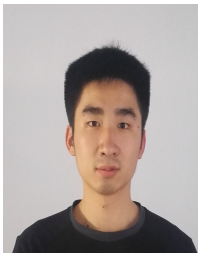
徐震 男, 博士, 1976 年生。中国科学院信息工程研究所第五研究室主任, 正高级工程师, 中科院信息化规划咨询专家, 中国电子学会高级会员, 密码行业标准化技术委员会专家。研究领域为网络安全、智能设备安全等。曾主持十余项国家信息安全科技项目, 获省部级科技进步一等奖两项。E-mail: xuzhen@iie.ac.cn

周晓军 男, 博士, 1988 年生。现任中国科学院信息工程研究所助理研究员。研究领域为工业控制系统安全、嵌入式系统安全、物联网安全。研究兴趣包括: 漏洞挖掘、协议安全分析、入侵检测。E-mail: zhouxiaojun@iie.ac.cn

王利明 男, 博士, 1978 年生。中国科学院信息工程研究所正高级工程师, 博导, CISSP, PMP。先后主持和参与了发改委、工信部、国新办等国家项目 20 余项, 申请专利 30 余项, 在国内外相关期刊或会议上发表论文 30 余篇。研究领域包括网络安全、云计算、数据安全等。研究兴趣包括: 云计算、网络安全。E-mail: wangliming@ iie.ac.cn

陈泽龙 陈泽龙, 男, 硕士, 1991 年生。2016 年毕业于中国科学院计算机技术研究所, 现任职中国科学院信息工程研究所研究实习员。研究领域工业控制系统安全、网络信息安全。研究兴趣包括: 安全数据采集、入侵检测、系统安全方案设计等。E-mail: chenzelong@iie.ac.cn

陈凯 男, 博士, 1985 年生。现任中国科学院信息工程研究所助理研究员。研究领域为网络与系统安全、身份认证、工业控制系统安全 。研究兴趣包括: 工业控制系统协议安全、漏洞挖掘和入侵检测。E-mail: chenk@iie.ac.cn



闫振博 男, 硕士, 1993 年生。于 2018 年获得中国科学院信息研究所硕士学位。研究领域为工业控制协议安全。研究兴趣包括工业代码静态检测和逆向分析。E-mail: yanzhenbo@iie.ac.cn

张伟 男, 博士研究生, 1988 年生。现在中国科学院大学信息工程研究所攻读博士学位。主要研究方向: 物联网安全, 轻量化公钥密码, 轻量化安全协议。E-mail: zhangwei@iie.ac.cn

陈聪 女, 博士, 2013 年于沈阳建筑大学机械电子工程专业获得工学硕士学位, 2018 年获得中国科学院信息工程研究所工学博士学位。研究领域为工业控制系统信息安全。研究兴趣包括: 工业控制系统网络入侵防护、异常检测识别与防御等。E-mail: chencong253@126.com