软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2024,35(2):1010−1027 [doi: 10.13328/j.cnki.jos.006943]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

机器人操作系统 ROS 安全性研究综述*

鲁敬敬[1,4], 秦云川[1], 刘志中[1], 唐 卓[1,2], 张拥军[3], 李肯立[1,2]

[1](湖南大学 信息科学与工程学院, 湖南 长沙 410082)

[2](国家超级计算长沙中心, 湖南 长沙 410012)

[3](中国人民解放军军事科学院 国防科技创新研究院, 北京 100850)

[4](湖南大学 重庆研究院, 重庆 401135)

通信作者: 秦云川, E-mail: qinyunchuan@hnu.edu.cn; 李肯立, E-mail: lkl@hnu.edu.cn

摘 要: 机器人日益走进人们的日常生活, 也受到了国内外越来越多的关注. 机器人系统的一个重要特性是安全性, 增强机器人系统的安全性可以保护机器人免受恶意攻击者的入侵. 机器人操作系统的安全性是机器人系统安全性的重要组成部分. 虽然近年来研究人员针对机器人操作系统的安全性做了许多研究工作, 但遗憾的是, 安全性目前还没有得到足够的重视. 为了引起人们对机器人系统安全性更多的关注, 同时帮助人们快速了解当前主流机器人操作系统 ROS (robot operating system) 的安全性解决方案, 对 ROS 的安全性进行系统的调研和总结. 一方面, 深入分析 ROS 的安全特性, 总结 ROS 中已知的安全问题. 另一方面, 对近年来 ROS 安全性相关的研究进行分类分析和概括总结, 并从机密性、完整性和可用性这 3 个方面, 对众多 ROS 的安全性解决方案进行比较. 最后, 对 ROS 安全性研究的前景进行展望.

关键词: 机器人; ROS; ROS2; 安全性

中图法分类号: TP316

中文引用格式: 鲁敬敬, 秦云川, 刘志中, 唐卓, 张拥军, 李肯立. 机器人操作系统ROS安全性研究综述. 软件学报, 2024, 35(2): 1010–1027. http://www.jos.org.cn/1000-9825/6943.htm

英文引用格式: Lu JJ, Qin YC, Liu ZZ, Tang Z, Zhang YJ, Li KL. Survey on Security of Robot Operating System ROS. Ruan Jian Xue Bao/Journal of Software, 2024, 35(2): 1010–1027 (in Chinese). http://www.jos.org.cn/1000-9825/6943.htm

Survey on Security of Robot Operating System ROS

LU Jing-Jing[1,4], QIN Yun-Chuan[1], LIU Zhi-Zhong[1], TANG Zhuo[1,2], ZHANG Yong-Jun[3], LI Ken-Li[1,2]

[1](College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China)

[2](National Supercomputing Center in Changsha, Changsha 410012, China)

[3](National Innovation Institute of Defense Technology, Academy of Military Sciences, Beijing 100850, China)

[4](Research Institute of HNU in Chongqing, Hunan University, Chongqing 401135, China)

Abstract: Robots are increasingly entering people's daily life and are receiving more and more attention in China and abroad. One of the important characteristics of robotic systems is security, and enhancing the security of robotic systems can protect robots from malicious attackers. The security of robot operating system (ROS) is an important part of the security of robotic systems. Although researchers have done a lot of research work on the security of ROSs in recent years, unfortunately, security has not received enough attention yet. In order to draw more attention to the security of robotic systems and help people quickly understand the security solutions of the current mainstream ROS, this study systematically investigates and summarizes the security of ROSs. On the one hand, this study analyzes the security features of ROSs and discusses the known security problems in ROSs. On the other hand, this study categorizes and summarizes

* 基金项目: 国家自然科学基金 (62172152); 国家重点研发计划 (2020YFB0204802, 2021YFA1000604)
收稿时间: 2021-10-03; 修改时间: 2022-05-26, 2022-11-03; 采用时间: 2023-04-03; jos 在线出版时间: 2023-09-27
CNKI 网络首发时间: 2023-10-12

the research related to the security of ROSs in recent years and compares the security solutions of ROSs in terms of confidentiality, integrity, and availability. Finally, this study prospects the future of security research on ROS.

Key words: robot; ROS; ROS2; security

随着计算技术、控制技术和人工智能技术的发展, 机器人逐渐被应用于诸多领域, 如图 1 所示. 在家庭服务领域, 机器人管家、扫地机器人正致力于为居民提供智能化的生活体验[1−3]. 在医疗健康领域, 手术机器人、康复机器人的智能化救治提高了治愈率, 帮助越来越多的人重获健康[4−6]. 在公共安全领域, 巡逻机器人、安保机器人在保障公共安全方面发挥着越来越重要的作用[7,8]. 国际机器人联合会发布的《Top 5 Robot Trends 2021》[9]显示, 在 2010–2019 年里工业机器人的年安装量增加了 3 倍多. 2018–2019 年, 专业服务机器人的全球销售额增长了 32%, 达到 112 亿美元. 这些数据充分显示了机器人市场蕴藏着巨大的潜力. 机器人可以高效完成流水线上的重复工作, 提高社会生产效率, 节约人力资本. 波士顿咨询公司曾估算, 到 2025 年, 更多的工业岗位将被机器人自动化取代, 制造业的人力成本将被压缩 16%[10]. 机器人还可以代替人类探索未知或危险领域, 促进全球科学技术进步. NASA 开发的“Robonaut 2”机器人可以驻守太空空间站, 协助宇航员完成太空探索任务[11,12]. 可以预见, 未来机器人必定会以更多全新的姿态走进大众视野, 并逐渐成为人们日常生活的重要部分.

(a) 扫地机器人 iRobot (b) 外科手术机器人 da Vinci (c) 安保机器人 K5

图 1 走进人类生活的机器人

然而, 机器人的大量使用可能会给人类带来许多潜在的安全威胁. 特别是在国防、医疗等涉及人类的关键领域, 机器人系统的安全性显得尤为重要[13,14]. 机器人系统的安全性有两方面含义, 一方面是指保护环境不受机器人系统的影响[15], 即防止机器人系统故障给人类及周围环境带来安全威胁. 机器人的任何系统故障都可能导致灾难性的后果. 例如, 2016 年, 谷歌的一辆无人车因为发生系统错误与一辆公交车相撞, 不仅损坏了当时的道路基础设施, 还给路人带来了身体和精神上的伤害. 另一方面是指保护机器人系统不受环境的影响[16,17], 主要是指防止恶意攻击者的非法入侵给机器人系统带来安全隐患. 恶意攻击者一旦突破机器人系统的安全防护, 可能会给系统带来严重的安全威胁. 例如, 攻击者可能会伪装成合法用户潜伏在机器人系统中伺机窃取机密信息, 甚至可能直接向机器人发送指令操纵机器人的行动. 本文所讨论的机器人系统的安全性指的是第 2 个方面的含义, 即机器人系统抵御来自攻击者恶意入侵的能力.

相比于传统计算平台, 机器人系统面临的安全威胁种类更多, 安全问题所引起的后果也更严重. 下面将进行具体分析. 机器人主要由计算系统、感知系统和执行系统构成, 是机械域与信息域一体化的产物[18], 这意味着机器人有着复杂的软硬件构成环境, 也暗示了机器人面临的安全威胁复杂多样. 首先, 机器人是建立在传统计算平台之上的, 因此要面临与传统计算平台相同的安全威胁和后果. 传统计算平台面临的安全威胁主要包含两个方面. 一是来自信息空间的攻击, 主要包括电子进攻、网络攻击[19−21]等. 来自信息空间的攻击可以干扰和破坏计算平台信息系统. 其中, 攻击信息装备和信息的可用性可以破坏、阻塞或扰乱计算平台的功能. 攻击信息传递过程和信息内容的完整性可以讹误和欺骗计算平台. 攻击信息传递过程和信息内容的保密性, 可以从计算平台中窃取和利用机密信息. 二是来自物理空间的攻击, 常见的攻击方式主要有冷启动攻击和探针攻击. 其中, 冷启动攻击通过获取计算机运行时内存快照, 实现获取密钥等机密信息. 微探针攻击则是通过直接访问芯片表面, 达到观察、操作和干扰设备的目的. 不难看出, 传统计算平台安全性是机器人安全性的重要组成部分, 忽视其安全性将导致不可估量的严重后果. 除了传统计算平台, 机器人通常还会接入一种或多种传感器用于感知环境数据, 部分机器人还配备了机械手臂或行走装置用于开展具体行动. 使用这些设备可以丰富机器人的功能、提高机器人的智能化水平, 但同时也为

攻击者提供了更多的攻击入口, 从而给机器人带来了额外的安全隐患. 例如, 恶意攻击者可以通过向传感器传递虚假环境信息, 从而达到扰乱和控制机器人行为的目的, 对于那些带有机械臂和行走装置的机器人, 攻击者一旦掌握了其控制权, 很可能会对人类及周围环境带来直接的物理伤害, 威胁人民群众的生命安全和身体健康, 进而扰乱社会秩序, 破坏社会安定. 因此, 为了机器人产业的可持续发展, 促进机器人更好地服务于人类, 研究机器人系统的安全性十分必要.

针对机器人系统安全性的研究由来已久. 早在 1950 年, Asimov 就在他的小说中创造性地提出机器人三大定律, 通过给机器人设立规则, 以确保机器人系统的安全性[22]. 近几年, 学术界针对机器人系统的安全性也开展了许多研究工作. 由于基于信标实时定位系统提供的数据在有攻击者和没有攻击者两种情况下存在统计学差异, Guerrero-Higueras 等人[23]提出一种攻击检测方法. Quarta 等人[24]首次系统分析了工业机器人控制器的安全性. 尽管如此, 无论是在学术界还是在工业界, 机器人系统的安全性都没有得到足够的重视, 机器人专家不会优先考虑机器人带来的安全和隐私风险, 因为机器人的投资人和开发商都更加注重机器人的功能性而不是安全性[25]. 庆幸的是, 近年来, 研究人员针对当前主流机器人操作系统 ROS[26]的安全性开展了许多研究工作, 一些研究提出的 ROS 安全性解决方案在提高机器人系统安全性方面发挥着重要作用. 本文对 ROS 系统安全性的相关研究进行了全面分析和总结.

本文首先分析了导致 ROS 缺乏安全性的原因, 总结了当前 ROS 系统中存在的主要安全问题. 然后, 总结回顾了研究人员在评估和增强 ROS 系统安全性方面所做的努力, 并进一步从机密性、完整性、可用性这 3 个方面对 ROS 安全性解决方案进行了比较. 最后, 本文对 ROS 安全性研究的前景进行了展望.

本文第 2 节对 ROS 进行了简要介绍. 第 3 节阐述了 ROS 缺乏安全性的事实, 深入剖析了其原因, 并概括介绍了 ROS 中已暴露的安全问题. 第 4 节对 ROS 安全性的相关研究进行了分类总结和优缺点分析, 特别是对 ROS2 的安全特性进行了深入剖析. 第 5 节对本文的主要工作进行了总结, 并基于当前 ROS 安全性的研究现状展望了 ROS 安全性研究的前景.

1 ROS 简介

ROS 是一个强大而灵活的开源机器人操作系统, 目前由美国开放机器人基金会负责维护.它提供了硬件抽象、设备驱动、函数库、可视化工具、消息传递和软件包管理等诸多功能[27]. ROS 最初是为了在机器人领域提高代码的复用率, 经过 14 年的发展, 良好的 ROS 社区生态以及丰富的功能包为机器人的开发提供了巨大的便利, 也推动 ROS 成为机器人领域事实上的标准[28]. ROS 很早就被应用到机器人领域, 很多知名的机器人开源库都是开源贡献者基于 ROS 开发的, 例如基于 Quaternion 的坐标转换[29]、基于 3D 点云的物体识别[30]以及 SLAM 定位算法[31,32]. 随着社会发展对提高生产效率产生强烈需求, 军事、教育、工业等诸多领域都根据实际需要开发了基于 ROS 的机器人, 图 2 所示的美国 Endeavor Robotics 公司研制的 PackBot[33]军用机器人、韩国 Yujin 公司开发的 TurtleBot2[34]机器人以及在学术领域广泛应用的 NAO[35]机器人等国际著名机器人都是利用 ROS 框架开发的. ROS 一直保持持续的版本更新, 撰写本文时 ROS Noetic Ninjemys 版本已经正式发布[36].

(a) PackBot (b) TurtleBot2 (c) NAO

图 2 基于 ROS 的机器人

从软件构架的角度来看, ROS 是一种基于匿名发布/订阅机制[37]和消息传递通信的分布式多进程框架, 开发者可以根据功能把软件拆分成为各个模块, 每个模块只是负责读取和分发消息, 模块间通过消息关联. 发布/订阅机

制具有松耦合特性, 并且支持异步通信和分布式组件交互等多种功能, 因此, 该机制在众多领域得到了广泛应用. ROS 中的消息传递通信是通过节点实现的点对点通信. 节点间的消息通信分为 3 种, 如图 3 所示, 它们分别是采用单向消息发送/接收方式的话题 (topic)、采用双向消息请求/响应方式的服务 (service) 以及采用双向消息目标 (goal)/结果 (result)/反馈 (feedback) 方式的动作 (action)[38,39]. 开发人员在对 ROS 进行编程时要根据实际需求选择合适的话题、服务、动作和参数, 以实现具体功能.

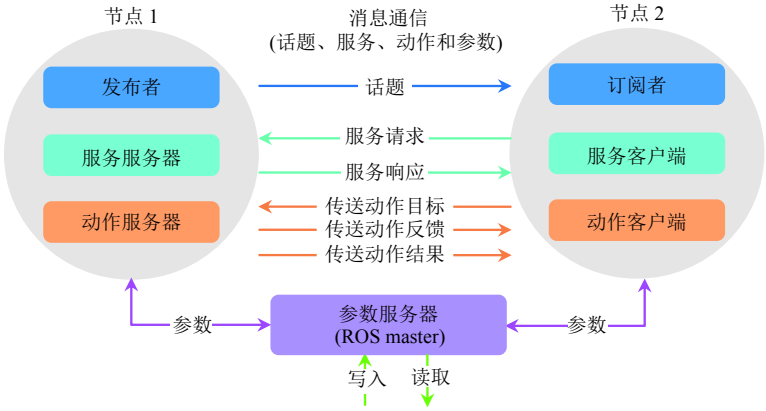

图 3 ROS 节点间消息通信图

2 ROS 安全性分析

本文所讨论的 ROS 安全性指的是 ROS 系统确保系统中传输的数据在产生、传输、存储等各个环节的机密性、完整性、可用性不被破坏, 保证 ROS 系统连续可靠正常运行的能力. 因为 ROS 最初是为科研服务的, 所以开发者在进行 ROS 架构设计时并没有充分考虑系统的安全性. 这直接导致 ROS 中隐藏着很多潜在的安全隐患, 也给基于 ROS 的机器人带来了很多不可预料的安全威胁, 进而限制了 ROS 的进一步应用与拓展[40].

导致 ROS 缺乏安全性的一个重要原因是 ROS 基于匿名发布/订阅机制. 实现该机制的系统容易遭受各种安全威胁, 因为发布者无法确保发布的消息在传输过程中不被攻击者访问和更改, 而订阅者无法验证收到的数据的完整性[41−43]. 在数据传输缺乏机密性和完整性的情况下, 攻击者可能会窃听已发布的数据或修改已订阅的消息. 这可能会导致机器人行为的改变, 从而产生无法预料的后果.

本文系统调查了 ROS 的安全特性, ROS 中已为人知的安全问题主要包括以下 4 个方面[25,44−47].

(1) ROS 中节点之间的通信为明文通信. 尽管未加密的文本在易于使用、方便调试、不过多消耗系统性能等方面有诸多优势, 但与此同时, 攻击者可以利用恶意节点轻易拦截和破译消息内容, 监听话题并趁机将虚假消息注入机器人系统, 给系统带来不可预测的安全威胁.

(2) ROS 的模块化特性暗藏着诸多安全隐患. 模块化特性在方便研究人员进行机器人开发的同时也暴露了许多安全缺陷, 例如, ROS 使用不安全的 TCP 端口, 并且缺乏身份验证机制, 攻击者可能会利用这些缺陷对系统进行中间人攻击, 甚至破坏机器人系统.

(3) ROS 采用弱授权方案. ROS 不进行消息发送方验证, 不检查数据的完整性和真实性, 也不定义访问级别. 实际上, 一些远程客户机不应该拥有对整个 ROS 系统的访问权限. 可以想象, 如果任何用户都可以向机器人发送任意指令, 并且直接操纵机器人, 那么机器人系统将面临巨大的安全威胁.

(4) ROS 不提供服务质量 (quality of service, QoS). 每个节点负责管理自己的通信, 类似的消息没有被压缩, 也没有任何减少网络通信的努力. 这可能会导致网络延迟和网络阻塞等网络安全问题得不到及时解决, 从而阻碍时间关键型应用程序的开发[48,49].

3 ROS 安全性研究

近年来, 基于 ROS 框架开发的机器人逐渐被应用到日常生活场景, 用于提高生产效率和生活质量. 然而, 一些机器人的安全状况却令人堪忧. 由于部分机器人系统缺乏安全性考虑或者安全防护层薄弱, 恶意攻击者往往可以轻易绕过或突破系统安全防护进而操纵机器人, 这会对机器人周围的人和物带来严重的安全威胁. 为了满足市场在机器人安全性方面的新需求, 近几年, 学术界和工业界在揭露 ROS 的安全漏洞以及为 ROS 提供安全性解决方案等方面进行了许多卓有成效的研究, 本节将对 ROS 安全性相关的研究进行分类和总结.

3.1 测试和评估 ROS 安全性的研究

虽然增强 ROS 系统的安全性十分迫切, 但是在制定合理的安全性解决方案之前, 定性和定量测试和评估 ROS 的安全性十分必要. 一方面, 对 ROS 进行安全性测试和评估可以揭露使用 ROS 框架构建的信息物理系统中潜在的安全漏洞, 从而帮助 ROS 用户了解他们使用的机器人所面临的安全威胁, 另一方面, 只有弄清楚 ROS 系统有哪些安全漏洞, 研究人员才能选择合适的信息安全技术增强机器人系统的安全防护能力.

3.1.1 渗透测试 ROS 系统

渗透测试是一种常见的安全测试实践, 它通过模拟恶意攻击者的攻击方法, 来测试应用程序是否存在潜在的安全隐患. 渗透测试过程会对系统漏洞或者技术缺陷进行主动分析, 然后输出渗透测试报告, 应用所有者通过渗透测试报告可以知晓系统中存在的安全隐患. 将渗透测试集成到开发周期, 有利于开发出更加安全的应用程序.

为了主动捕捉和分析基于 ROS 的机器人系统中的安全漏洞, 研究人员开发了专门针对 ROS 的渗透测试工具. Dieber 等人[50]引入了两个 ROS 应用程序渗透测试工具: ROSPenTo 和 Roschaos. 这两个工具可以利用 ROS API 中的漏洞, 支持手动或自动地对正在运行的 ROS 应用程序执行攻击, 进而分析和操纵 ROS 应用程序. 文献 [50] 首先对这两个工具及其使用方法进行了详细介绍, 然后使用 ROSPenTo 和 Roschaos 工具演示了攻击 ROS 应用程序的过程. 采用的攻击类型包括: (1) 隐形发布者攻击: 攻击者向运行中的 ROS 应用程序注入虚假数据, 并欺骗订阅者使用虚假数据, 而不被任何其他应用程序节点或 ROS 主节点注意到. (2) 服务隔离攻击: 攻击者将特定服务与其他 ROS 网络隔离, 以便攻击者可以调用, 而其他 ROS 节点无法调用这些服务. (3) 恶意参数更新攻击: 攻击者可以随意操纵 ROS 参数服务器. 文献最后作者展示了这些攻击可能产生的影响, 并给出了抵御这些攻击的几种对策, 包括使用 roswtf 工具检测 ROSPenTo 的攻击模式, 使用 SROS 增强 ROS 系统安全性, 以及使用具有安全特性的 ROS2 等. 同样是为了协助进行 ROS 安全性研究, Rivera 等人[51]提出了一种新的开发工具 ROSploit, 用于模拟对 ROS 系统的攻击. 它还提出了一个新的 ROS 系统安全模型, 并将该模型作为 ROS 系统安全分析的基础. 与上述的 ROSPenTO 工具不同的是, ROSploit 能够分析带有未知主节点的系统, 并能够部分分析基于端口号的 ROS 系统.

无论是 ROSPenTo、Roschaos 还是 ROSploit, 都可以方便开发人员对他们开发的基于 ROS 的机器人系统进行安全测试, 从而提前发现系统漏洞, 尽早进行安全防护, 最终开发出安全可靠的系统.

3.1.2 评估 ROS 系统安全性

除了使用专门的渗透测试工具, 研究人员还利用了其他方式对 ROS 系统的安全性进行了有效评估. 2012 年, 在 DEF CON 20 信息安全大会上举行的信息物理安全竞赛中, McClean 等人[52]通过引入一个名为信息物理安全蜜罐的研究工具, 首次对 ROS 系统的安全性进行了实测和展示. 作者将一个基于 ROS 的小型汽车式机器人配置成信息物理安全蜜罐, 然后进行了 ROS 安全性实验. 该蜜罐与传统信息安全蜜罐相比, 既有相同之处, 也有不同之处. 首先, 该蜜罐与传统信息安全蜜罐一样, 都具有捕获和分析攻击行为的功能, 这样就可以了解攻击工具和方法, 推测攻击意图和动机, 进而了解机器人所面临的具体的安全威胁. 其次, 与传统信息安全蜜罐不同的是, 该蜜罐还额外配备了传感器和执行器, 以此增加系统漏洞类型, 扩大攻击者的攻击范围. 为了模拟真实的应用场景, 作者在机器人中有意保留了 ROS 的已知漏洞, 包括明文通信、未受保护的 TCP 端口和未加密的数据存储等. 作者邀请了众多与会者参与到实验中, 并让他们对用作蜜罐的机器人进行了普通的、低成本、低开销的网络攻击. 实验参与者中不乏信息安全领域的专家和熟悉 ROS 的专业人士. 作者在实验过程中对信息安全蜜罐的网络流量进行了

监控和记录, 用作后续使用. 实验结果证明了网络攻击的有效性, 因为攻击不但直接导致用于控制蜜罐的固态硬盘出现了故障, 还使蜜罐上的软件故障保护失效, 进而导致前后摄像头无法再发送数据. 不仅如此, 熟悉 ROS 的攻击者还轻易地向机器人系统中注入了虚假消息并直接操纵了机器人. 会后作者还通过其他实验验证了这样一个猜想: 即使是那些缺乏信息安全经验的人, 也可以使用 Wireshark 等免费工具嗅探和破译机器人系统中的通信消息, 甚至在数据包中注入虚假信息, 进而对机器人系统进行中间人攻击. 此外, 作者通过诸多实验证明了检测恶意攻击比预期的困难得多, 因为攻击者可以故意将攻击行为伪装成简单的 bug, 这会导致很难区分信息物理安全漏洞以及硬件或软件漏洞. 因此, 作者认为为了保证信息物理系统的安全, 具备信息安全、物理安全以及机器人技术等知识背景的新型安全专业人员是必不可少的. 此外, 针对 ROS 消息的纯文本性质带来的安全威胁, 作者认为要对接收到的消息进行身份验证并验证发送方的身份, 并且建议利用标准的基于散列的 HMAC (hash-based message authentication code) 实现验证功能.

基于对 ROS 系统缺乏安全性的基本认识, 再考虑到机器人研究的跨学科特性给机器人安全性研究带来了诸多困难, 特别是评估机器人系统安全性的工具和标准十分欠缺, Vilches 等人[53]深刻意识到综合的机器人安全评估工具对促进机器人安全性的重要作用, 并提出了机器人安全框架 RSF. RSF 是一种能够全面评估机器人系统安全性的标准化方法, 当然也包含基于 ROS 的机器人系统. RSF 是一个开源框架, 可以识别、分类和报告机器人系统的中漏洞. RSF 主要由 4 层组成, 包括物理层、网络层、固件层和应用层. 作者将每一层划分为数个方面, 为每个方面指定了安全评估标准, 并且描述了每个标准的评估目标、评估原理以及评估方法. 为了方便开发人员理解和使用该框架, 作者开发了一套 RSF 专用的术语, 还提供了该框架的使用指南, 并用实例演示了该方法评估机器人安全性的全过程.

为了揭示和衡量机器人研究的安全状况, DeMarinis 等人[54]以在机器人研究中广泛使用的 ROS 为例, 使用互联网端口扫描工具 ZMap, 对整个 IPv4 地址空间进行了多次扫描, 以识别暴露在公共互联网上的 ROS 主机. 作者总共进行了 3 次扫描, 每次扫描都观察到超过 100 个 ROS 实例, 跨越 28 个国家, 其中超过 70% 的观察实例使用属于大学网络或研究机构的地址. 作者对扫描结果进行了定量和定性评估. 作者定量评估了各种类型的传感器和执行器的话题数量, 并且定性地远程访问和控制了某个机器人. 评估结果证明了许多 ROS 用户的安全意识薄弱, 他们将机器人暴露在一个充满安全威胁的公共网络环境中, 恶意攻击者可以轻易地获取想要的话题数据或者操纵机器人, 这可能会导致十分严重的后果. 因此, 作者提醒机器人使用者, 无论是在工业生产中还是学术研究中, 都应该将机器人的安全性作为一个重要因素进行考虑. 此外, 作者还给出了几个提高机器人安全性的可行建议, 包括使用具有安全特性的 ROS2, 为 ROS 的安全性提供临时解决方案, 例如建立防火墙、使用工具检查网络暴露、远程访问可以通过 VPN (virtual private network) 实现等.

上述测试和评估 ROS 系统安全性的研究工作不仅揭露了 ROS 中的诸多安全问题, 例如, 明文通信、TCP 端口未收到保护等, 而且还证实了即使是缺乏信息安全经验的人也可以轻易成功入侵 ROS 系统. 因此增强 ROS 安全性十分重要, 迫在眉睫.

3.2 增强 ROS 安全性的研究

机器人智能化程度的提升离不开对网络的应用, 现在越来越多的机器人承载着直接面向互联网的应用程序. 因此, 网络攻击所带来的安全威胁也就成了机器人必须面对的安全威胁之一. 针对机器人的网络攻击通常指对机器人计算机网络的软件实施攻击, 包括计算机网络入侵、计算机病毒攻击和引发错误攻击等多种攻击方式. 攻击者可以利用机器人系统中的网络安全漏洞对系统和资源进行攻击, 以非授权方式达到破坏、欺骗和窃取数据信息等目的. 因此, 提高机器人系统的网络安全是提高其整体安全性的重要方面. 当前, ROS 安全性解决方案的着力点主要集中在保障机器人系统的网络安全方面, 目的在于防止 ROS 遭受网络攻击. 本节首先对增强 ROS 安全性的相关研究进行了分类, 分类标准参考的是 OSI (open system interconnection) 模型, ROS 安全性解决方案主要集中在 OSI 模型的应用层、传输层和网络层, 如图 4 所示. 然后本节概括了每个方案的主要研究内容, 并对其优势和不足进行了分析和阐述.

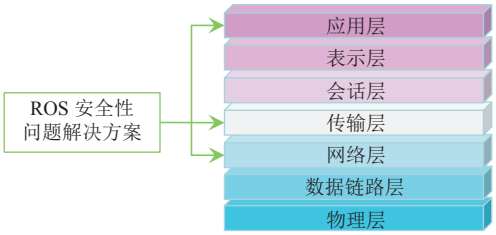

图 4 ROS 安全性解决方案在 OSI 模型中的分布图

3.2.1 应用层安全加固

在应用层实现安全加固是指不修改 ROS 框架, 即将 ROS 视为黑盒, 并在 ROS 之上, 也就是 OSI 模型中的应用层实现安全措施.

Lera 等人[19]使用 3DES 加密算法对 ROS 进程间的通信数据进行了加密, 并且增加了两个独立的节点分别负责完成加密和解密任务, 从而为 ROS 提供安全防护. 作者还搭建了专门的实验平台, 并从系统的 CPU 消耗和网络流量两个方面评估了数据加密对机器人系统性能的影响. 实验结果表明, 数据加密会显著增加机器人系统的 CPU 性能开销和通信负载. 不难看出, 虽然使用 3DES 加密算法可以避免 ROS 进程间的明文通信, 提高 ROS 的安全性, 但同时也影响了系统的实时性.

受上个文献的启发, Portugal 等人[55]使用了比 3DES 更快的 AES 加密算法对节点间传输的数据进行加密, 加密/解密节点是使用 Crypto++库用 C++编码的. 与上述两个文献不同的是, Rodríguez-Lera 等人[56]研究了使用 3 种著名的加密算法 (3DES、AES 和 Blowfish) 对 ROS 节点间交换的 TCPROS 消息进行加密. 作者还搭建了实验环境, 并且从网络开销、CPU 使用率、自主效应 3 个方面对不同算力和 3 种加密算法对机器人系统性能的影响进行了定量分析. 实验数据表明, 加密算法要应用到合适的场景中才能在提供安全防护的同时又不对机器人系统的性能产生严重影响. 因此, 研究人员在寻求最优的加密算法以增强 ROS 安全性时, 要在加密算法和机器人自主性之间进行权衡, 其中, 机器人自主性包括机器人使用的传感器类型、所需的安全级别、ROS 节点之间交换的数据量以及机器人的计算能力等. 为了帮助研究人员选择最优的安全方案, 作者提供了一个模型, 该模型可以帮助研究人员根据传感器的数据类型和机器人系统的计算能力选择合适的加密算法.

上述 3 个文献都对 ROS 系统中的通信数据实施了加密策略, 在一定程度上确保了 ROS 系统中数据传输的机密性. 然而, 因为数据加密并不改变 ROS 核心, 任意节点或客户端对 ROS 主服务器的请求和查询都会得到响应和满足. 所以仅对传输数据进行加密并不是十分安全的解决方案. 因此, 一些研究人员还考虑了针对数据传输过程进行身份验证和授权来确保通信过程的安全性.

Dieber 等人[57]首次提出并实现了在应用层增强 ROS 安全性的解决方案. 作者实现了一个专用的身份验证服务器, 该服务器可以跟踪哪些 ROS 节点可以订阅或发布到某个话题, 还负责管理节点的身份验证, 并生成针对特定话题的加密密钥. 此外, 作者还将发布/订阅模式重新定义为广播通信模式, 通过广播加密实现安全可靠的通信, 从而确保数据的机密性和完整性. 为了更好地描述加密过程, 作者根据发布者和订阅者的生命周期将提出的应用层安全体系结构描述划分为数个阶段, 并描述了每个阶段的加密操作. 最后, 作者评估了额外的加密开销, 包括额外的数据开销和计算开销. 该方案有效解决了 ROS 中的一些安全漏洞. 身份验证服务器只会将特定话题的加密密钥分发给授权的节点, 确保了订阅者只有在获得正确的解密密钥的情况下才能读取消息内容. 这样可以防止未经授权的节点发布和订阅话题, 从而防止恶意节点试图向话题中注入虚假数据和窃听消息内容. 该方案也有很多不足之处, 例如, 没有对消息头加密使得攻击者依然可以对一些消息类型进行频率分析, 并且该方法无法阻止恶意节点加入 ROS 图, 恶意节点可以发布消息并对特定节点展开拒绝服务攻击等.

在医疗领域, 许多医疗机器人系统是直接面向患者的, 因此机器人系统的安全性至关重要. 基于这样的考虑, Dóczi 等人[58]通过在应用层控制节点间数据流, 为基于 ROS 的医疗机器人系统提供了合适的安全性解决方案. 该方案是受 ALG (application level gateway) 启发的, 功能与 ALG 类似. 作者创建了一个独立的节点, 用于完成身份

验证和授权, 实现访问控制, 从而保证 ROS 中数据传输的机密性和完整性. 其中, 身份验证是通过验证节点或用户的登录名和密码实现的, 授权功能是通过检查请求者发送的数据与数据库中的安全记录是否匹配实现的. 但是该方案有一个显著的缺点, 那就是一旦恶意节点获取了某个受信任节点的名称和密钥, 它就可以利用这些关键信息伪装成受信任的节点, 并且连接到 ROS 主服务器, 进而获取机密数据甚至直接发送命令操纵机器人.

Huang 等人[59]为了解决 ROS 中存在的关键安全问题, 提出了一个名为 ROSRV 的基于 ROS 的机器人应用程序运行时验证框架. 该框架具有轻量级、表达性和透明性等特性. ROSRV 使用正式的规范语言定义了系统中的安全属性, 并根据规范自动生成监视器. ROSRV 的核心就是这个监视器, 它具有拦截、观察和修改系统中传递的命令和消息的功能, 并且可以执行消息中定义的事件. 监视器是作为一个普通节点实现的, 该节点负责监视节点间的通信, 当检测到对机器人系统具有潜在威胁的消息时, 它会立即干预并拒绝该消息的传输和执行. 另外, ROSRV 使用一个基于 IP 地址的访问控制规范来定义访问控制策略, 并且将安全策略作为系统配置强制执行, 提高了数据传输的机密性和完整性. ROSRV 基于用户提供的访问策略规范作为输入配置来实施访问控制. 当节点向 ROS Master 发送 XMLRPC 请求时, RVMaster 节点会截获该请求, 并根据规范决定是否允许该请求转到 ROS Master. 这样可以实现持续观察所有通信请求和订阅/发布消息, 从而实现了消息监控的安全策略. 作者使用无人地面车辆 LandShark3 演示了 ROSRV 利用监控安全属性和执行访问控制策略来提高 LandShark3 安全性的整个过程. 该方案具有以下优点: (1) ROSRV 可以与 ROS 无缝集成, 而无需修改 ROS 框架. (2) ROSRV 是开源的, 所有源代码、资料和演示都可以在指定网址获得. (3) ROSRV 已经在商用 LandShark3 机器人上实践和应用. 该方案也有一些缺点: (1) ROSRV 过度信任 IP 地址, 特别是依赖网络路由来保障机器人系统的安全性. 当攻击者使用与可信节点相同的机器对机器人发起攻击时, 系统不能提供安全防御. (2) ROSRV 采用集中式的监控方式, 所有监控节点都运行于同一个多线程进程, 当 ROS 系统中节点过多时, 这可能会给机器人系统带来不可接受的通信延迟.

3.2.2 传输层安全加固

为了克服 Dieber 等人[57]提出的纯应用程序级方法固有的局限性, Breiling 等人[60]提出了一种 ROS 的安全通信信道. 该方法适应于所有节点之间的通信, 包括发布节点与订阅节点以及服务/操作与它们的客户端之间的通信. 其中 ROS 中的 TCP 通信通道使用 TLS (transport layer security) 来为通信过程提供安全保障, 而 UDP 通信通道则是使用 DTLS (datagram transport layer security) 来确保通信的安全性. (D)TLS 要求两个节点之间通信之前, 必须执行额外的握手以完成基于 X.509 证书和公钥加密的相互认证和授权, 然后使用对称加密算法 AES-256 完成对通信数据的安全加密以实现消息传输的机密性, 使用消息认证码 MACs 来确保通信数据的完整性. 不仅如此, 该方法还在每个话题的基础上执行了细粒度授权, 实现了对话题的访问控制. 为了实现安全通信通道, 作者找到了 roscpp 包中 TCPROS 和 UDPROS 的实现, 并修改了 roscpp 包. 也就是说, 该方法修改了 ROS 核心. 安全实现不可避免会向机器人系统中引入额外的开销, 作者从握手开销、传输开销、应用开销这 3 个方面对开销进行了评估. 该方法可以有效应对 ROS 中的诸多安全漏洞. 它可以有效阻止未经授权的数据访问, 攻击者在没有证书的情况下无法订阅应用程序中的话题, 也就无法执行数据监听. 这样就减少泄露机密信息的风险. 它还可以防止未经授权的节点发布任意话题的数据, 并趁机向应用程序中注入虚假的数据或命令, 以干扰或者控制机器人. 同时, 它还可以减少针对特定 ROS 节点的 DoS (denial of service) 攻击. 该方法的优势在于它通过在直接通信的两个节点之间建立信任关系, 达到了保护节点间的 TCP 和 UDP 通信通道的目的, 实现了针对特定话题和特定节点的身份验证、授权和访问控制, 从而保证了传输数据的机密性和完整性. 但是, 该方法不能保护 ROS 的 XMLRPC API, 攻击者仍然可以检索节点图, 发送 publisherUpdates 并关闭节点. 此外, 在 ROS 中, 该 API 的 master 端是用 Python 实现的, 因此仅修改 roscpp 的方法不能覆盖这一部分.

在 ROS 的诸多安全性解决方案中, 最著名的是 SROS (securing ROS)[61], 它是开源机器人基金会 (OSRF) 为了在 ROS 中支持现代加密和安全措施以解决 ROS 现有的漏洞而进行的 ROS 安全性增强项目. 该项目致力于提出一组新的安全特性的核心 ROS 代码库. 在 2016 年举行的 ROSCon 大会上, White 等人[62]首次公开了 SROS. SROS 的安全策略主要包括两个方面: (1) 通过使用公钥基础设施 (public key infrastructure, PKI) 为每个 ROS 节点提供一个 X.509 证书, 实现了对所有数据传输使用传输层安全 (TLS) 进行加密, 提高数据传输的机密性. 为了支持

和简化在 ROS 中使用 PKI 的任务, SROS 提供了一个密钥服务器用于生成证书, 还提供了一个低级 API 用于在设置过程中向节点分发加密的证书. (2) SROS 还为话题、服务、参数和 XMLRPC 调用提供了访问控制机制、身份验证和授权, 保证了数据传输的完整性. 考虑到普通用户和开发人员对 SROS 的关注层面可能有所不同, 作者对 SROS 做出了有针对性的介绍, 其中, 针对普通用户, 作者通过实例介绍了使用加密、访问控制和内核模块来保护本机 ROS 应用程序的方法. 面向开发人员, 作者介绍了 SROS 的底层实现和架构设计原理. 为了促进 SROS 的进一步发展, 作者深入剖析了 SROS 在设计和执行方面的不足, 并给出了可行的改进建议. (1) 针对 SROS 的策略语法不够简明和表达性不强的问题, 作者打算采用一种类似于 AppArmor 中使用的语法语言, 这种语言具有灵活性和直观性, 方便不同用户使用. (2) 为了进一步自动化一般访问控制策略的构建和验证, 作者希望标准化安全事件日志消息, 使每个节点都可以自我报告潜在的安全事件. (3) 为了实现传输安全和访问控制使用相同的 X.509 证书, 公共握手过程中暴露了潜在的敏感内容, 针对这个问题, 作者建议将用于建立身份验证和授权的文件分开. 除此之外, 相关文献还总结了 SROS 的其他方面的缺点, 包括 (1) 证书和密钥的生成和分发在前面的初始化阶段由 keyserver 节点处理, 但是, 无法保证初始化阶段运行环境的安全性. (2) SROS 仅为 Python 实现, 不考虑 C++实现的节点. (3) 仅保护 TCP 通信通道, 没有考虑 UDP 通信通道. SROS 仍然处于开发阶段, 以上这些都表明 SROS 仍然有很大的改进空间. 然而, 值得一提的是, Portugal 等人[55]定量分析了包括 SROS 在内的 5 个独立的安全性解决方案对系统性能的影响以及安全防护效果, 实验结果证明 SROS 是经过测试的最安全的方案.

为了解决基于 ROS 的应用程序面临的网络安全威胁, Mukhandi 等人[63]提出了一种将 ROS 与消息队列遥测传输 (MQTT) 协议相结合新方法, 来保护可移动机器人系统. MQTT 是机对机 (M2M) 通信和物联网 (IoT) 中的理想通信协议, 具有轻量、开放、简单、易于实现等特点. MQTT 具有利用 SSL/TLS 的内置安全特性, 实现这些特性可以为所有数据提供安全性和隐私性. 它将 ROS 与轻量级 MQTT 协议相集成, 促使 MQTT 成为客户机与机器人之间连接的桥梁, 这样可以隔离远程用户对 ROS 的直接访问, 并提供机器人与远程客户端之间的安全通信, 从而保护支持 ROS 的机器人系统之间的数据共享. 它还采用 MQTT 内置的安全功能来保护支持 ROS 的机器人系统和远程客户机之间的通信, 通过提供身份验证和数据加密保证数据传输的机密性和完整性. 具体来说, 一方面, 它使用 MQTT 成熟的身份验证机制, 包括公钥基础设施 (PKI)、X.509 数字证书和传输层安全, 为机器人网络通信提供身份验证和数据隐私. 另一方面, 它利用了 MQTT 协议提供的授权机制, 用于指定客户机对特定资源的访问权限. 通过使用访问控制列表 (ACL), 可以在连接到 MQTT 服务器时指定客户机的访问权限. 这里的 ACL 指的就是权限列表, 根据表中设置的权限范围, 用户只能访问指定的话题, 并对这些话题执行指定操作. 如果远程客户机想与机器人建立连接, 它首先要连接到 MQTT 服务器, 并使用有效的数字证书进行身份验证. 身份验证之后, 客户机从 MQTT 服务器获得授权, 进而与机器人建立安全通信通道. 作者通过实验从额外的延迟、消息吞吐量和消息速率这 3 个方面评估了该方法对基于 ROS 的机器人监视系统性能的影响. 实验数据表明, 加密解决方案在客户端和服务器通信期间增加了可以忽略的网络延迟, 当使用数字证书集成身份验证和有效载荷加密时, 消息吞吐量和消息速率会降低. 但是作者认为, 相比于实施安全机制提供的明显优势, 如数据隐私、认证和防止网络攻击, 安全机制对性能的影响就显得微不足道了. 该方法解决了 ROS 框架的一些安全问题, 通过提供身份验证和数据加密来确保机器人的网络通信安全, 该方法可以防止中间人攻击和劫持攻击. 该方法可以用于几乎所有支持 ROS 的网络应用程序. 虽然使用该方法可以与远程客户端建立安全的网络连接, 但是作者指出 ROS 内部的通信仍然是不安全的, 因为没有任何数据加密.

3.2.3 网络层安全加固

Rivera 等人[64]专门为 ROS 设计了一个名为 ROS-Defender 的网络层监控和安全工具, 为增强 ROS 安全性提供了一个整体解决方案, 它基于安全事件管理系统、入侵防御系统和机器人防火墙, 并且结合了应用 (ROS) 级和网络级的异常检测系统, 可以使用软件定义网络 (SDN) 在网络层保护 ROS 免受大规模攻击, 通过在顶部集成防火墙和监控监视两个安全应用程序, 从而实现对 ROS 系统的访问控制, 确保 ROS 中传输数据的机密性和完整性. 它可以监控 ROS, 同时也可以通过检测违反现有策略的行为或新的攻击并调整策略, 以此来为 ROS 提供安全特性. 但是, 使用 ROS-Defender 将带来很大的性能成本, 因为它会给机器人系统带来额外的开销, 此外, ROS-Defender

只支持 ROS. 后来, 作者为了对 ROS-Defender 进行优化, 又开展了进一步的研究工作, Rivera 等人[65]利用 eBPF (extended Berkely packet filters) 防火墙组件和 XDP (express data path) 技术构建了一个高性能内联网络监控框架 ROS-FM, 其目的是为 ROS 提供模块化、可扩展和安全的监控软件. 通过将 ROS-FM 建立在 eBPF 防火墙组件上, 作者为依赖于 4.5 层路由的系统 (如 TCPROS 和 DDS) 提供了第 1 个 eBPF 防火墙, 通过访问控制提高 ROS 中数据传输的机密性和完整性. ROS-FM 将 ROS-Defender 扩展到了支持 ROS2, 而且它极大地改进了现有 ROS 监控工具的性能, 同时提高了 ROS 系统的整体安全性. ROS-FM 安全监控工具具有明显的优势: (1) 在性能方面, 与通用 ROS 监控工具相比, 它极大地改进了现有 ROS 监控工具的性能. (2) 在安全性方面, 与两个现有 ROS 渗透测试工具 (ROSPenTo、 Amazon's ROS2 security test node) 进行比较, 它对 ROS 和 ROS2 中常见攻击是有效的, 它可以提高 ROS 系统的整体安全性. (3) 在再现性方面, ROS-FM 的代码和数据是公开的.

ROS 与网络通信的集成可以实现数据共享, 但同时也把 ROS 暴露于网络攻击. 其中虚假数据注入攻击会对 ROS 造成严重损害. 虽然利用消息身份验证码 (MAC) 可以减轻完整性攻击, 但是 MAC 所需的大量计算会给系统带来巨大延迟. 考虑到许多机器人系统是延迟敏感的, Xu 等人[66]提出了一种轻量级密码的跨层设计方法, 该方法在保证 ROS 安全性的同时减少安全机制带来的延迟. 具体来说, 作者设计了一个轻量级的密钥长度较短的 MAC (LMAC), 这种轻量级 MAC 适用于基于 ROS 的延迟敏感的机器人系统, 在保证 ROS 中数据传输的机密性和完整性的同时可以降低计算复杂度, 减少延迟. 但是, LMAC 会增加攻击者破坏安全机制的风险, 因为 LMAC 很容易破坏. 因此, 作者需要考虑 ROS 的安全性和物理性能之间的权衡. 在网络层, 作者定义了两种网络状态来描述 LMAC 的状态. 一种状态是工作状态, 此时 LMAC 可以保护 ROS 免受完整性攻击; 另一种是失败状态, 此时 LMAC 受到攻击者的威胁, 无法保护系统. 考虑到如果防御者不知道攻击者是否破坏了 MAC, 就无法直接观察网络状态, 作者为防御者制定了一个网络部分观察马尔科夫决策过程 (POMDP) 来捕捉网络状态的不确定性. 基于分析结果, 作者提出了 POMDP 的算法解决方案. 作者使用具体的实例数据来评估系统的控制性能、安全性和弹性之间的权衡. 在物理层, 作者为系统设计了一个时延控制器, 来减少 LMAC 带来的延迟. 作者通过这种跨层控制系统实现了对延迟敏感的基于 ROS 的机器人系统的安全、弹性控制, 并保持其实时控制性能.

随着机器人智能化程度越来越高, 复杂任务分布式计算和远程控制成为机器人的高效运作的一种重要方式, 也催生了基于云的解决方案和远程客户端. ROS 社区开发了一个名为 Rosbridge[67]的协议规范用于建立本机 ROS 系统和远程客户机 (本机或非本机) 之间的通信. 然而, 对 ROS 系统远程访问带来的安全问题日益引起研究人员的关注, 因为缺乏适当的安全程序保障远程通信的安全性. 针对这个问题, Toris 等人[68]开发了一个系统独立的用于远程、非本地 ROS 客户机的基于 MAC 的自定义身份验证模式, 即 rosauth 的模式, 它是一种利用 Web 身份验证令牌通过任意外部用户管理系统来验证远程客户端的模式, 可以保证数据传输的机密性和完整性. 它通过使用非本机 ROS 客户端对来自任何 IP 地址的远程用户进行身份验证, 确保只有从可信的外部身份验证源进行身份验证的客户端才允许访问机器人. 这样可以确保 ROS 设备之间远程通信的安全性, 防止恶意访问. 与 VPN 相比, rosauth 具有较好的通用性. VPN 是建立 ROS 系统与远程客户端 (本地和非本地) 之间安全连接的常用方法. 但是因为它的配置十分复杂, 给非专业用户的使用带来困难. 另外, 想要使用 VPN, 必须额外安装单独的用户端软件. 而 rosauth 不依赖于任何特定的用户管理系统, 也不要求使用者使用特定的身份验证系统. 这种通用模式允许使用各种开箱即用 (例如 RMS) 系统, 还支持自定义用户管理系统来控制对机器人的访问. rosauth 模式已经集成到了 Rosbridge 协议中, 并且具有良好的可扩展性. 但是 rosauth 也有一些不足, 一方面, 由于 rosauth 模式不具备授权功能, 任何通过身份验证的远程客户机都可以访问整个 ROS 系统, 不过 rosauth 具有扩展并实现授权功能的能力, 它可以使用安全令牌中级别字段与当前客户端的用户权限级别进行关联. 另一方面, rosauth 只关注远程、非本机 ROS 客户机与 ROS 系统进行远程通信的场景, 不过 rosauth 包的抽象性使其具有实现本地 ROS 客户机与 ROS 系统之间的安全通信的能力.

3.2.4 传输层+应用层安全加固

Dieber 等人[69]将 Dieber 等人[57]和 Breiling 等人[60]提出的安全性增强方法进行了结合, 进一步减少 ROS 系统面临的安全风险. 然后, 作者使用 Dieber 等人[50]提出的 ROS 专用渗透测试工具 RosPenTo 对 ROS 应用程序进行

渗透测试. 最后, 作者将“不安全”的 ROS、应用层安全方法、安全通信通道 3 种场景下 ROS 的安全性进行了对比, 记录并比较了文中提到的多种攻击行为对每个场景下的 ROS 系统所产生的影响: 在“不安全”的 ROS 中, 节点不需要进行身份验证和授权就可以任意发布和订阅任何话题, 也可以随意使用 ROS 中的服务. 使用应用层安全方法增强 ROS 的安全性以后, 未授权节点仍然可以订阅话题, 但是由于未获得正确的解密密钥, 无法解释话题消息内容. 未授权节点也可以发布话题, 但发送的消息将由于缺少签名而被忽略. 未授权节点无法使用 ROS 中的服务. 在具有安全通信通道的 ROS 中, 未授权的节点无法订阅和发布, 也不能使用 ROS 中的服务, 因为在通信开始时, 未授权的节点由于缺少有效的证书, (D)TLS 握手失败, 通信会被取消. 此外, 在攻击者可以提供有效证书的情况下, 两种安全方法都将使用细粒度的话题访问控制机制来确保只有以前已授予的话题可以订阅或发布.

SRI 国际公司开发的 Secure ROS[70], 用于提供核心 ROS 包的替代版本, 使 ROS 节点之间能够进行安全通信. Secure ROS 的主要目标是使 ROS 的常规用户能够进行安全通信. 为此, 作者在 ROS 中集成了 IP 安全扩展 (IPSec). IPSec 用于传输模式, 对交换消息的有效载荷进行加密和验证, 保证数据传输的机密性和完整性. 此外, 传输层和应用层总是由散列保护, 因此不能以任何方式修改它们. 用户可以在运行时为 ROS 主服务器指定授权的话题订阅者和发布者、参数的设置者和获取者、服务的提供者 (服务器) 和请求者 (客户端), 从而实现访问控制. 因此, Secure ROS 将只允许授权节点连接到指定配置中列出的特定话题、服务和参数. 在实现方面, 与著名的 SROS 相比, Secure ROS 的安装过程更加迅速, 因为有可用的 debian 版本. 此外, Secure ROS 同时支持 rospy 和 roscpp, 它易于设置, 对于普通用户来说简单透明, 只需要提供所需的配置和对每个 ROS 实体的访问规则. Secure ROS 的一个缺点是, 它没有提供正式的验证手段来保证所需的属性符合规范.

以上内容是对近几年 ROS 安全性相关研究进行的概括总结, 值得一提的是, Portugal 等人[55]对自身以及上述文献[57,61,68,70]中提出的共 5 种 ROS 安全性解决方案在增强 ROS 安全性方面的效果进行了测试和分析. 作者将上述 5 种举措和官方发布的“不安全的” ROS Kinetic Kame 分别运行在同一台机器上, 然后测试了发布者和订阅者节点之间传输数据时, 每个举措以及 ROS 的通信性能, 包括: 通信中的延迟、丢失消息的数量、保持预期发布速率的能力、ROS 网络中来自未授权节点的访问级别. 最后将上述 5 种举措的测试结果与 ROS 的测试结果进行了对比和分析. 可以根据实验结果评估每种方法在安全性和可操作性之间的权衡. 测试中设计了两种类型的消息, 由两个单独的 ROS 节点以 3 档不同的速率 (适中的、快速的和几乎无法忍受的速率) 进行多次发布和订阅, 分别是 (1) 由 27 个字节组成的字符串, 以 3 种不同的预期发布速率 (1 kHz、10 kHz 和 30 kHz) 发布了 60 万次. (2) 343 KB 的数据组成的网格地图 (grid map), 以 3 种不同的预期发布速率 (250 Hz、2.5 kHz 和 7.5 kHz) 发布了 15 万次. 作者对实验结果进行了定量分析和定性分析.

(1) 定量分析: 通信中的延迟、丢包率、保持预期发布速率的能力

上述所有消息传输机制在将消息从非本地客户端传输到 ROS 时, 都会对发布/订阅通信性能产生影响.

当传输只有 27 个字节大小的字符串消息时, 在不考虑部分举措无法达到预期发布率的情况下, 所有举措的通信延迟和丢包率与 ROS 相比都非常接近, 这说明此时所有举措的性能都可以与官方发布的 ROS 版本的性能相媲美.

当传输 343 KB 的大消息时, 实验结果与字符串实验结果截然不同. 总的来说, 与其他安全举措相比, Secure ROS 具有较好的通信性能, 它不但能够达到预期的发布速率, 而且通信延迟比其他举措的通信延迟小, 并且具有较低的丢包率, 这证明了 Secure ROS 是在不影响传输性能的情况下增强 ROS 安全性的最有前途的举措之一. 其他举措的通信性能各不相同, 例如, ROS-AES-Encryption 算法虽然在实验中的丢包率为 0, 但是其最高只能达到约 186 Hz 的发布频率. 其传输消息的平均延迟是常规 ROS 传输的 15–45 倍. 具体数据请参考原文.

(2) 定性分析: ROS 网络中来自未授权节点的访问级别

ROS 支持在命令行工具中键入指定的命令来匿名检索指定的数据, 作者通过在 ROS 网络内运行了 rostopic list、rosnode list、rosservice list、rosnode kill <node>、rostopic echo <topic> 命令, 检查每个安全举措授予的访问级别, 即检查每个安全举措对未经授权的节点发出的请求所返回的数据.

实验数据显示, SROS 不响应来自未授权节点对 ROS 主节点的任何类型的查询, 它是具有更高安全级别的安

全举措. 其次是 Secure ROS, 它不允许未经授权的节点列出 ROS 话题, 也不允许查看话题中的消息, 同时也不允许杀死节点. 但是它允许列出 ROS 使用的节点和服务. 其他安全举措授予的访问级别可以查看原文.

通过对实验结果的定量分析和定性分析, 总的来说, SROS 和 Secure ROS 目前是 5 个安全举措中最有可能增强 ROS 安全性的举措. 其中 Secure ROS 开销小, 传输性能好, 而且它在一定程度上可以防止未经授权的节点访问 ROS 中的数据. SROS 无疑是增强 ROS 安全性最有力的举措, 而且它可以提供较好的高吞吐量传输性能.

3.2.5 ROS2

虽然不少开发者和研究机构针对 ROS 的局限性进行了适当改进, 但是这些局部功能的改善很难带来整体性能的提升, 这就为新一代 ROS 的开发创造了机遇. 于是, 在 ROSCon2014 上, 新一代 ROS 的设计架构 (next-generation ROS: building on DDS) 被正式公布.

● ROS2 简介

ROS2 (robot operating system 2)[71]中集成和应用了众多新技术和新概念, 这不仅带来了架构上的颠覆性设计, 而且增强了 ROS2 的综合性能. ROS2 自 2017 年发布第 1 个正式版本 Ardent Apalone 以来, 一直保持着持续的更新, 撰写本文时 ROS2 Galactic Geochelone 版本已经正式发布[72].

为了对 ROS 和 ROS2 进行更深入的介绍, 本文将对两者的架构进行分析和比较. 架构如图 5 所示.

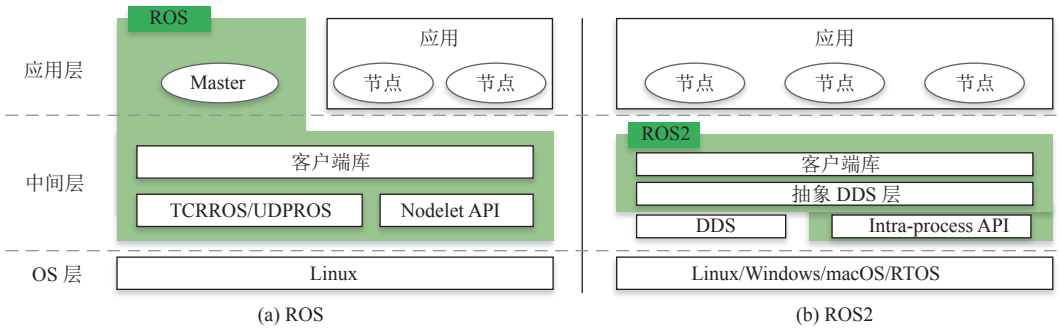

图 5 ROS 和 ROS2 的架构图[73]

(1) 操作系统层 (OS 层)

ROS 主要构建在 Linux 系统上, ROS2 支持构建的系统包括 Linux、Windows、macOS、RTOS, 甚至还支持没有操作系统的裸机.

(2) 中间层

ROS 的通信系统基于 TCPROS/UDPROS, 而 ROS2 的通信系统基于 DDS (data distribution service). ROS2 内部提供了 DDS 的抽象层实现, 用户无需关注底层 DDS 的提供厂商.

在 ROS 架构中, Nodelet 和 TCPROS/UDPROS 是并列的层次, 可以为同一进程中的多个节点提供更优的数据传输方式. ROS2 中也保留了类似的数据传输方式, 命名为“Intra-process”, 同样独立于 DDS.

(3) 应用层

ROS 强依赖于 ROS Master, Master 宕机会使整个机器人系统陷入困境. ROS2 架构去除了 Master, 节点间使用名为“Discovery”的发现机制来帮助彼此建立连接.

● ROS2 安全性分析

为了让 ROS 能够符合工业级的运行标准, ROS2 在设计之初就考虑了安全通信, ROS2 在其传输层中采用了 OMG 指定的被称为 DDS[74]的行业标准[75], 通过在系统中集成 DDS-security[76]规范进行动态数据保护, 以此提高通信的可靠性[77].

DDS-Security 规范是 DDS 规范的扩展, 它通过定义服务插件接口 (service plugin interface, SPI) 体系结构、一组 SPI 的内置实现以及 SPI 强制执行的安全模型来定义安全机制. DDS-Security 规范共定义了 5 个 SPI, ROS2 的

安全特性目前只使用了其中 3 种 SPI, 它们分别是身份验证、访问控制和加密. (1) ROS2 使用 SPI 体系结构内置的身份验证插件 DDS: Auth: PKI-DH. 该插件使用了经过验证的公钥基础设施 (PKI), 它需要每个参与者的公钥和私钥, 并且将参与者的公钥绑定到特定名称的 X.509 证书. 每个 X.509 证书必须由插件配置为信任的特定证书颁发机构 (CA) 签名 (或拥有到 CA 的签名链). 身份验证插件实现了对每个参与者的身份验证. 只有通过身份验证的参与者才可以进行下一步动作, 例如, 如果一个节点想要订阅某个话题, 它首先要通过身份验证, 然后才能订阅话题. (2) ROS2 使用 SPI 体系结构内置的访问控制插件 DDS: Access: Permission, 该插件负责对通过认证的参与者可以执行的 DDS 相关操作进行限制. 它同样使用了 PKI. (3) ROS2 使用 SPI 体系结构内置的加密插件 DDS: Crypto: AES-GCM-GMAC, 该插件提供了使用高级加密标准 (AES) 的认证加密. 加密插件负责处理所有与加密相关的操作, 包括加密、解密、签名、散列等. 身份验证和访问控制插件都利用加密插件的功能来验证签名等, 这也是加密 DDS 话题通信的功能所在.

值得注意的是, 在缺省情况下, ROS2 没有开启 DDS 的安全特性. ROS2 通过使用 Secure ROS2 工具启用 DDS 支持的安全特性. ROS2 中已经集成了两个厂商的 DDS 实现, 分别是 eProsima 的 FastRTPS 和 Adlink 的 OpenSplice, 用户可以根据需求进行选择, 也可以通过源码编译的方式使用其他厂商的 DDS 实现.

ROS2 还提供了多种服务质量 (QoS) 策略, 允许对节点之间的通信进行调整[78]. 通过正确设置服务质量策略, ROS2 可以像 TCP 一样可靠, 或者像 UDP 一样“尽力而为”, 且可以有很多种介于 TCP 和 UDP 两者之间可能状态.

3.2.6 ROS 安全性解决方案的对比分析

在安全性方面, 机密性、完整性和可用性 (confidentiality、integrity、availability, CIA) 是信息安全的核心, 这 3 个安全性概念也同样适用于机器人系统[19,63,69], 任何危及上述 3 个安全需求的行为都被视为探索系统漏洞的攻击[79]. 其中, 机密性是指传输的数据必须仅对授权的主体可用. 完整性是指传输的数据不能被未经授权的主体修改. 可用性是指组件必须运行并提供相应的服务. 本节将机密性、完整性和可用性作为 3 个关键指标, 对上述 ROS 安全性解决方案进行整体的对比和分析, 如表 1 所示.

表 1 ROS 安全性解决方案对比

OSI分层	ROS安全性解决方案	解决的ROS中的安全问题				在保证CIA上所做的努力		
		传输数据未加密	未进行身份验证	未进行访问控制	不提供服务质量	机密性	完整性	可用性
应用层	ROS-3DES-ENCRYPTION[19]	√	×	×	×	√	×	×
	ROS-AES-ENCRYPTION[55]	√	×	×	×	√	×	×
	ROS-3DES/AES/Blowfish-ENCRYPTION[56]	√	×	×	×	√	×	×
	SECURE-ROS-TRANSPORT[57]	√	√	×	×	√	√	×
	SECURE-ROS 1.x-communication[58]	×	√	√	×	√	√	×
	ROSRV[59]	×	×	√	×	√	√	×
传输层	SECURE-ROS-communication[60]	√	√	√	×	√	√	×
	SROS[61]	√	√	√	×	√	√	×
	SECURE ROS With MQTT[63]	√	√	√	×	√	√	×
	ROS-Defender[64]	×	×	√	×	√	√	×
	ROS-FM[65]	×	×	√	×	√	√	×
	Cross-layer secure ROS[66]	×	√	×	×	√	√	×
	SECURE-ROS-remote non-native client connections[68]	×	√	×	×	√	√	×
传输层+应用层	Security for ROS[69]	√	√	√	×	√	√	×
	Secure ROS[70]	√	√	√	×	√	√	×
—	ROS2[71]	√	√	√	√	√	√	√

在表 1 中, 每个 ROS 安全性解决方案所在行都有“√”和“×”两个符号, 其中, 在“解决的 ROS 中的安全问题”所在列中, “√”表示该方案针对 ROS 的某个安全性问题, 给出了相应的解决方案, “×”则表示没有给出相应的解决方案. 而“在保证 CIA 上所做的努力”所在列中, “√”和“×”表示该方案是否能提高机密性、完整性、可用性.

下面以 SROS 为例, 进一步解释表 1 的内容. ROS 系统主要有 4 个方面的安全性问题, 针对 ROS 中存在的传输数据未加密、未进行身份验证和未进行访问控制 3 个方面的安全问题, SROS 分别给出了相应的解决方案, 但是对于 ROS 不提供服务质量这一安全问题, SROS 没有给出相应的解决方案. SROS 所提供的安全性解决方案在一定程度上保证了 ROS 中数据传输的机密性、完整性, 但是未明确说明该方案在提高可用性方面发挥的作用.

根据现实应用场景中对机器人系统安全性的实际需求, 研究人员利用现有的多种信息安全技术, 针对 ROS 系统中暴露的主要安全问题, 有针对性地提出了 ROS 安全性解决方案. 从表 1 可以看出, 每种解决方案都在一定程度上减轻了 ROS 给机器人系统带来的安全威胁. 研究人员在提高机器人系统的机密性、完整性方面或多或少都开展了具体的工作. 不过, 由于对安全性的需求不同, 安全性增强的侧重点不同. 例如 ROSRV 优先考虑了数据传输的机密性, 而没有验证数据的完整性, 也没有在提高系统可用性方面做出努力. 由于 ROS 系统架构本身缺乏安全性的设计, 当前的安全性解决方案无法从根本上全面增强 ROS 的安全性, 每种方案都有各自的缺陷. 与基于 ROS 的诸多安全性解决方案不同, ROS2 全新的架构体系所具有的健壮的安全特性, 很好地避免了 ROS 中的诸多安全漏洞, 在机密性、完整性、可用性方面为机器人系统提供了强有力的安全保障. 不仅可以保证 ROS 中数据传输的机密性和完整性, ROS2 抵御外界恶意攻击的能力还可以为机器人系统正常提供服务提供更多的保障, 因此其可用性更强.

4 总结与展望

本文对 ROS 的安全性状况进行了分析和阐述, 并对近几年 ROS 安全性相关的研究进行了分类和概括, 特别地, 针对众多 ROS 安全性解决方案, 本文不仅总结了每种方案的主要研究内容, 还分析了方案的优缺点, 并且从方案所解决的 ROS 安全问题以及是否提高系统机密性、完整性和可用性两个方面, 对众多方案进行了分析和比较. 研究人员可以通过本文了解当前 ROS 安全性相关的研究工作, 从而更好地选择应用合适的 ROS 安全性解决方案, 也可以为未来实现更优的方案提供重要参考.

虽然研究人员在增强 ROS 安全性方面做出了大量努力, 但是不难看到, 并没有一个通用的、全面的、高效便捷的 ROS 安全性解决方案. 由于应用场景要求的安全等级不同或者针对的机器人安全问题不同, 每种解决方案都有各自的短板. 因此, 开发通用的 ROS 安全性解决方案来灵活适应不同的安全需求可能是未来的一个研究方向.

因为 ROS2 在系统架构设计时考虑了系统的安全性, 所以 ROS2 的安全性明显优于 ROS, 目前 ROS2 有取代 ROS 的趋势. 这个事实让人们不禁开始思考未来是否还有必要进行 ROS 安全性相关的研究. 考虑到 ROS 的使用群体十分庞大, 基于 ROS 的应用涉及的领域繁多, 将基于 ROS 的应用迁移到 ROS2 不仅需要花费额外的时间, 还需要花费额外的资金, 为了避免产生这些额外的成本, 开发者或投资商可能会继续选择使用 ROS. 此外, 经过十几年的发展, ROS 已经具有良好的社区生态以及丰富的功能包. 相比之下, ROS2 在 2017 年 12 月才发布第 1 个正式版本, 其社区生态尚不成熟, 许多功能包尚未从 ROS 移植到 ROS2. 因此, 开发者可能依然会选择使用 ROS 作为机器人的操作系统. 由此可以推断, 在未来一段时间内, ROS 仍会是主流的机器人操作系统. 所以从目前来看, 研究 ROS 安全性仍然是有需求的、有必要的. 但是从更长久来看, 等到 ROS2 的社区生态建立起来, ROS2 未来必将取代 ROS. 到那时, 研究 ROS2 的安全性也许更能满足实际需求.

References:

[1] Jones JL. Robots at the tipping point: The road to iRobot Roomba. IEEE Robotics & Automation Magazine, 2006, 13(1): 76–78. [doi: 10.1109/MRA.2006.1598056]

[2] Wang ZL, Tian GH, Shao XY. Home service robot task planning using semantic knowledge and probabilistic inference. Knowledge-based Systems, 2020, 204: 106174. [doi: 10.1016/j.knosys.2020.106174]

[3] Ji Z, Qiu RX, Noyvirt A, Soroka A, Packianather M, Setchi R, Li DY, Xu S. Towards automated task planning for service robots using semantic knowledge representation. In: Proc. of the 10th IEEE Int'l Conf. on Industrial Informatics. Beijing: IEEE, 2012. 1194−1201. [doi: 10.1109/INDIN.2012.6301131]

[4] Freschi C, Ferrari V, Melfi F, Ferrari M, Mosca F, Cuschieri A. Technical review of the da Vinci surgical telemanipulator. The Int'l Journal of Medical Robotics and Computer Assisted Surgery, 2013, 9(4): 396–406. [doi: 10.1002/rcs.1468]

[5] Kazanzides P, Chen ZH, Deguet A, Fischer GS, Taylor RH, DiMaio SP. An open-source research kit for the da Vinci® surgical system. In: Proc. of the 2014 IEEE Int'l Conf. on Robotics and Automation (ICRA). Hong Kong: IEEE, 2014. 6434–6439. [doi: 10.1109/ICRA.2014.6907809]

[6] He W, Ge SS, Li YN, Chew E, Ng YS. Neural network control of a rehabilitation robot by state and output feedback. Journal of Intelligent & Robotic Systems, 2015, 80(1): 15–31. [doi: 10.1007/s10846-014-0150-6]

[7] Luo RC, Hsu TY, Lin TY, Su KL. The development of intelligent home security robot. In: Proc. of the 2005 IEEE Int'l Conf. on Mechatronics, 2005. Piscataway: IEEE, 2005. 422–427. [doi: 10.1109/ICMECH.2005.1529294]

[8] Luo RC, Chou YT, Liao CT, Lai CC, Tsai AC. NCCU security warrior: An intelligent security robot system. In: Proc. of the 33rd IECON Annual Conf. of the IEEE Industrial Electronics Society. IEEE, 2007. 2960–2965. [doi: 10.1109/IECON.2007.4460380]

[9] Top 5 robot trends 2021. 2021. https://ifr.org/ifr-press-releases/news/top-5-robot-trends-2021

[10] Craig JJ. Introduction to Robotics. Pearson Educacion, 2006.

[11] Diftler MA, Mehling JS, Abdallah ME, Radford NA, Bridgwater LB, Sanders AM, Askew RS, Linn DM, Yamokoski JD, Permenter FA, Hargrave BK, Platt R, Savely RT, Ambrose RO. Robonaut 2—The first humanoid robot in space. In: Proc. of the 2011 IEEE Int'l Conf. on Robotics and Automation. Shanghai: IEEE, 2011. 2178–2183. [doi: 10.1109/ICRA.2011.5979830]

[12] Badger J, Gooding D, Ensley K, Hambuchen K, Thackston A. ROS in space: A case study on Robonaut 2. In: Koubaa A, ed. Robot Operating System (ROS). Berlin: Springer, 2016. 343–373. [doi: 10.1007/978-3-319-26054-9_13]

[13] Kirschgens LA, Ugarte IZ, Uriarte EG, Rosas AM, Vilches VM. Robot hazards: From safety to security. arXiv:1806.06681, 2018.

[14] Chun WH, Papanikolopoulos N. Robot surveillance and security. In: Siciliano B, Khatib O, eds. Springer Handbook of Robotics. Berlin: Springer, 2016. 1605–1626. [doi: 10.1007/978-3-319-32552-1_61]

[15] Liu T, Wang SL, Zhan NJ. Safety verification of trajectory planning for multiple robots. Ruan Jian Xue Bao/Journal of Software, 2017, 28(5): 1118−1127 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5207.htm [doi: 10.13328/j.cnki.jos.005207]

[16] Khalid A, Kirisci P, Ghrairi Z, Pannek J, Thoben KD. Towards implementing safety and security concepts for human-robot collaboration in the context of Industry 4.0. In: Proc. of the 39th Int'l MATADOR Conf. on Advanced Manufacturing. Manchester: Springer, 2017. 1–7.

[17] Duncan Swinscow-Hall. The interaction between safety and security. 2017. https://blogs.imperial.ac.uk/security-institute/2017/01/03/the-relationship-between-safety-and-security/

[18] Yang K, Wang R, Guan Y, Li XJ, Shi ZP, Song XY. Attack detection of CPS system with multi-sensors. Ruan Jian Xue Bao/Journal of Software, 2019, 30(7): 2018−2032 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5756.htm [doi: 10.13328/j.cnki.jos.005756]

[19] Lera FJR, Balsa J, Casado F, Fernández C, Rico FM, Matellán V. Cybersecurity in autonomous systems: Evaluating the performance of hardening ROS. Málaga, 2016. 47.

[20] ruffsl. SROS. 2016. http://wiki.ros.org/SROS

[21] Liu T, Tian J, Wang JZ, Wu HY, Sun LM, Zhou YD, Shen C, Guan XH. Integrated security threats and defense of cyber-physical systems. Acta Automatica Sinica, 2019, 45(1): 5–24 (in Chinese with English abstract). [doi: 10.16383/j.aas.2018.c180461]

[22] Asimov I. I, Robot. Bantam: Spectra, 2004.

[23] Guerrero-Higueras ÁM, DeCastro-García N, Rodríguez-Lera FJ, Matellán V. Empirical analysis of cyber-attacks to an indoor real time localization system for autonomous robots. Computers & Security, 2017, 70: 422–435. [doi: 10.1016/j.cose.2017.06.013]

[24] Quarta D, Pogliani M, Polino M, Maggi F, Zanchettin AM, Zanero S. An experimental security analysis of an industrial robot controller. In: Proc. of the 2017 IEEE Symp. on Security and Privacy (SP). San Jose: IEEE, 2017. 268–286. [doi: 10.1109/SP.2017.20]

[25] Portugal D, Pereira S, Couceiro MS. The role of security in human-robot shared environments: A case study in ROS-based surveillance robots. In: Proc. of the 26th IEEE Int'l Symp. on Robot and Human Interactive Communication (RO-MAN). Lisbon: IEEE, 2017. 981–986. [doi: 10.1109/ROMAN.2017.8172422]

[26] Quigley M, Conley K, Gerkey B, *et al*. ROS: An open-source robot operating system. ICRA Workshop on Open Source Software, 2009, 3(3.2): 5.

[27] Robot operating system. 2023. https://en.wikipedia.org/wiki/Robot_Operating_System

[28] Zhang L, Merrifield R, Deguet A, Yang GZ. Powering the world's robots—10 years of ROS. Science Robotics, 2017, 2(11): eaar1868. [doi: 10.1126/scirobotics.aar1868]

[29] ShaneLoretz. Quaternions. 2022. http://wiki.ros.org/tf2/Tutorials/Quaternions

[30] Rusu RB, Cousins S. 3D is here: Point cloud library (PCL). In: Proc. of the 2011 IEEE Int'l Conf. on Robotics and Automation. Shanghai: IEEE, 2011. 1–4. [doi: 10.1109/ICRA.2011.5980567]

[31] Pajaziti A. Slam-map building and navigation via ROS. Int'l Journal of Intelligent Systems and Applications in Engineering, 2014, 2(4): 71–75. [doi: 10.18201/ijisae.08103]

[32] Shi DX, Yang ZY, Jin SC, Zhang YJ, Su XD, Li RH. A multi-UAV collaborative SLAM method oriented to data sharing. Chinese Journal of Computers, 2021, 44(5): 983–998 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2021.00983]

[33] PackBot. 2023. https://en.wikipedia.org/wiki/PackBot

[34] TurtleBot2. 2023. https://www.turtlebot.com/turtlebot2/

[35] Nao (robot). 2023. https://en.wikipedia.org/wiki/Nao_(robot)

[36] Distributions. 2023. http://wiki.ros.org/Distributions

[37] Eugster PT, Felber PA, Kermarrec AM. The many faces of publish/subscribe. ACM Computing Surveys, 2003, 35(2): 114–131. [doi: 10.1145/857076.857078]

[38] Quigley M, Gerkey B, Smart WD. Programming Robots with ROS: A Practical Introduction to the Robot Operating System. O'Reilly Media Inc., 2015.

[39] Hellmund AM, Wirges S, Taş ÖŞ, Bandera C, Salscheider NO. Robot operating system: A modular software framework for automated driving. In: Proc. of the 19th IEEE Int'l Conf. on Intelligent Transportation Systems (ITSC). Rio de Janeiro: IEEE, 2016. 1564–1570. [doi: 10.1109/ITSC.2016.7795766]

[40] Wang Y, Wang BQ, Guan Y, Li XJ, Wang R. Differential fuzz testing of robot operating system. Ruan Jian Xue Bao/Journal of Software, 2021, 32(6): 1867−1881 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6254.htm [doi: 10.13328/j.cnki.jos.006254]

[41] Martín F, Soriano E, Cañas JM. Quantitative analysis of security in distributed robotic frameworks. Robotics and Autonomous Systems, 2018, 100: 95–107. [doi: 10.1016/j.robot.2017.11.002]

[42] Maji AK, Bagchi S. *v*-CAPS: A confidentiality and anonymity preserving routing protocol for content-based publish-subscribe networks. In: Proc. of the 7th Int'l Conf. on Security and Privacy in Communication Systems. Berlin: Springer, 2011. 281–302. [doi: 10.1007/978-3-642-31909-9_16]

[43] Ion M, Russello G, Crispo B. Design and implementation of a confidentiality and access control solution for publish/subscribe systems. Computer Networks, 2012, 56(7): 2014–2037. [doi: 10.1016/j.comnet.2012.02.013]

[44] Goerke N, Timmermann D, Baumgart I. Who controls your robot? An evaluation of ROS security mechanisms. In: Proc. of the 7th Int'l Conf. on Automation, Robotics and Applications (ICARA). Prague: IEEE, 2021. 60–66. [doi: 10.1109/ICARA51699.2021.9376468]

[45] Mayoral-Vilches V, Pinzger M, Rass S, Dieber B, Gil-Uriarte E. Can ROS be used securely in industry? Red teaming ROS-Industrial. arXiv:2009.08211, 2020.

[46] Jeong SY, Choi IJ, Kim YJ, Shin Y, Han J, Jung G, Kim K. A study on ROS vulnerabilities and countermeasure. In: Proc. of the 2017 Companion of ACM/IEEE Int'l Conf. on Human-robot Interaction. Vienna: ACM, 2017. 147–148. [doi: 10.1145/3029798.3038437]

[47] Teixeira RR, Maurell IP, Drews PLJ. Security on ROS: Analyzing and exploiting vulnerabilities of ROS-based systems. In: Proc. of the 2020 Latin American Robotics Symp. (LARS), the 2020 Brazilian Symp. on Robotics (SBR) and the 2020 Workshop on Robotics in Education (WRE). Natal: IEEE, 2020. 1–6. [doi: 10.1109/LARS/SBR/WRE51543.2020.9307107]

[48] Xiao XP, Ni LM. Internet QoS: A big picture. IEEE Network, 1999, 13(2): 8–18. [doi: 10.1109/65.768484]

[49] Menasce DA. QoS issues in Web services. IEEE Internet Computing, 2002, 6(6): 72–75. [doi: 10.1109/MIC.2002.1067740]

[50] Dieber B, White R, Taurer S, Breiling B, Caiazza G, Christensen H, Cortesi A. Penetration testing ROS. In: Koubaa A, ed. Robot Operating System (ROS). Berlin: Springer, 2020. 183–225. [doi: 10.1007/978-3-030-20190-6_8]

[51] Rivera S, Lagraa S, State R. ROSploit: Cybersecurity tool for ROS. In: Proc. of the 3rd IEEE Int'l Conf. on Robotic Computing (IRC). Naples: IEEE, 2019. 415–416. [doi: 10.1109/IRC.2019.00077]

[52] McClean J, Stull C, Farrar C, Mascareñas D. A preliminary cyber-physical security assessment of the robot operating system (ROS). In: Proc. of SPIE 8741, Unmanned Systems Technology XV. Baltimore: SPIE, 2013. 874110. [doi: 10.1117/12.2016189]

[53] Vilches VM, Kirschgens LA, Calvo AB, Cordero AH, Pisón RI, Vilches DM, Rosas AM, Mendia GO, Juan LUS, Ugarte IZ, Gil-Uriarte E, Tews E, Peter A. Introducing the robot security framework (RSF), a standardized methodology to perform security assessments in robotics. arXiv:1806.04042, 2018.

[54] DeMarinis N, Tellex S, Kemerlis VP, Konidaris G, Fonseca R. Scanning the internet for ROS: A view of security in robotics research. In:

Proc. of the 2019 Int'l Conf. on Robotics and Automation (ICRA). Montreal: IEEE, 2019. 8514–8521. [doi: 10.1109/ICRA.2019.8794451]

[55] Portugal D, Santos MA, Pereira S, Couceiro MS. On the security of robotic applications using ROS. In: Yampolskiy RV, ed. Artificial Intelligence Safety and Security. New York: Chapman and Hall/CRC, 2018. 273. [doi: 10.1201/9781351251389]

[56] Rodríguez-Lera FJ, Matellán-Olivera V, Balsa-Comerón J, Guerrero-Higueras ÁM, Fernández-Llamas C. Message encryption in robot operating system: Collateral effects of hardening mobile robots. Frontiers in ICT, 2018, 5: 2. [doi: 10.3389/fict.2018.00002]

[57] Dieber B, Kacianka S, Rass S, Schartner P. Application-level security for ROS-based applications. In: Proc. of the 2016 IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS). Daejeon: IEEE, 2016. 4477–4482. [doi: 10.1109/IROS.2016.7759659]

[58] Dóczi R, Kis F, Sütő B, Póser V, Kronreif G, Jósvai E, Kozlovszky M. Increasing ROS 1.x communication security for medical surgery robot. In: Proc. of the 2016 IEEE Int'l Conf. on Systems, Man, and Cybernetics (SMC). Budapest: IEEE, 2016. 4444–4449. [doi: 10.1109/SMC.2016.7844931]

[59] Huang J, Erdogan C, Zhang Y, Moore B, Luo QZ, Sundaresan A, Rosu G. ROSRV: Runtime verification for robots. In: Proc. of the 5th Int'l Conf. on Runtime Verification. Berlin: Springer, Cham, 2014. 247–254. [doi: 10.1007/978-3-319-11164-3_20]

[60] Breiling B, Dieber B, Schartner P. Secure communication for the robot operating system. In: Proc. of the 2017 Annual IEEE Int'l Systems Conf. (SysCon). Montreal: IEEE, 2017. 1–6. [doi: 10.1109/SYSCON.2017.7934755]

[61] White R, Caiazza G, Christensen H, Cortesi A. SROS1: Using and developing secure ROS1 systems. In: Koubaa A, ed. Robot Operating System (ROS). Berlin: Springer, 2019. 373–405. [doi: 10.1007/978-3-319-91590-6_11]

[62] White R, Christensen DI, Quigley M. SROS: Securing ROS over the wire, in the graph, and through the kernel. arXiv:1611.07060, 2016.

[63] Mukhandi M, Portugal D, Pereira S, Couceiro MS. A novel solution for securing robot communications based on the MQTT protocol and ROS. In: Proc. of the 2019 IEEE/SICE Int'l Symp. on System Integration (SII). Paris: IEEE, 2019. 608–613. [doi: 10.1109/SII.2019.8700390]

[64] Rivera S, Lagraa S, Nita-Rotaru C, Becker S, State R. ROS-Defender: SDN-based security policy enforcement for robotic applications. In: Proc. of the 2019 IEEE Security and Privacy Workshops (SPW). San Francisco: IEEE, 2019. 114–119. [doi: 10.1109/SPW.2019.00030]

[65] Rivera S, Iannillo AK, Lagraa S, Joly C, State R. ROS-FM: Fast monitoring for the robotic operating system (ROS). In: Proc. of the 25th Int'l Conf. on Engineering of Complex Computer Systems (ICECCS). Singapore: IEEE, 2020. 187–196. [doi: 10.1109/ICECCS51672.2020.00029]

[66] Xu ZH, Zhu QY. Cross-layer secure and resilient control of delay-sensitive networked robot operating systems. In: Proc. of the 2018 IEEE Conf. on Control Technology and Applications (CCTA). Copenhagen: IEEE, 2018. 1712–1717. [doi: 10.1109/CCTA.2018.8511500]

[67] Crick C, Jay G, Osentoski S, Pitzer B, Jenkins OC. Rosbridge: ROS for non-ROS users. In: Christensen HI, Khatib O, eds. Robotics Research. Berlin: Springer, 2017. 493–504. [doi: 10.1007/978-3-319-29363-9_28]

[68] Toris R, Shue C, Chernova S. Message authentication codes for secure remote non-native client connections to ROS enabled robots. In: Proc. of the 2014 IEEE Int'l Conf. on Technologies for Practical Robot Applications (TePRA). Woburn: IEEE, 2014. 1–6. [doi: 10.1109/TePRA.2014.6869141]

[69] Dieber B, Breiling B, Taurer S, Kacianka S, Rass S, Schartner P. Security for the robot operating system. Robotics and Autonomous Systems, 2017, 98: 192–203. [doi: 10.1016/j.robot.2017.09.017]

[70] Secure ROS 0.9.2 documentation. 2016. https://sri-csl.github.io/secure_ros/

[71] Macenski S, Foote T, Gerkey B, *et al*. Robot operating system 2: Design, architecture, and uses in the wild. Science Robotics, 2022, 7(66): eabm6074.

[72] Distributions. 2023. http://docs.ros.org/en/rolling/Releases.html

[73] Maruyama Y, Kato S, Azumi T. Exploring the performance of ROS2. In: Proc. of the 13th Int'l Conf. on Embedded Software. Pittsburgh: IEEE, 2016. 1–10. [doi: 10.1145/2968478.2968502]

[74] Data Distribution Service. 2023. https://en.wikipedia.org/wiki/Data_Distribution_Service

[75] ROS on DDS. 2019. https://design.ros2.org/articles/ros_on_dds.html

[76] DDS-security. 2018. https://www.omg.org/spec/DDS-SECURITY/1.1/PDF

[77] ROS2 DDS-security integration. 2020. https://design.ros2.org/articles/ros2_dds_security.html

[78] ROS 2 Documentation: Foxy. 2023. http://docs.ros.org/en/foxy/Concepts/About-Quality-of-Service-Settings.html

[79] Kumar R, Pattnaik PK, Pandey P. Detecting and Mitigating Robotic Cyber Security Risks. Hershey: IGI Global, 2017.

附中文参考文献:

[15] 刘涛, 王淑灵, 詹乃军. 多机器人路径规划的安全性验证. 软件学报, 2017, 28(5): 1118–1127. http://www.jos.org.cn/1000-9825/5207.htm [doi: 10.13328/j.cnki.jos.005207]

[18] 杨康, 王瑞, 关永, 李晓娟, 施智平, Song XY. 具有多传感器的CPS系统的攻击检测. 软件学报, 2019, 30(7): 2018–2032. http://www.jos.org.cn/1000-9825/5756.htm [doi: 10.13328/j.cnki.jos.005756]

[21] 刘烃, 田决, 王稼舟, 吴宏宇, 孙利民, 周亚东, 沈超, 管晓宏. 信息物理融合系统综合安全威胁与防御研究. 自动化学报, 2019, 45(1): 5–24. [doi: 10.16383/j.aas.2018.c180461]

[32] 史殿习, 杨卓越, 金松昌, 张拥军, 苏向东, 李睿豪. 面向数据共享的多无人机协同SLAM方法. 计算机学报, 2021, 44(5): 983–998. [doi: 10.11897/SP.J.1016.2021.00983]

[40] 王颖, 王冰青, 关永, 李晓娟, 王瑞. 面向ROS的差分模糊测试方法. 软件学报, 2021, 32(6): 1867–1881. http://www.jos.org.cn/1000-9825/6254.htm [doi: 10.13328/j.cnki.jos.006254]

鲁敬敬(1991－), 女, 博士生, 主要研究领域为机器人系统安全, 点云补全.

唐卓(1981－), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为大数据并行处理体系结构, 分布式机器学习.

秦云川(1983－), 男, 博士, CCF 专业会员, 主要研究领域为自主无人系统, 智能驾驶, 点云三维重建, 硬件系统安全.

张拥军(1972－), 男, 博士, 研究员, 博士生导师, 主要研究领域为高性能计算, 人工智能, 分布式系统.

刘志中(1990－), 男, 博士, 主要研究领域为计算智能, 机器学习.

李肯立(1971－), 男, 博士, 教授, 博士生导师, CCF 会士, 主要研究领域为并行分布式处理, 超级计算与云计算, 面向大数据和人工智能的高效能计算.