基于融合马尔科夫模型的工控网络流量异常检测方法

马 标[1], 胡梦娜[1], 张重豪[1], 周正寅[1], 贾俊铖[1], 杨荣举[2]

[1]苏州大学计算机科学与技术学院 苏州 中国 215006

[2]西门子(中国)有限公司 北京 中国 100102

摘要 虽然工业互联网为现代工业注入了新的活力, 极大地提高了工业生产效率, 但是网络化也给工业控制系统带来了更多的威胁。近年来, 国内外发生了多起工控入侵事件, 严重影响了工业生产安全, 工控安全问题愈发突出。为确保现代工业向着数字化、自动化等方向稳定发展, 有效的工控系统入侵检测方法成为了研究重点。针对工业控制系统中现有的方法对于多周期混合的流量无法进行有效分离、难以检测和防御更加复杂的语义攻击的情况, 充分利用工业流量高周期性和高相关性的特点, 提出一种基于融合马尔科夫模型的工控网络流量异常检测方法。首先深度解析报文语义并将原始流量序列映射为 hash 字符串序列, 然后根据字符串序列间的相关性生成状态转移图。接下来, 根据状态转移图间各状态的出入关系和频率将子周期符号进行分类并依次构建 DFA 模型。为了检测更多语义攻击, 该方法根据子周期间的出入关系和模型误报率将错误分解的长周期模式进行融合并在每个 DFA 模型的节点中加入时间间隔信息。在 SCADA 测试平台上进行实验验证, 结果表明此方法能检测更多类型的攻击, 对复杂语义攻击具有较高的检出率。

关键词 工业控制系统; 网络流量; 异常检测; 语义攻击

中图法分类号 TP39 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2022.05.02

Industrial Control Flow Anomaly Detection Method Based on Fusion Markov Model

MA Biao[1], HU Mengna[1], ZHANG Zhonghao[1], ZHOU Zhengyin[1], JIA Juncheng[1], YANG Rongju[2]

[1] School of Computer Science and Technology, Soochow University, Suzhou 215006, China

[2] Siemens, Ltd., Beijing 100102, China

Abstract Although the Industrial Internet has injected new vitality into modern industries and greatly improved the efficiency of industrial production, networking has also brought more threats to industrial control systems. In recent years, there have been many industrial control intrusion incidents at home and abroad, which have seriously affected the safety of industrial production, and the problem of industrial control security has become more and more prominent. In order to ensure the stable development of modern industry towards digitalization and automation, effective intrusion detection methods for industrial control systems have become the focus of research. Aiming at the situation that the existing methods in the industrial control system cannot effectively separate the multi-period mixed traffic, and it is difficult to detect and defend against more complex semantic attacks, making full use of the characteristics of high periodicity and high correlation of industrial traffic, this paper proposes a new method based on Anomaly detection method of industrial control network traffic by integrating Markov model. Firstly, the semantics of the packets are deeply analyzed and the original traffic sequence is mapped to the hash string sequence, and then the state transition diagram is generated according to the correlation between the string sequences. Next, according to the in-out relationship and frequency of each state in the state transition diagram, the sub-period symbols are classified and the DFA model is constructed in turn. In order to detect more semantic attacks, the method fuses the long-period patterns that are wrongly decomposed according to the in-out relationship between subperiods and the model false positive rate, and adds time interval information to the nodes of each DFA model.The experiment was carried out on a real SCADA test platform. The results show that this method can detect more types of attacks and has a higher detection rate for complex semantic attacks.

Key words Industrial Control System(ICS), netflow, anomaly detection, semantic attacks

通讯作者: 贾俊铖, 博士, 副教授, Email: jiajuncheng@suda.edu.cn。

本课题得到中国博士后科学基金资助及项目(No. 2017M611905)、苏州市产业技术创新专项(民生科技)项目(No. SS201701)、江苏高校优势学科建设工程资助项目(PAPD)资助。

收稿日期: 2021-03-06; 修改日期: 2021-06-29; 定稿日期: 2022-03-15

1 引言

传统的工业生产环境中, 工业控制系统(Industrial Control System, ICS)主要负责控制和协调各种设备按照生产要求执行各种任务。随着信息化的不断发展和普及, 工业生产不再是简单地完成生产任务, 还需要满足数据分析、远程操控等需求, 所以越来越多的工业控制系统与互联网融合, 形成一个开放式的网络环境。

由于传统的工业控制网络的主要功能是完成工业生产, 在设计时缺乏安全考虑, 在开放的网络环境中容易遭受攻击。从工控系统自身结构看, 由于采用专用的通信协议、操作系统和软硬件设施, 没有相应的安全防御措施, 使得系统固有的漏洞容易被攻击者利用进行破坏性的操作; 从外部网络环境看, 由于工业网络采用 TCP/IP 技术进行通信, 传统的 IT 系统攻击行为能够进入到工控系统网络, 使得工控系统面临更大的安全挑战。

工业控制系统中可编程逻辑控制器(Programmable Logic Controller, PLC)和人机界面(Human Machine Interface, HMI)的通信多采用专有的通信协议(比如 S7Comm、Modbus 等), 这导致很多互联网中通用的安全技术在工业控制网络中并不适用。并且由于传统的工业控制网络已经应用多年, 使用范围广泛, 将现有工业网络全部替换代价太大, 所以在不改变现有工业网络的基础上制定新的入侵检测系统(Intrusion Detection System, IDS)是亟需解决的问题。

在真实的工业生产中, 工业控制系统除了面对传统的网络攻击外, 还要面临一种专门针对工业控制系统的语义攻击。攻击者对工业生产流程和物理设备有着详细的了解, 可以通过构造一组看似“合法”的消息序列来有针对性地对工业设备或者工业生产造成破坏。比如在红绿灯系统中, 攻击者发起攻击时将正常流量的顺序打乱造成红绿灯顺序或时间出现故障, 从流量本身来分析, 攻击行为的流量都是正常的, 但是其造成的结果却是极其严重的。

针对现有工业控制系统异常检测方法对于多周期混合流量的语义攻击检测方面存在不足, 本文提出基于融合马尔科夫模型(Fusion Markov Model, FMM)的工控流量异常检测方法。首先将流量映射为状态事件来构造状态转移图, 然后根据状态转移图中的各节点出入关系和出入度频率将多周期混合的流量进行周期分离降低模型误报率, 为了让模型可以检测更加复杂的语义攻击对分离后的子周期根据模型误报率再次进行融合来还原真实的周期模式, 最后确定模型中确定有限自动机(Deterministic Finite Automaton, DFA)节点的时间间隔来检测时序攻击。

本文的主要贡献如下:

(1) 本文实验数据是基于 S7 协议的流量, 所以针对 S7 协议进行深度解析, 选取 S7 流量中的关键参数项将流量映射为等长的 hash 字符串, 使得后面构建状态转移图更加简洁。

(2) 针对多周期混合流量的建模, 本文将初步分离后的子周期模式根据模型误报率进行进一步融合, 使得模型可以检测更加复杂的语义攻击。在 DFA 节点中加入时间间隔信息来检测时序攻击。

(3) 使用真实的 SCADA 测试平台捕捉的流量数据进行试验。试验结果表明, FMM 方法能够更准确得分离混合周期流量, 对于复杂语义攻击具有更高的检出率。

2 相关工作

由于 ICS 在设计之初强调的是可用性和有效性, 安全方面考虑欠佳。ICS 与远程网络的连接没有完善的安全边界准入机制, 使得基于 TCP/IP 通信的工业互联网极易受到来自 IT 网络的入侵, 比如中间人攻击、拒绝服务攻击等。攻击者通过互联网入侵到过程网络, 并通过过程网络或系统漏洞将恶意程序植入到 PLC 等控制设备中, 进而直接干扰物理过程的生产, 给工业生产造成严重破坏。表 1 总结了典型的工控系统各类攻击。

表 1 攻击种类表

Table 1 Attack type table

攻击名称	攻击类型	攻击结果
Stuxnet[1-2]	渗透攻击	严重破坏浓缩铀离心机
Irongate[3]	渗透攻击	允许攻击者隐藏攻击行为
Blackenergy[4]	邮件欺骗	控制 PLC 或 RTU
Harvey[5]	渗透攻击	控制 PLC
PLC 蠕虫[6]	渗透攻击	控制 PLC
Dragonfly[7]	渗透攻击	收集 ICS 信息

其中 Stuxnet 和 Irongate 是两种针对 ICS 的渗透攻击, 它们在获取 PLC 的控制权后, 会捕获 PLC 正常工作时的出站值并重播以掩盖对受控进程发起攻击时产生的异常, 这使得生产者无法通过 HMI 来了解 ICS 真实的生产情况, 但是这些渗透攻击多是通

过 SCADA 系统进行渗透, 通过检测 SCADA 流量可以有效预防这类攻击。在真实的工业生产中, ICS 除了面对传统的网络攻击外, 还要面临一种专门针对 ICS 的语义攻击。攻击者对工业生产流程和物理设备有着详细的了解, 可以通过构造一组看似“合法”的消息序列来有针对性地对工业设备或者工业生产造成破坏, 本文主要针对复杂的语义攻击展开研究。ICS 的运行是为了完成固定的工业生产任务, 它遵循着严格的生产逻辑, 由于工业生产是高周期性的, 所以 ICS 的行为模式也是高周期性的, 通过对正常行为进行建模分析可以有效检测入侵攻击。基于行为周期的入侵检测主要利用 ICS 正常运行时的通信、运行、操作行为构建正常行为模型, 对要检测的行为与模型进行实时对比发现异常。基于正常行为模型的一个优点是它可以检测未知类型的攻击。

ICS 正常运行时需要满足工业生产规则, 通过对比 ICS 运行状态、流量特征、日志信息和传感器状态等可以发现不符合 ICS 正常运行特征的攻击行为。Hadeli H 等[8]从系统描述文件中提取信息来生成综合通信模型, 然后使用通信模型生成安全措施的不同配置文件用以检测入侵。Kwon Y J 等[9]通过分析基于 IEC 61850 的网络流量来检测异常事件, 利用静态特征和动态特征来检测网络流量数据异常。Barbosa R R R 等[10-11]比较了 SCADA 系统和传统 IT 系统, 发现 SCADA 流量有明显的周期性和自相关性, 根据固定的网络设备数、有限的协议数和规则的通信模式三个重要特征来建立 IDS。Zhanwei S 等[12]从 Modbus 流量中提取 ICS 的行为数据序列来建立控制系统的正常行为模型, 通过对比系统实际的行为数据和模型预测的行为数据来检测异常。Kalech M 等[13]通过提取 SCADA 系统中原始流量的时间特征来构建 ICS 的正常行为模型, 使用隐马尔可夫模型和人工神经网络算法来对比系统实际行为和预测行为差异。宋站威等[14]从工控网络流量中提取正常行为的离散数据序列,使用离散的多输入多输出线性模型来表示 ICS 的正常行为,通过比较分析实时提取的行为数据与模型预测的行为数据,判断是否出现异常。针对现有方法难以有效提取工控流量数据特征问题, 石乐义等[15]提出一种基于相关信息熵和 CNN-BiLSTM 的入侵检测模型, 使用信息熵的特征选择能够有效去除噪声数据和冗余特征, 减少计算量, 提高检测精度。张艳升等[16]提出一种基于卷积神经网络(CNN)的异常流量检测模型, 将工控流量特征数值与灰度图像对应生成网络流量灰度图, 并使用网络流量灰度图来训练模型, 使得模型识别精度得到了有效提高。

针对工业控制系统中的语义攻击, 国内外学者根据工业流量的高周期性特征提出了一系列有效的异常检测方法。Goldenberg N 等[17]根据工业流量高周期性的特点, 使用基于 DFA 的方法对正常运行的 Modbus 流量进行建模。该方法在简单周期的流量上取得不错效果, 但是在实际的工业生产中流量会出现一些由于人为操作、数据包丢失和重传等造成噪声, 这使得 DFA 模型的规模相当庞杂。为了减低 DFA 模型的复杂度, Kleinmann A 等[18]使用两层 DFA 串联的方法来构建模型, 并在 S7 协议上进行了验证。此方法在一定程度上缩短了 DFA 模型的整体长度, 但是对于多周期混合的工业流量无法进行有效建模, 模型也无法检测复杂的语义攻击。Yoon M K 等[19]采用概率后缀树(Probabilistic Suffix Tree, PST)构建异常检测模型, 将 Modbus 协议的请求/应答流量转化为事件序列并用 PST 表示事件间的转移关系, 根据实时计算结果与阈值的偏移程度识别异常。杨安等[20]提出一种信息流和状态流融合的工控系统异常检测方法, 使用 PST 对工控流量形成的信息流和从物理设备提取的状态流进行建模, 从操作次序和时序 2 个维度检测操作序列是否正常。Yang A 等[21]提出了一种基于状态的序列检测方法(State-Based Sequence Detection , SBSD), 通过使用隐马尔可夫模型(HMM)对 PLC 设备的观测信息序列进行建模来检测序列攻击。SBSD 方法在 ICS 中分别对连续和离散序列构建 HMM。然后通过 HMM 和加权求和, 可以获得测试序列的输出概率。最后通过将输出概率与预定阈值进行比较, 可以将测试序列分类为正常或异常。将 PLC 设备的状态流信息引入到异常检测模型中可以有效识别序列攻击并发现工控设备的异常状态, 同时 PLC 状态信息能真实反映 ICS 运行情况, 使用状态信息建模能有效降低异常检测模型的误报率。张仁斌等[22]利用工业流量间的前后关联性, 提出基于马尔科夫树模型的异常检测系统。马尔科夫树是基于状态转移图和正常流量序列构建而成, 树中的每个节点代表一个合法的状态事件, 通过树结构保存所有状态事件间的关系, 每个树的节点都维护一个转移分布表, 表明各状态事件的转移情况。此模型通过多层马尔科夫树结构和转移分布表能够检测出简单语义攻击和较为复杂的分支节点攻击, 然而由于该模型没有对多周期混合流量进行有效分离, 不同周期的流量混合在一起互相成为噪声, 使得构建的模型在准确度上面表现一般。针对多周期混合的多路复用流量, Kleinmann A 等[23]使用马尔科夫链对

SCADA 流量构建状态转移图, 然后根据状态转移图中各节点的频率和出入关系对多周期混合的流量进行周期分离, 最后分别对每个子周期构建 DFA 模型。此方法在一定程度上对多周期混合流量进行了有效分离, 但是也会将一个长周期流量分离成多个短周期流量, 这使得模型在检测子周期重放攻击时存在明显不足。

3 问题描述

3.1 周期复杂性

数据采集和监视控制系统(Supervisory Control and Data Acquisition, SCADA)被用于监视和控制关键基础设施, 例如废水分配设施、天然气生产系统和发电站等。SCADA 系统依赖 HMI 和 PLC 间的通信, HMI 根据业务需求按照一定逻辑定期向 PLC 发送相关指令, PLC 根据接收的指令内容访问现场设备的信息并将信息返回给 HMI, HMI 接收到返回信息后对信息进行呈现以达到监视控制的目的。在实际工业生产中, 存在明确的周期行为和操作顺序, 因此 SCADA 流量在业务逻辑上也存在高周期性。

SCADA 流量的周期类型[24]分为: 轮询周期和定时周期, 轮询周期指 SCADA 系统按照工业生产的业务逻辑依次执行一系列指令, 多用于从现场设备中检索数据, 定时周期是 SCADA 系统每隔固定时间执行某类操作, 常用于调整现场设备状态。在 HMI 和 PLC 通信通道中可能存在多个轮询周期和多个定时周期混合的情况。Caselli M 等[25]提出一种更加复杂的情况, 其假设 HMI 和 PLC 间的通信采用多线程的体系结构, 每个线程都负责独立任务, 线程间并发运行。在这种情况下, 工控系统中的流量是多路复用的, 即某个流量可能出现在多种周期模式中。

3.2 攻击复杂性

互联网化带来便利的同时也带来了很多安全威胁, 在真实的工业生产中, SCADA 系统除了面对传统的网络攻击, 如功能码异常、DoS (Denial of Service)、缓冲区溢出等攻击外, 还要面临一种专门针对 SCADA 系统的语义攻击[22]。在语义攻击中, 攻击者对工业生产流程和物理设备有着详细的了解, 可以通过构造一组看似“合法”的消息序列来有针对性地对工业设备或者工业生产造成破坏。图 1 是一个轮询周期状态序列“abcdbedfabcdbedf……”和另一个定时周期状态序列“AAA……”混合的状态转移图。

针对该状态图简单语义攻击可分成两类: 次序攻击和时序攻击。

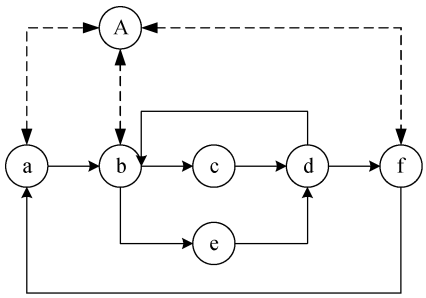


图 1 多周期混合的状态转移图

Figure 1 Multi-period mixed state transition diagram

(1) 次序攻击是指攻击者将消息指令以非法、恶意的顺序发送, 如将序列“abcdbedfabcdbedf……”中的“ab”子序列颠倒构成异常序列“bacdbedfbacdbedf……”进行次序攻击。Fovino I N 等[26]列举了一个次序攻击影响一个高压输气管的真实案例, 输气管的压强由两个阀门控制, 攻击者控制了输气管道的 PLC, 他们发送指令强制将一个阀门完全打开另一个完全关闭, 导致输气管的压强过大而停止工作。这些指令在单独检测时都是合法的, 但是当它们以一种非法顺序发送时会将系统停止工作。

(2) 时序攻击是指攻击者将消息指令在非法的时间发送, 如将序列“AAA……”的周期时间由 5s 改为 2s 构成时序攻击。美国工业安全报告[27]提及了一个时序攻击的案例。在输水系统中, 攻击者以一种异常频率给 PLC 发送正常顺序的指令, 导致输水管道的阀门快速地打开和关闭, 形成气锤效应, 造成大量输水管道破裂。

此外, 若攻击者对工业生产流程有更深的了解可以构建更加复杂的语义攻击: 分支节点攻击和子周期重放攻击。

(1) 分支节点攻击是指攻击者颠倒子序列的发送顺序进行攻击。对于分支节点 b, $b \rightarrow c$以及$b \rightarrow e$的状态转移都是合法的, 攻击者通过颠倒子序列“bcd”和“bed”的顺序, 可以构建分支节点攻击序列“abedbcdfabedbcdf……”。

(2) 子周期重放攻击是指攻击者多次重复发送子序列造成攻击。对于状态图中的子周期“AAA……”, 攻击者可以多次发送状态 A 使得整个生产流程发生改变来干扰工业生产。

4 融合马尔科夫模型构建

4.1 建模流程

首先将原始的 HMI 和 PLC 正常数据流根据 IP 通道进行分离。因为本文中的数据集流量是 S7 协议

的流量, 所以其次对通道中的 S7 协议流量进行深度解析, 提取重要特征, 并将流量数据符号化, 得到带时间戳的符号序列。接下来根据正常符号序列构建状态转移图。然后根据状态转移图中的出入关系和频率构建马尔科夫模型。最后为了检测更加复杂的语义攻击, 对马尔科夫模型进行融合处理, 确定次序模型后根据符号的时间戳确定 DFA 节点的时间间隔。具体流程如图 2 所示。

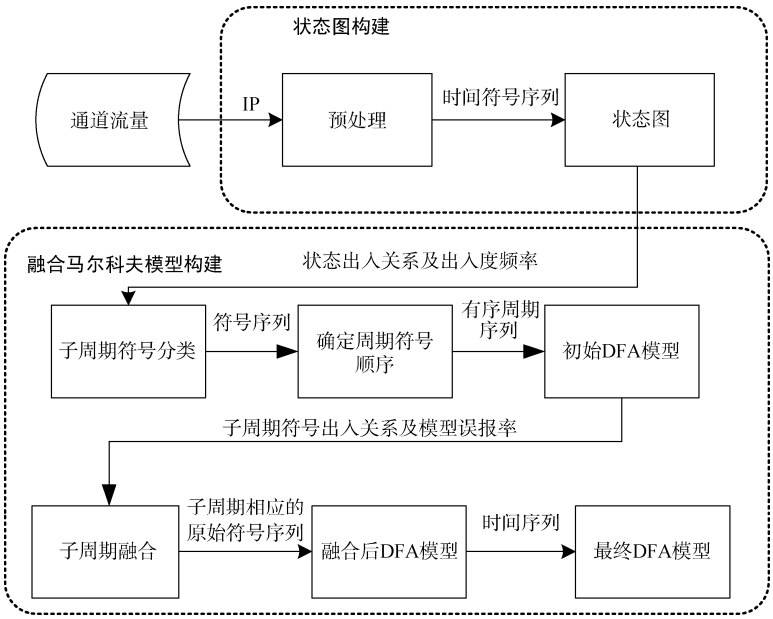


图 2 融合马尔科夫模型的建模整体流程

Figure 2 Integrated modeling process of Markov model

4.2 状态图构建

4.2.1 基于 S7 协议的状态事件定义

工业生产行为具有高周期性, 通过研究通信流量之间的关系可以挖掘工业生产实际的行为周期。在工业控制系统的交互过程中, 工业生产的具体操作是通过通信流量来执行, 将通信流量转为状态事件可以简化模型构建过程。不同的工业协议包含的字段是不同的, 对状态事件的定义也是不同的, 为了确保通信流量能准确转化为状态事件, 需要分析协议的语义特征, 选取合适的特征字段, 根据这些选取的特征字段可以将原始的流量序列转化为时间符号序列, 每个时间符号代表一个状态事件。本文以 S7 协议为例, 制定基于 S7 协议的特征提取和状态转换规则。

S7 协议是西门子自主开发的私有协议, 通信主要采用主从模式, 即主设备发起事务请求, 从设备根据请求数据执行相应的操作并将结果响应给主设备。S7 协议栈的传输层使用 TCP/IP 实现依赖于面向块的 ISO 传输服务, 协议的固定 TCP 端口号为 102, 会话层和表示层分别采用 TPKT 和 ISO-COTP 协议将服务进行封装, 应用层用 S7 协议进行数据传输。其特定形式如图 3:

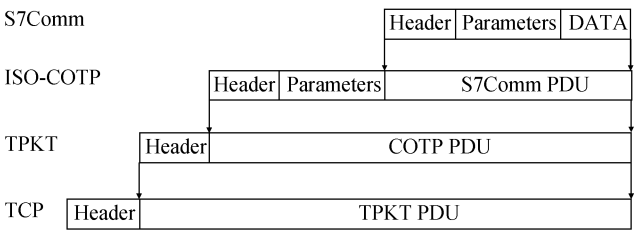


图 3 S7 协议封装结构

Figure 3 S7 protocol encapsulation structure

S7 协议是面向功能/命令的, 因此 S7 协议的消息传输多是由请求和响应组成。S7 协议控制单元(Protocol Data Unit, PDU)由头(Header), 参数(Parameters)和数据(Data) 3 个部分组成, 其中报文头由图 4[18]中前 6 个字段构成, 报文参数由 Function Code 和 Item Count 两个字段组成, 数据部分即为 Item, 由剩下的八个数据头参数和数据内容组成, 一条报文中可以包含多个 Item。

其中主要参数解释如下:

- “ROSCTR”: 表示此报文类型, 0x01 表示为请求报文, 0x03 表示为响应报文。
- “The Protocol Data Unit Reference(Request Id)”: 即事务标识符, 是一个递增的值, 用于匹配请求/响应报文对。

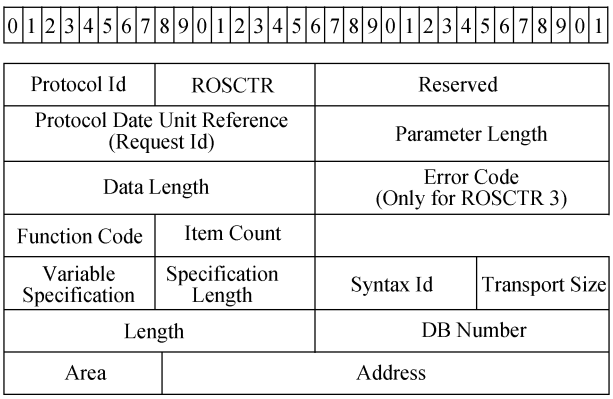


图 4　S7 PDU 结构

Figure 4　S7 PDU structure

- “Function Code” 表示协议所执行的操作类型, 如读操作为 0x04, 写操作为 0x05。
- “Item Count”: 是报文所携带的命令数, 西门子 HMI 会将多条功能相似报文合并为一条发送给 PLC, 每条报文的命令包含在 Item 项, 本文为了更好地对流量行为进行建模, 将多条融合报文进行了分离处理。

本文对 S7 协议进行语义分析, 提取协议中 Protocol Id、ROSCTR、Parameter Length、Data Length、Function Code、Item Count 和 Item 的数据头来定义状态事件。通过这些特征定义的状态事件转换规则如下:

(1) S7 报文功能码为读操作(0x04)时, 请求报文读取的对象地址相同(读取同一个现场设备信息)即为同一状态事件, 响应报文返回值的各参数一致时(返回的设备信息一致)即为同一状态事件;

(2) S7 报文功能码为写操作(0x05)时, 请求报文待写入的对象地址和写入值相同(向同一个现场设备写入相同值)即为同一状态事件, 响应报文返回值的各参数一致时(写入是否成功)即为同一状态事件。

根据上述的转换规则, 流量序列中任何一条 S7 报文都可以映射成唯一的状态事件。为了方便模型的构建, 本文先将 S7 报文中需要的参数项的值取出, 进行字符串拼接, 这样一条 S7 报文就变成了一个字符串。然后再使用 SHA-1 函数对字符串转化, 得到等长的 hash 字符串, 同时保留每个 hash 字符串的时间戳, 由此就完成了报文到时间符号的转换, 每个符号代表一个唯一状态事件。为了方便表示, 本文用 a,b,c……表示相应的 hash 字符串, 例如符号 a 表示读取某一个传感器, 符号 b 表示修改某个控制器的参数。

4.2.2　状态转移图构建

完成上述的数据预处理后, 原始的流量序列已经被处理为符号序列$SymSeq$和与符号序列对应的时间序列$TimeSeq$, 本文将这两个序列简称为时间符号序列。$States$表示状态字母表, 状态字母表中元素为$SymSeq$中的状态种类, $State$为字母表$States$中的状态。ICS 的运行是为了完成固定的工业生产任务, 它遵循着严格的生产逻辑, 由于工业生产是高周期性的, 所以 ICS 的行为模式也是高周期性的。在真实的生产过程中存在 PLC 定期自我查询、人工正常操作等特殊情况, 这些噪声是系统正常行为, 却会导致检测模型的误报。因此需要将字母表$States$中的极低频率的噪声行为去除, 当状态$State$的频率低于 0.05 时, 将此状态从$States$中去除并加入到噪声字母表$NoiseStates$中。分离完噪声行为后需要去除符号序列$SymSeq$和时间序列$TimeSeq$中噪声符号和相应的时间。

根据时间符号序列构建状态转移图可以得到事件的状态转移关系,状态转移关系使用矩阵$adjStates$来表示, 其中$adjStates[i][j]$为$adjStates$的元素, 序号i对应$States$中第i个状态:i', 序号j为$States$中第j个状态:j', $adjStates[i][j]$表示状态i'到状态j'的转移次数。构建完$adjStates$后, 再用$adjStates$矩阵中的元素除以$TimeSeq$总时间长度得到频率矩阵$adjF$, $adjF$即为最终构建的状态转移图矩阵, 其中$adjF$的元素$adjF[i][j]$表示状态i'到状态j'的转移频率。具体构建过程如下:

(1) 依次从$SymSeq$取出符号$symbol$, 判断当前符号$symbol$是否为新状态事件, 如果是, 则将当前符号$symbol$添加到状态事件字母表$States$中;

(2) 更新当前状态与前一历史状态的转移关系, 即将矩阵$adjStates$对应元素加 1, 转执行(1);

(3) 依次将矩阵$adjStates$中的值除以时间序列的总时间间隔来计算每个状态节点的出入度频率得到频率矩阵$adjF$;

(4) 根据频率矩阵$adjF$的值是否为 0, 统计每个状态节点出入度值以及相关出入状态集合。

图 5 是一个周期长度为 2 的定时周期序列“ij……”和一个周期长度为 50 的轮询序列“ababcdefghefghefg hefghefghabcdefghefghefg-hefghefgh……”组成的状态图, 其中定时周期序列由一个请求响应对即字符“ij”构成, 轮询序列由四个请求响应对即字符“ab”、“cd”、“ef”和“gh”构成, 状态图节点中的数字为每个状态的频率, 其中状态“a”的入状态集合为$\{b, h, j\}$, 出状态集合为$\{b\}$。

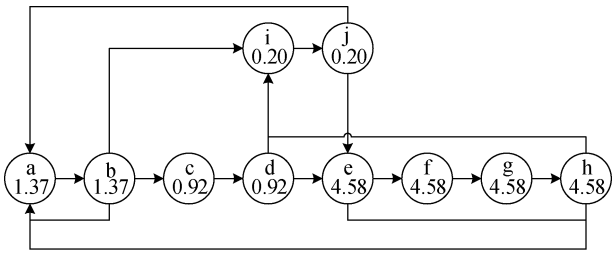


图 5　状态转移图

Figure 5　State transition diagram

理想状态下只需要一个完整工业生产周期的流量就可以得到各状态间的出入关系和每个状态在周期中的频率, 最终构建工控流量的状态转移图。但是实际的ICS运行中会产生很多数据波动, 比如ICS初始运行时工控设备需要一段时间来运行到稳定状态, 工控网络也会存在一定波动性, 如果仅仅根据少量流量构建状态转移图很难生成准确的 DFA 模型, 因此更多的正常运行流量可以有效保障状态转移图的准确构建。因此在捕获 ICS 正常运行所产生的流量时, 需要保证ICS已经平稳运行, 从而确保状态图构建的准确性。

4.3　融合马尔科夫模型构建

4.3.1　子周期符号分类算法

状态转移图反映了各状态间的出入关系和转移频率, 基于状态转移图矩阵$adjF$, 本文基于Kleinmann A等[23]的工作采用算法1中的子周期符号分类算法将字母表$States$中不同状态所代表的符号按照出入关系和出入频率分类到不同的符号集S中, 从而达到分离多个子周期的目的。最终分类得到子周期集合 $C=\{S_1,\ S_2,\ldots,S_i,\ldots,S_n\}$, 其中$S_i,\ i\in[1,n]$表示第$i$个子周期的符号集合。具体分类过程如下所示:

(1) 由于出入度都为1的节点, 在状态转移图中没有分支, 它们只属于一个子周期集合, 所以首先选择状态转移图中入度和出度都为1的节点V, 根据节点频率F_v将其分到相似频率的集合S中, 集合频率为F_s。本文用符号≈表示频率相似, 即$F_s\times(1-F_T)\leqslant F_v\leqslant F_s\times(1+F_T)$, 其中$F_T$是频率阈值, 本文实验中将$F_T$设置为0.05。若该频率的集合不存在, 则新建一个频率为F_v的集合, 将节点V加入新集合。

(2)$remainV$表示状态图中未全部分配的节点集合, 即$\forall V\in remainV,\ F_v\neq 0$。从$remainV$中选取出度或入度为1的节点$V$, 根据节点频率$F_v$将其分配到已经存在的相似频率的集合中, 并将节点$V$从$remainV$中去除。

(3) 节点可以是多个集合的成员, 表示符号在多个循环模式中的出现。V_{in}为节点V的入节点且V_{in}只属于一个已知集合, V_{out}为节点V的出节点且V_{out}只属于一个已知集合, 若所有V_{in}或所有V_{out}所在集合频率的和与节点V的频率F_v似, 则将节点V分别加入到所有相关集合中, 同时将节点V从remainV中去除。若相关集合频率和与节点频率不相似, 则再判断所有V_{in}和V_{out}所在集合频率的和与节点V的频率F_v是否相似, 如果相似则将节点V分别加入到所有相关集合中, 同时将节点V从$remainV$中去除。

(4) 若节点 V 相邻的出入节点至少有一个在已知集合中, 并且此相邻节点的出入度为 1, 则将节点 V 加入该集合, 同时修改节点的频率为$F_v=F_v-F_s$, 若F_v的值为0, 则将其中剩余节点集合remainV中去除。

(5) 先判断剩余节点集合remainV中是否还有节点, 若有, 则从剩余节点中找到频率最小的节点, 根据其频率创建新的集合并将此节点加入集合, 然后将剩余的节点都加入此集合, 同时修改各节点的频率, 若节点频率小于阈值F_T,则将节点V从remainV中去除。

(6) 最后将remainV中剩余节点根据频率加入到已知的相似集合中。

算法 1　子周期符号分类算法[23]

输入: 状态转移图矩阵 *adjF*, 状态字母表 *States*

输出: 所有子周期符号集合 *C*

设状态字母表中的状态为 *V*, *V.inDegree* 为节点 *V* 的入度, *V.outDegree* 为节点 *V* 的出度, *S* 为子周期符号集合, *F* 为频率, *V.inStates* 为节点 *V* 的入状态集合, *V.outStates* 为节点 *V* 的出状态集合, *remainV* 为完全未分配节点集合。

Step1:

1) $\forall V\in remainV|V.inDegree=1\ and\ V.outDegree=1$

2) IF $\exists S|F_s\approx F_v$

3) $S\leftarrow S\cup\{V\}$;

4)ELSE

5) Create $S;C\leftarrow C\cup\{S\};S\leftarrow S\cup\{V\};\ F_s\leftarrow F_v;$

Step2:

6) $\forall V\in remainV|V.inDegree=1\ or\ V.outDegree=1$

7) IF $\exists S|F_s\approx F_v$

8) $S\leftarrow S\cup\{V\}$;

Step3:

9) $\forall V\in remainV|V_{in}\in V.inStates,V_{out}\in V.outStates,V_{in}\ only\ \in S_{in},V_{out}\ only\ \in S_{out}$

10) IF $Sum(all\ F_{S_{in}})==F_v$

11) $\forall S_{in} \leftarrow S_{in} \cup \{V\}$;

12) ELSE IF $Sum(all\ F_{S_{out}}) == F_v$

13) $\forall S_{out} \leftarrow S_{out} \cup \{V\}$;

14) ELSE IF $Sum(all\ F_{S_{in}}, F_{S_{out}}) == F_v$

15) $\forall S_{in} \leftarrow S_{in} \cup \{V\}; \forall S_{out} \leftarrow S_{out} \cup \{V\}$;

Step4:

16) $\forall V \in remainV | V_{related} \in V.inStates\ or\ V.outStates, V_{related} \in S_{in}$

17) IF $V_{related}.inDregree = 1\ and\ V_{related}.inDregree = 1$

18) $S \leftarrow S \cup \{V\}; F_v \leftarrow F_v - F_s$;

Step5:

19) IF $remainV \neq \emptyset$

20) $\forall V \in remainV | V_{min} \leftarrow Min(V)$

21) Create $S; C \leftarrow C \cup \{S\}; F_s \leftarrow F_{V_{min}}$;

22) $\forall V \in remainV | S \leftarrow S \cup \{V\}, F_v \leftarrow F_v - F_{V_{min}}$Step6:

23) $\forall V \in remainV$

24) IF $\exists S | F_s \approx F_v$

25) $S \leftarrow S \cup \{V\}$;

根据状态转移图出入关系和频率, 子周期符号分类算法将图 4 中的字符分为了 4 个子周期: {a,b},{c,d},{e,f,g,h}和{i,j}, 成功分离轮询周期和定时周期。然而长度为 50 的轮询周期也被算法分成了 3 个子周期, 这使得下文 4.3.3 节中构建的 DFA 模型在检测复杂的语义攻击时会有所不足。

4.3.2 确定子周期内符号顺序

经过上面的符号分类得到了各子周期的符号集合 C, 为进一步构建 DFA 模型需要确定每个子周期中符号顺序。根据状态转移图中符号的出入关系和原始符号序列的顺序可以确定各集合中符号的顺序, 依次对每个子周期符号集合S_i进行下列操作:

(1) 首先根据原始的符号序列$SymSeq$确定子集合S_i中的第一个字符s_{first};

(2) 然后根据状态转移图出入关系, 确定s_{first}的下一个符号, 重复执行此操作直到集合S_i中所有的符号顺序确定;

(3) 确定最后一个符号s_{last}的下一个符号为子周期的第一个符号s_{first}, 构成一个完整的循环。

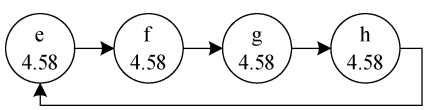


图 6 子周期{e,f,g,h}的符号顺序

Figure 6 Sign order of sub-period {e,f,g,h}

以序列“ababcdefghefghefghefghefghabcdefgh efghefghefghefgh……”中的子集合{e,f,g,h}为例, 详细阐述上述构建过程。首先根据原始的符号序列从子集合中选出第一个符号 e, 然后根据状态转移图得到 e 的下一个符号 f, 确定顺序e → f, 再依次执行确定顺序e → f → g → h。由于 h 是符号集合中最后一个符号且子周期是循环的, 所以确定再h → e的顺序, 最后构建出如图 6 所示的顺序关系。

4.3.3 DFA 模型构建

DFA 由五元组（$Q; \Sigma; \delta; q_0;$ F）组成,其中包括有穷非空的状态集合 Q, 有穷非空的输入符号字母表集合Σ, 转换函数 δ, 唯一的初始状态$q_0(q_0 \in Q)$和非空终态集合 $F(F \in Q)$。

针对每个子周期分别构建 DFA 模型, 为了让 DFA 模型符合实际建模需求, 对 DFA 进行了如下两项修改:

(1) 去除终态集合 F, 因为异常测模型在检测阶段的输入是一个无休止的重复数据流, 除非数据流结束, 否则 DFA 模型不会终止;

(2) 开始状态定义为周期流量模式中第一个符号对应的状态。

输入符号分为两类: 已知符号和未知符号。 已知符号集合由学习阶段观察到的所有输入符号组成, 并具有对应的 DFA 状态, 其余的都是未知符号, 未知符号会被直接检测为未知(Unknown)攻击。令$State_i$表示为当前状态, s_i表示将状态转移到$State_i$的输入符号, s_j表示当前接收到的输入符号并将使得状态转移到$State_j$。在 DFA 中, 将流量行为描述为正常(Normal)、重传(Retransmission)、丢失(Miss), DFA 中定义了三种类型的状态转移(输出符号):

(1) 正常(Normal): 在接收已知符号后发生“正常”状态转移, 使得状态转移到周期性序列的下一个状态, 即$s_j = s_{i+1}$。作为“正常”事件的结果, DFA 转移为其下一个状态$State_{i+1}$。

(2) 重传(Retransmission): “重传”是与前一个符号相同的已知符号的出现。作为“重传”事件的结果, DFA 保持在其当前状态$State_i$, 即$s_j = s_i$。如果模式中存在连续两个相同的符号, 则 DFA 将使得状态对同一符号有两种不同的状态转移, 即向前的“正常”状态转移和自循环的“重传”状态转移。这种不确定性可以在运行时通过选择“正常” 状态转移而不是自循环“重传”状态转移来解决。

(3) 丢失(Miss): “丢失”是指已知符号s_j未在状态$State_{i+1}$中被接收(未出现在状态转换过程的预期位置), 即$s_j \neq s_{i+1}$。作为“丢失”事件的结果, DFA 状态转移为接收s_j符号的状态$State_j$。

图 7 表示一个包含两组请求响应对(即四个状态q_1，r_1，q_2，r_2)的 DFA 模型[17]。其中黑色箭头表示接收正常符号的状态转移，蓝色箭头表示发生重传情况的状态转移，橙色箭头表示丢失情况的状态转移，为了防止第一个状态$State_1$在接受第一个符号q_1时误判为“重传”，本文将符号序列整体后移一位，使用r_1，q_2，r_2，q_1序列来构建 DFA 模型。

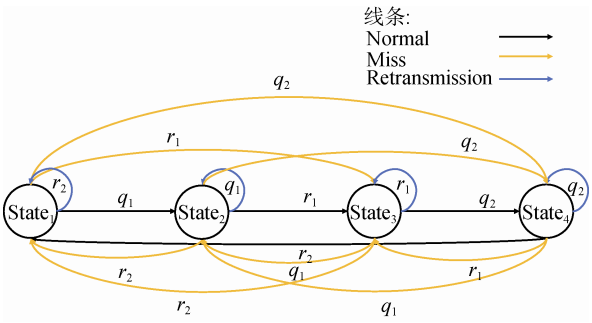


图 7　一个包含两组请求响应对的 DFA

Figure 7　A DFA containing two request-response pairs

此阶段的 DFA 是根据 3.3.2 节中获取的确定顺序的子周期符号序列来构建，对于每个子周期，DFA 模型构建过程如下:

(1) 先构造序列间的正常转换关系，即$s_j = s_{i+1}$，当前状态$State_i$在接受符号s_j后正常转移到状态$State_{i+1}$;

(2) 再构造序列间的重传转移关系，即$s_j = s_i$，当前状态$State_i$在接受符号s_j后判断为重传，当前状态不变;

(3) 最后构造序列间的丢失转移关系，即$s_j \neq s_{i+1}$，当前状态$State_i$在接受符号s_j后判断为丢失，当前状态不变。

图 8 展示了分别为每子周期单独构建 DFA 的总体结构，其中可根据子周期的周期模式情况设置 DFA 选择器。当通道中存在至少 2 种子周期模式时，需要增加 DFA 选择器来将流量符号送入相应的 DFA 模型中。

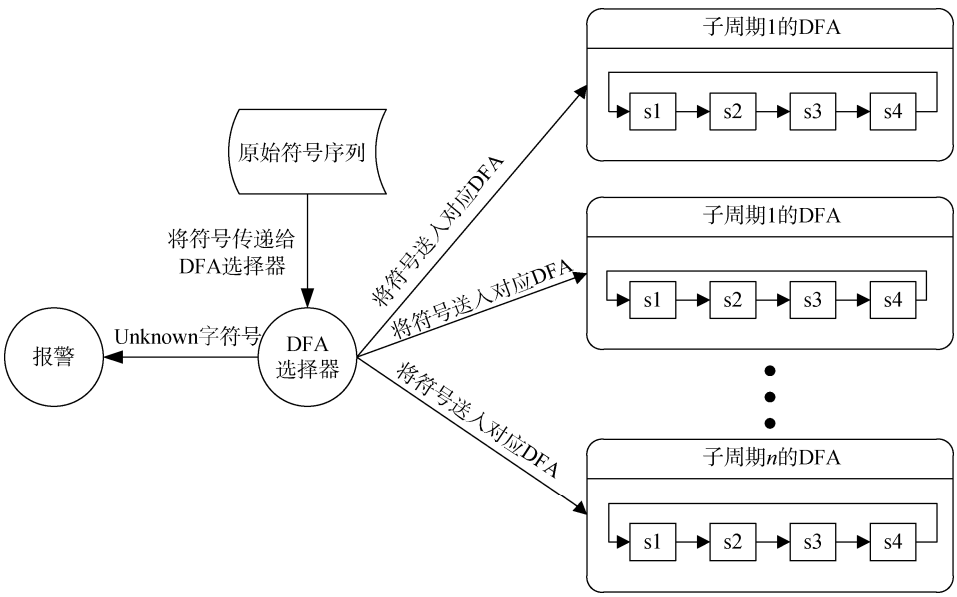


图 8　DFA 模型结构

Figure 8　DFA model structure

DFA 选择器通过分析比较通道内符号内容及时间戳来发挥选择功能[23]，主要针对以下两种情况进行设计:

(1) 子周期模式所含的符号内容均不相同，可直接按照符号内容进行选择，将流量符号送入对应的 DFA 中;

(2) 子周期模式所含的符号内容有重复的，在符号内容的基础上增加时间戳与周期值的比较来将流量符号送入对应的 DFA 中。

4.3.4　子周期融合

4.3.3 节中 DFA 模型是基于子周期分类算法进行构建，此算法在分离多周期混合的流量时也可能将一个长周期流量分解为多个子周期流量，如图 4 中的长度为 50 的轮询周期会被分类算法分解为 3 个子周期“abab……”，“cdcd……”，“efghefgh……”，如果攻击者在充分了解生产环节后不断重放“ab”子周期，DFA 模型无法检测出攻击行为，这可能导致严重的生产事故。为了提高模型针对复杂语义攻击的

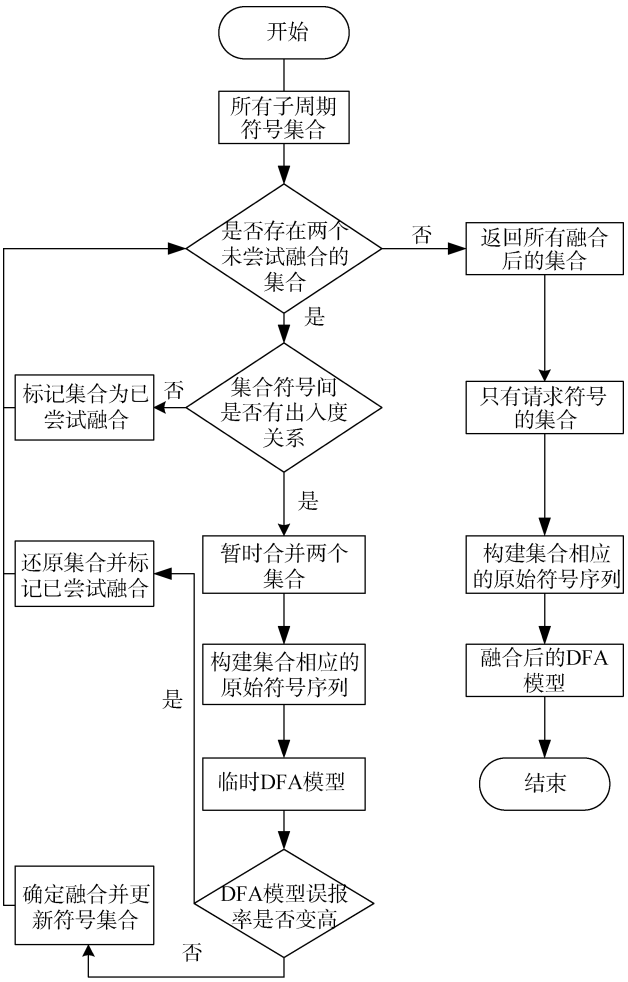


图 9　子周期融合流程图

Figure 9　Flow chart of sub-cycle fusion

检测能力, 本文在此基础上进一步提出子周期融合的方法。主要流程如图 9 所示, 具体操作如下:

(1) 通过 4.3.3 节中的 DFA 模型获取所有子周期符号集合 $C=\{S_1, S_2, \ldots, S_i, \ldots, S_n\}$, 其中 S_i, $i \in [1,n]$表示第i个子周期的符号集合, 同时可以获取模型误报率FPR_{orig}。

(2) 对于所有的符号集合, 创建矩阵$adjSet$来记录集合是否已经尝试融合, 由于集合自身不需要尝试融合, 因此初始化时 $adjSet[i][j]=1, i=j; adjSet[i][j]=0, i \neq j$, $i,j \in [1,n]$。

(3) 查找矩阵$adjSet$,从 C 中选出两个未尝试融合的集合S_i和S_j即$adjSet[i][j]=0$, 若 C 中没有符合要求的集合则执行(6), 否则对S_i和S_j进行以下操作。

(4) 对于两个集合内所有字符$V_i \in S_i$和$V_j \in S_j$, 若 $\exists V_i \in V_j.inDegreeStates$ 或 $\exists V_i \in V_j.outDegreeStates$ 即S_i和S_j中符号存在出入关系, 则暂时融合S_i和S_j中的字符生成新集合S_{temp}, 从 C 中去除S_i和S_j并将S_{temp}添加进来。然后依次根据 C 中所有子周期符号集合S_t对原始符号序列$SymSeq$进行分类生成新的子序列$SubSeq_i$, 即对于$\forall V \in SymSeq$若$V \in S_t$则$SubSeq_t \leftarrow V$。最后采用无监督学习方法对每个$SubSeq_t$构建临时 DFA 模型。无监督学习方法[17]在 DFA 构建时需要不断进行模型验证来校准 DFA 模型, 模型验证通过自动设置一个阈值, 不断比较模型表现值与阈值、模型长度与学习窗口长度来确定能够准确表示流量周期模式的 DFA。若S_i和S_j中符号不存在出入关系, 则$adjSet[i][j]=1$并重复执行(3)。

(5) 临时 DFA 模型的误报率为FPR_{temp}, 若 $FPR_{temp} > FPR_{orig}$, 将 S_{temp} 还原为 S_i 和 S_j 同时 $adjSet[i][j]=1$, 重复执行(3)。否则, 确认融合S_i和S_j并更新 C, 重复执行(2)和(3)。

(6) 经过上面步骤, 得到融合后的集合 C, 由于请求包和响应包的正常延迟会形成误报(即$q_1, r_1, q_2, r_2, \ldots$序列因为网络正常延迟会形成$q_1, q_2, r_1, r_2, \ldots$这类序列, 可能会被模型误报)所以本文根据协议解析时已匹配的请求响应将集合 C 中的$q_1, r_1, q_2, r_2, \cdots$请求响应对用$q_1, q_2, \cdots$请求来表示, 生成新的符号集合$C'$, 并根据$C'$中的每一个$S_i$生成相应的$SubSeq_i$, 采用无监督学习方法对每个$SubSeq_i$构建 DFA 模型, 得到融合后的 DFA 模型。

4.3.5　确定周期时间

对于已经完成符号序列建模的 IP 通道, 在获取每个通道对应的 DFA 序列模型后, 为了使模型可以检测时序攻击, 本文对 DFA 模型中的每个节点添加时间标记。将原始时间符号序列异常输入到融合后的DFA模型中, 对于每个子DFA模型DFA_i记录每个状态$State_{ij}$中符号的时间戳。当所有时间符号都输入结束, 通过每个状态记录的时间戳获得此状态的平均时间间隔 $T_{ij}.avg=(T_{ij}.last-T_{ij}.first)/Length(T_{ij})$, 其中$T_{ij}$表示第$i$个 DFA 的第$j$个状态的所有时间戳。需要注意的是, 每个子 DFA 的节点都有自己的$T_{ij}.avg$。

5　检测流程

在上述训练建模阶段, 获得了系统正常运行时的行为模型——融合马尔科夫模型, 下文详细介绍如何使用该模型进行异常检测, 图 10 为融合马尔科夫模型的异常检测流程图。

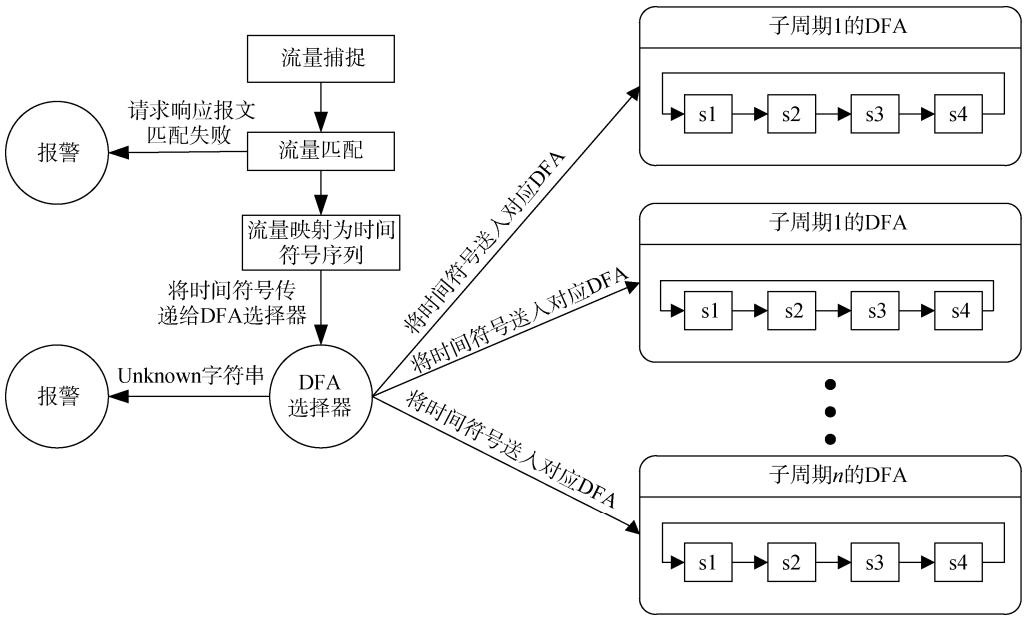


图 10　异常检测流程图

Figure 10 Anomaly detection flowchart

5.1　检测流量预处理

首先根据 S7 协议中的事务标识符对流量中的请求和响应的报文对进行匹配，如果匹配失败直接返回“丢失异常”(Miss)，否则将匹配成功的流量按照 3.2.1 节描述的过程映射为带时间戳的 hash 符号序列，此处映射时只需要处理配对后的请求报文。

5.2　异常检测

在获取待检测的时间符号序列后，对时间符号序列执行以下操作:

(1) 按照顺序依次获取符号序列中的值s_i;

(2) 将s_i送入 DFA 选择器中，判断s_i是否在 DFA 字母表$States$中，如果s_i在字母表中则将其送入相应的DFA_i，执行(3)；如果s_i不在字母表$States$但是在噪声字母表$NoiseStates$中，直接判断为正常，执行(1)；否则s_i被检测为“未知”异常(Unknown)，执行(1);

(3)DFA_i当前的状态为$State_j$，输入s_i后状态转移为$State_{j+1}$则为“正常”转换，执行(4)；输入s_i后状态转移还是$State_j$则检测“重传”异常，执行(1)；输入s_i后状态转移为State_{j+k}，$k\geqslant 2$则检测为“丢失”异常，执行(1);

(4) 对于次序正常的符号，需要进一步检测时序，T为符号s_i的时间戳，DFA_i的每个状态$State_j$都有相应的平均时间间隔$T_{ij}.avg$，$durationThreshold$为时间间隔偏差阈值，若$\left|T-T_{ij}.avg\right|\leqslant durationThreshold$则$s_i$在次序和时序上都是“正常”，否则$s_i$被检测为“时间”异常。然后执行(1)检测下一个符号。

6　实验结果与分析

6.1　实验数据集

本章所使用的训练数据集来自某一真实的 SCADA 系统测试平台。所需的实验流量从控制层网络(环网)中的多个交换机中获取。测试平台网络拓扑如图 11 所示。

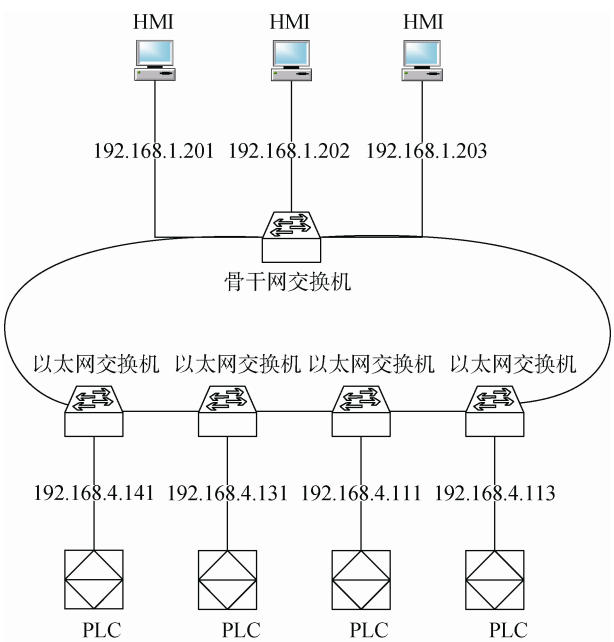


图 11　SCADA 系统测试平台网络拓扑结构

Figure 11　Network topology structure of SCADA system test platform

表 2 模型训练数据集概况

Table 2 Overview of the model training data set

#	信道	时间长度(s)	总报文数量	S7 报文数量	通道描述
1	192.168.1.201-192.168.4.141	43472	2177349	534898	控制信号灯
2	192.168.1.202-192.168.4.131	43471	1542508	119394	控制信号灯
3	192.168.1.203-192.168.4.111	43472	1293783	517513	控制 SiMotion G120C 设备
4	192.168.1.203-192.168.4.113	43472	1584727	517511	控制 SiMotion G120C 设备

本文模型训练数据集采样总时间为 12h, 总报文数量为 6598367, 其中 S7 报文数量为 1689316, 具体情况如表 2 所示。其中, 192.168.4.111 和 192.168.4.113 是两个相同的 PLC 组成的热备, 因此通道#2 和通道#4 具有基本一致的流量情况。另外, 从 SCADA 测试平台上获取的均为良性数据(即正常流量)。

为了验证模型对各种类型攻击检测的有效性, 本文模拟了交换机劫持篡改数据包的过程, 基于测试平台中获取的数据人工合成了以下几种类型的攻击数据。

(1) 非语义攻击

在非语义攻击中, 攻击者通过篡改协议的功能码将“0x04”(读操作)改为“0x29”(关闭 PLC), 形成功能码异常攻击; 或者当 HMI 进行写操作时将协议中数据单元部分变长, 使得 PLC 寄存器在读取数据时造成缓冲区溢出攻击。由于构建模型的正常流量中未出现这类异常流量, 所以在检测时这些流量都会被映射为未知流量。本文在原始数据集中随机插入“未知”流量, 来模拟此类非语义攻击。

(2) 次序攻击

模拟数据包劫持篡改的方式, 将原始流量序列中流量前后次序颠倒来构成次序攻击。如将正常序列“bcdbedfabcdbedf…”中的“ab”子序列颠倒构成异常序列“bacdbedfbacdbedf…”。

(3) 时序攻击

模拟流量延迟或者缩短周期时间的攻击方式, 将原始流量中的周期时间变为原来的一倍或缩短为之前的一半, 形成时序攻击。

(4) 分支节点攻击

模拟数据包劫持篡改的方式, 调整状态图中的出度大于 1 的分支节点顺序来构造分支节点攻击。比如颠倒正常序列“abcdbedfabcdbedf…”中子序列“bcd”和“bed”的顺序, 可以构建分支节点攻击序列“abedbcdfabedbcdf…”

(5) 子周期重放攻击

模拟数据包劫持篡改的方式, 多次重放子周期序列来构成子周期重放攻击, 如将正常序列“abcdbedfabcdbedf…”中子序列“bcd”多次重复发送来构造子周期重放攻击序列“abcdbcdbcdbedf…”。

6.2 实验评估指标

本文实验数据在原始流量数据基础上合成了异常情况数据, 因此不同的模型构建方法检测过程中会出现不同程度的误报和漏报。为了能够有效评估模型的表现, 本文引入混淆矩阵使用误报率(False Positive Rate, FPR)和漏报率(False Negative Rate)来对模型进行综合分析。

数据集中异常流量为正例, 正常流量为负例, 混淆矩阵中的每一项定义如下:

(1) 真正例(True Positive, TP): 表示实际为正例且被模型检测为正例的个数;

(2) 假正例(False Positive, FP): 表示实际为负例但被模型检测为正例的个数;

(3) 假反例(False Negative, FN): 表示实际为正例但被模型检测为负例的个数;

(4) 真反例(True Negative, TN): 表示实际为负例且被模型检测为负例的个数。

根据上述定义可计算得到误报率、漏报率等评估指标。其中误报率和漏报率定义如公式 1 和 2 所示:

$$FPR = \frac{FP}{TN + FP} \tag{1}$$

$$FNR = \frac{FN}{TP + FN} \tag{2}$$

6.3 实验结果

实验对比了 FMM 与先前的四种建模方法在模型构建上的效果, 参与对比实验的方法如下:

(1) 使用单个 DFA 构建模型(DFA)[17];

(2) 使用两个 DFA 串联构建模型(2-DFA)[18];

(3) 马尔科夫树模型(MMTM)[22];

(4) 使用马尔科夫链和状态转移图构建模型(DTMC)[23];

(5) 融合马尔科夫模型(FMM)。

6.3.1 正常流量建模

在真实的工业生产中, 往往无法根据已知的工业生产流程推导出理想的 DFA 模型来与训练得到的 DFA 模型进行对比, 因此本文通过正常流量构建相应的 DFA 模型, 并将这些正常流量重新送入模型中进行检测, 根据模型检测正常流量所得的误报率可以间接得到建模准确度, 理想情况下这些正常流量应该全部识别为正常。将上述的正常数据集分别用 DFA 方法、2-DFA 方法、DTMC 方法和 MMTM 方法进行训练,将得到的模型进行对比。图 12 展示了在正常数据集中各方法建模误报率的分布对比, 其中各方法误报率结果中出现的异常点属于同一多周期模式混合流量情况下的结果。由于本文实验的流量数据都是从测试平台捕捉的, 所以各通道中流量的周期模式是已知的, 使用人工手段可以确定正常的周期模式来建立理想模型。通过人工对比各模型和理想模型发现, DFA 方法使用单个 DFA 构造模型, 由于流量中存在多种周期和一些人工操作等噪声使得此方法构建的模型很庞杂, 在检测正常流量时会出现较高误报率。2-DFA 方法使用两个 DFA 串联的方式减小了模型规模, 误报率有所降低。MMTM 方法使用树结构来构造模型并使用剪枝的方法去除噪声, 一定程度上降低了误报率。但是上面 3 种方法都没有考虑多周期混合流量的情况, 采用将所有多周期混合流量构建在一个模型中, 使得不同周期的流量相互成为彼此的噪声, 增加了模型误报率。相比之下, DTMC 方法和 FMM 方法采用先分离多周期混合流量再对分离后的多个子周期分别建模的方式来构建异常检测模型。但是 DTMC 方法会将完整的长周期模式分离为多个子周期模式, 影响模型对复杂语义攻击的检测。本文提出的 FMM 方法先将混合周期模式进行分离, 再根据子周期相关性进行融合来还原完整的长周期模式, 所构建的模型最接近理想模型。

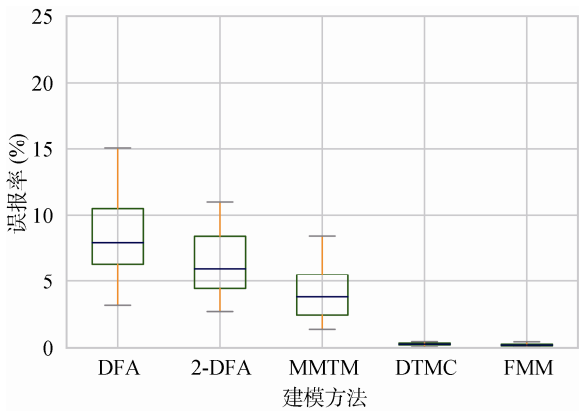


图 12 各方法在原始数据集中的建模误报率

Figure 12 Modeling false alarm rate of each method in the original data set

6.3.2 异常流量检测

使用训练得到的模型依次检测以上 5 种合成的攻击数据集, 将检测结果与本文提出的 FMM 进行对比, 其结果如表 3 所示。由表 3 可知, 本文所提的 FMM 方法与 DFA 方法、2-DFA 方法、MMTM 方法和 DTMC 方法相比具有更完整的检测能力。这 5 种方法都是基于对正常流量构建行为模型来检测异常攻击, 因此在非语义攻击检测方面检测能力相同, 但是在语义攻击检测方面差异很大。首先, 由于 5 种方法都包含状态转移的合法性判断, 所以都能检测出次序攻击引起的状态转移异常。然后, 由于 DFA 方法、2-DFA 方法和 DTMC 方法没有在模型中加入时间间隔信息, 因此无法检测时序攻击。其次, 因为 2-DFA 方法和 DTMC 方法会将完整的长周期模式分离为多个子周期模式, 使得模型无法检测分支节点攻击。最后, 5 种方法中只有 DTMC 方法和 FMM 方法对多周期混合流量进行周期分离, 但是 DTMC 方法会将完整的长周期模式分离为多个子周期模式且 DTMC 方法没有加入时间间隔信息, 使得模型无法检测子周期重放攻击。由上述分析可知, FMM 方法对多周期混合流量进行了周期分离且对子周期进行再次融合防止完整的长周期模式被错误分离, 并在模型中加入时间间隔信息, 与其他 4 种方法相比 FMM 方法能检测更加复杂的语义攻击。

表 3 各方法攻击检测结果对比

Table 3 Comparison of attack detection results of various methods

模型	非语义攻击	次序攻击	时序攻击	分支节点攻击	子周期重放攻击
DFA	检测成功	检测成功	检测失败	检测成功	检测失败
2-DFA	检测成功	检测成功	检测失败	检测失败	检测失败
MMTM	检测成功	检测成功	检测成功	检测成功	检测失败
DTMC	检测成功	检测成功	检测失败	检测失败	检测失败
FMM	检测成功	检测成功	检测成功	检测成功	检测成功

图 13 展示了各方法针对不同类型攻击的检测结果。子图(a)为非语义攻击中各方法检测结果, MMTM 方法和 DTMC 方法误报率较低。分析可知 DFA 方法、2-DFA 方法和 MMTM 方法没有对多周期混合流量进行分离使得各子周期流量互为噪声提高了模型误报率。由于五种方法都是基于正常流量构建行为模型, 所以在检测未知流量时, 漏报率都接近 0。子图(b)为次序攻击中各方法检测结果, 由于 DFA 在复杂周期流量中检测到次序攻击时状态转移可能发生错误转移造成较高的误报率, DTMC 方法将复杂周期分离为多个子周期间接降低了周期复杂度, 所以 DTMC 方法误报率最低, 但是 FMM 方法的漏报率最低。子图(c)为时序攻击的检测结果, FMM 方法在误报率和漏报率方面都低于 MMTM 方法。MMTM 和 FMM 方法都在模型中加入时间间隔来检测时序攻击, 但是由于 MMTM 方法没有将复杂周期流量进行分离, 所以误报率较高。子图(d)为分支节点攻击检测结果, 相比其他两种方法, FMM 方法的误报率和漏报率更低。子图(e)为子周期重放攻击的检测结果, FMM 方法漏报率低, 但是误报率偏高。综上可知, FMM 方

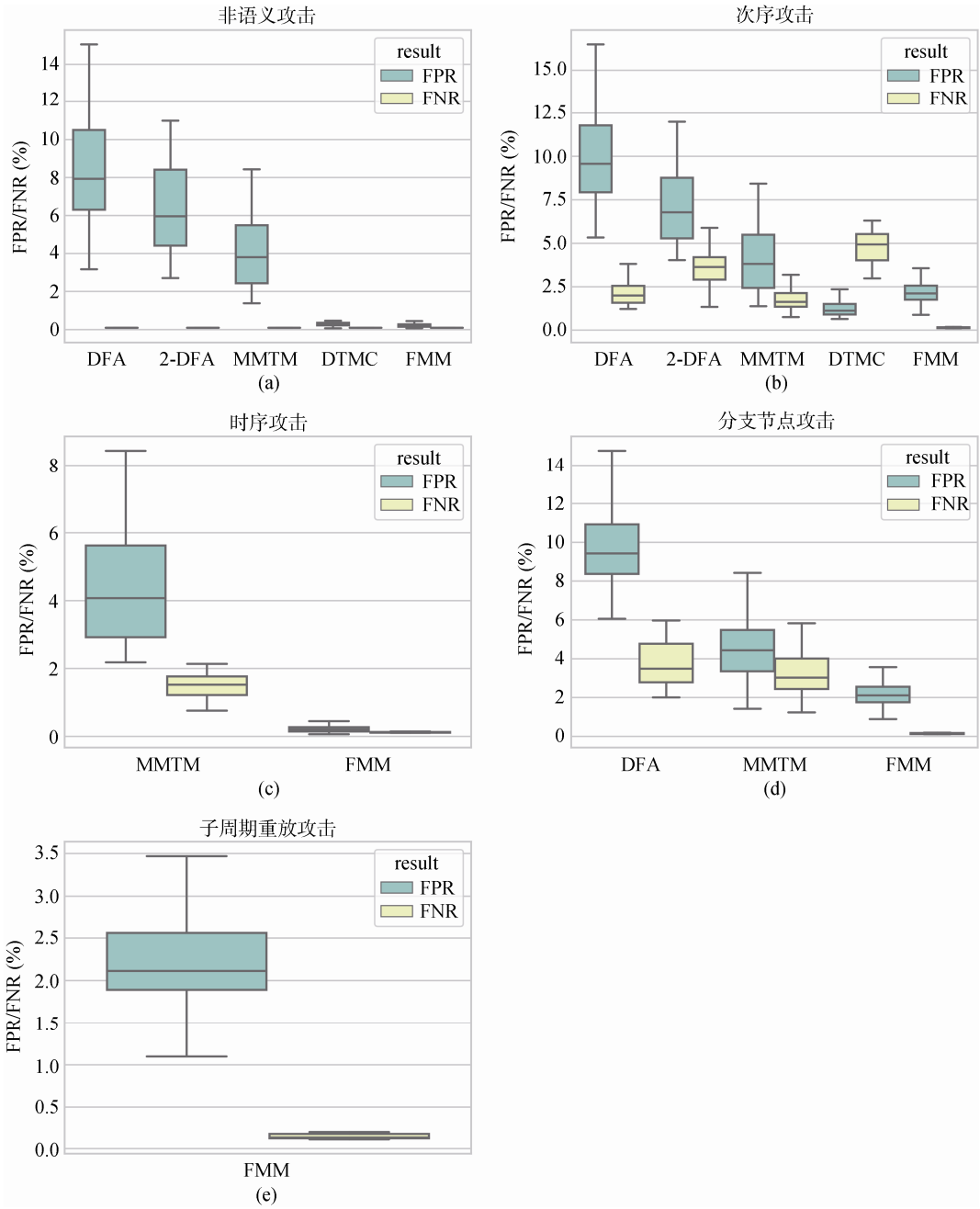


图 13　各方法针对不同类型攻击的检测结果对比

Figure 13　Comparison of detection results of various methods for different types of attacks

法在可以检测更多类型的攻击，且在多数情况下都有较低的误报率和漏报率。

图 14 比较了 5.2 节中不同时间阈值 $durationThreshold$ 下检测模型的误报率和漏报率。分析可知，$durationThreshold$ 由 0.001 增大到 0.014 时，误报率呈现缓慢下降趋势，在 0.01 以后误报率趋向于 0。而漏报率在 0.001 到 0.013 间都趋于 0,在 0.013 以后呈现上升趋势。因此，为了降低检测模型的误报率和漏报率，本文将 $durationThreshold$ 设置为 0.013。

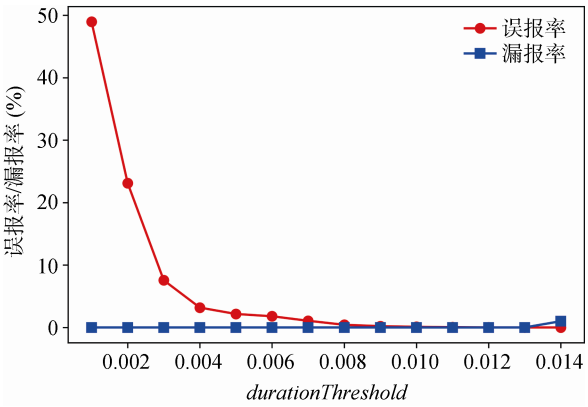


图 14 不同时间阈值 $durationThreshold$ 下误报率和漏报率

Figure 14 False alarm rate and false alarm rate under different time threshold $durationThreshold$

7 结语

本文针对 SCADA 系统多周期混合流量面临的语义攻击问题，充分利用工业流量高周期性和高相关性的特点，提出基于融合马尔科夫模型的异常检测方法。该方法首先深度解析报文语义并将原始流量序列映射为 hash 字符串序列，然后根据字符串序列间的相关性生成状态转移图。接下来，根据状态转移图间各状态的出入关系和频率将子周期符号进行分类并依次构建 DFA 模型。为了检测更多语义攻击，该方法根据子周期间的出入关系和模型误报率将错误分解的长周期模式进行融合并在每个 DFA 模型的节点中加入时间间隔信息。实验结果表明，该方法相比于现有的异常检测方法可以检测更多类型的语义攻击并且检测模型的误报率和漏报率更低。

参考文献

[1] Falliere N, Murchu L O, Chien E. W32. stuxnet dossier[J]. *White paper, Symantec Corp., Security Response*, 2011, 5(6): 29.

[2] Masood R, Um-e-Ghazia, Anwar Z. SWAM: Stuxnet Worm Analysis in Metasploit[C]. *2011 Frontiers of Information Technology*, 2011: 142-147.

[3] Milenkovic N, Damjanovic M, Ristic M. Study of Heavy Metal Pollution in Sediments from the Iron Gate (Danube River), Serbia and Montenegro[J]. *Polish Journal of Environmental Studies*, 2005, 14(6).

[4] Khan R, Maynard P, McLaughlin K, et al. Threat Analysis of BlackEnergy Malware for Synchrophasor Based Real-Time Control and Monitoring in Smart Grid[C]. *Electronic Workshops in Computing*, 2016: 53-63.

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

[6] Spenneberg R, Brüggemann M, Schwartke H. Plc-blaster: A worm living solely in the plc[J]. *Black Hat Asia*, 2016, 16: 1-16.

[7] Nelson N. The Impact of Dragonfly Malware on Industrial Control Systems [J]. *SANS Institute*, 2016.

[8] Hadeli H, Schierholz R, Braendle M, et al. Leveraging Determinism in Industrial Control Systems for Advanced Anomaly Detection and Reliable Security Configuration[C]. *2009 IEEE Conference on Emerging Technologies & Factory Automation*, 2009: 1-8.

[9] Kwon Y, Kim H K, Lim Y H, et al. A Behavior-Based Intrusion Detection Technique for Smart Grid Infrastructure[C]. *2015 IEEE Eindhoven PowerTech*, 2015: 1-6.

[10] Barbosa R R R, Pras A. Intrusion Detection in SCADA Networks[M]. Mechanisms for Autonomous Management of Networks and Services. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010: 163-166.

[11] Barbosa R R R, Sadre R, Pras A. Difficulties in Modeling SCADA Traffic: A Comparative Analysis[C]. *Passive and Active Measurement*, 2012: 126-135.

[12] Song Z W, Liu Z H. Abnormal Detection Method of Industrial Control System Based on Behavior Model[J]. *Computers & Security*, 2019, 84: 166-178.

[13] Kalech M. Cyber-Attack Detection in SCADA Systems Using Temporal Pattern Recognition Techniques[J]. *Computers & Security*, 2019, 84: 225-238.

[14] Song Z W, Zhou R K, Lai Y X, et al. Anomaly Detection Method of ICS Based on Behavior Model[J]. *Computer Science*, 2018, 45(1): 233-239.
(宋站威，周睿康，赖英旭，等. 基于行为模型的工控异常检测方法研究[J]. *计算机科学*, 2018, 45(1): 233-239.)

[15] Shi L Y, Zhu H Q, Liu Y H, et al. Intrusion Detection of Industrial Control System Based on Correlation Information Entropy and CNN-BiLSTM[J]. *Journal of Computer Research and Development*, 2019, 56(11): 2330-2338.
(石乐义，朱红强，刘祎豪，等. 基于相关信息熵和CNN-BiLSTM的工业控制系统入侵检测[J]. *计算机研究与发展*, 2019, 56(11): 2330-2338.)

[16] Zhang Y S, Li X W, Li D, et al. Abnormal Flow Monitoring of Industrial Control Network Based on Convolutional Neural Network[J]. *Journal of Computer Applications*, 2019, 39(5):

1512-1517.
(张艳升, 李喜旺, 李丹, 等. 基于卷积神经网络的工控网络异常流量检测[J]. *计算机应用*, 2019, 39(5): 1512-1517.)

[17] 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.

[18] Kleinmann A, Wool A. Accurate Modeling of the Siemens S7 SCADA Protocol for Intrusion Detection and Digital Forensics[J]. *Journal of Digital Forensics*, *Security and Law*, 2014, 9(2):37-50.

[19] Yoon M K, Ciocarlie G. Communication Pattern Monitoring: Improving the Utility of Anomaly Detection for Industrial Control Systems[C]. *The 2014 Workshop on Security of Emerging Networking Technologies*, 2014.

[20] Yang A, Hu Y, Zhou L, et al. An Industrial Control System Anomaly Detection Algorithm Fusion by Information Flow and State Flow[J]. *Journal of Computer Research and Development*, 2018, 55(11): 2532-2542.
(杨安, 胡堰, 周亮, 等. 基于信息流和状态流融合的工控系统异常检测算法[J]. *计算机研究与发展*, 2018, 55(11): 2532-2542.)

[21] Yang A, Sun L M, Shi Z Q, et al. SBSD: Detecting the Sequence Attack through Sensor Data in ICSs[C]. *2018 IEEE International Conference on Communications*, 2018: 1-7.

[22] Zhang R B, Wu P, Lu Y, et al. Anomaly Detection Algorithm in ICS Based on Mixed-Order Markov Tree Model[J]. *Acta Automatica Sinica*, 2020, 46(1): 127-141.
(张仁斌, 吴佩, 陆阳, 等. 基于混合马尔科夫树模型的 ICS 异常检测算法[J]. *自动化学报*, 2020, 46(1): 127-141.)

[23] Kleinmann A, Wool A. Automatic Construction of Statechart-Based Anomaly Detection Models for Multi-Threaded Industrial Control Systems[J]. *ACM Transactions on Intelligent Systems and Technology*, 2017, 8(4): 1-21.

[24] Barbosa R R R, Sadre R, Pras A. Exploiting Traffic Periodicity in Industrial Control Networks[J]. *International Journal of Critical Infrastructure Protection*, 2016, 13: 52-62.

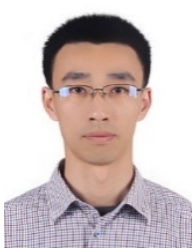
[25] Caselli M, Zambon E, Kargl F. Sequence-Aware Intrusion Detection in Industrial Control Systems[C]. *The 1st ACM Workshop on Cyber-Physical System Security*, 2015: 13-24.

[26] Fovino I N, Carcano A, de Lacheze Murel T, et al. Modbus/DNP3 State-Based Intrusion Detection System[C]. *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, 2010: 729-736.

[27] Ellis J, Fisher D, Longstaff T, et al. Report to the President's Commission on Critical Infrastructure Protection[R]. Defense Technical Information Center, 1997: 19-20.

马标 于 2018 年在金陵科技学院获得本科学位。现在苏州大学软件工程专业攻读硕士学位。研究领域为工控安全，数据分析。Email：20185227071@stu.suda.edu.cn

张重豪 现在苏州大学软件工程专业攻读本科学位。研究领域为深度学习、异常检测。研究兴趣包括：深度学习、数据分析。Email：1819284722@qq.com



贾俊铖 于 2009 年在香港科技大学计算机科学专业获得博士学位。现任苏州大学计算机科学与技术学院副教授。研究领域为工业互联网、物联网。研究兴趣包括：工业流量分析、异常检测。Email：jiajuncheng@suda.edu.cn

胡梦娜 于 2019 年在湖北工业大学计算机科学与技术专业获得本科学位。现在苏州大学计算机技术专业攻读硕士学位。研究领域为深度学习、异常检测。研究兴趣包括：深度学习、网络流量。Email：20195227019@stu.suda.edu.cn

周正寅 于 2020 年在浙江工业大学网络工程专业获得学士学位。现在苏州大学计算机技术专业攻读硕士学位。研究领域为计算机网络。研究兴趣包括深度学习和网络安全。Email：zhou_zhengyin@163.com

杨荣举 于 2002 年在华中科技大学通信工程专业获得硕士学位。现任西门子（中国）有限公司工业信息安全实验室经理，研究领域为工业信息安全。研究兴趣包括：工业网络流量异常检测，安全大数据分析等。Email：rongju.yang@siemens.com