Инженерный вестник Дона, №12 (2023)
ivdon.ru/ru/magazine/archive/n12y2023/8862

Уязвимости и методы защиты операционной системы ROS при реализации мультиагентной системы на базе робота Turtlebot3

И.А. Хорсик, Л.В. Бунина

МИРЭА — Российский технологический университет

Аннотация: Рассматривается проблема уязвимостей в операционной системе Robot Operating System (ROS) при реализации мультиагентной системы на базе робота Turtlebot3. ROS предоставляет мощные инструменты для коммуникации и обмена данными между различными компонентами системы. Однако, при обмене данными между роботами Turtlebot3 могут возникать уязвимости, которые могут быть использованы злоумышленниками для несанкционированного доступа или атак на систему. Одной из возможных уязвимостей является перехват и подмена данных между роботами. Злоумышленник может перехватить данные, изменить их и повторно отправить, что может привести к непредсказуемым последствиям. Другой возможной уязвимостью является несанкционированный доступ к командам и управлению роботами Turtlebot3, что может привести к потере контроля над системой. Для решения данных уязвимостей разработаны и представлены методы защиты от возможных угроз безопасности, возникающих в ходе эксплуатации указанных систем.
Ключевые слова: роботизированная операционная система (ROS), мультиагентная система, системные пакеты, шифрование, SSH, TLS, система аутентификации и авторизации, канал связи, ограничение доступа, анализ угроз, Turtlebot3.

Введение

Операционная система ROS и ее возможности значительно выросли в последние годы в связи с широким применением робототехники в различных областях, таких как логистика, медицина, производство и т.д. ROS обеспечивает удобный интерфейс для управления роботами и позволяет интегрировать различные датчики и устройства, что дает возможность создавать сложные мультиагентные системы.

Однако, при реализации мультиагентной системы Turtlebot3 с использованием ROS, существуют риски безопасности. Уязвимости ROS могут привести к потенциальному нарушению конфиденциальности, целостности, и доступности данных. Разрабатывая приложения ROS, необходимо учитывать опасности утечки персональных данных, манипулирования результатами и обмена сообщениями между узлами.

Уязвимости ROS могут быть использованы злоумышленниками для создания ботнетов, распространения вирусов и выполнения других видов кибератак.

Таким образом, при проектировании мультиагентной системы на основе ROS, необходимо уделять повышенное внимание безопасности и защите от угроз. Это может предусматривать использование аутентификации, шифрования, а также контроля доступа и аудита.

В статье [1] было проведено множество работ по анализу уязвимостей робототехнических комплексов с роевым интеллектом. Одной из основных уязвимостей является возможность вмешательства в работу роевого интеллекта со стороны злоумышленников. В этом случае, злоумышленник может использовать уязвимости для внедрения своих команд и захвата контроля над роевым интеллектом.

Другой тип уязвимостей состоит в возможности атаки на интерфейсы устройств, которые управляют роевым интеллектом. Такая атака может привести к сбою работы системы или к выходу ее из строя.

Еще одна область, которая вызывает озабоченность – это уязвимости, связанные с передачей информации между устройствами роевого интеллекта. Это может быть связано с шифрованием и защитой данных от несанкционированного доступа.

Также важно не забывать о физической безопасности роботов, используемых в роевых комплексах. Роботы могут быть повреждены или выведены из строя, что может привести к сбою работы всего комплекса.

Итак, анализ уязвимостей робототехнических комплексов с роевым интеллектом подчеркивает необходимость обеспечения безопасности в различных аспектах работы системы. Надежность, защищенность и безопасность являются важнейшими задачами при создании таких систем.

Статья [2] посвящена проблемам безопасности роботизированных операционных систем (ROS).

Данная статья описывает различные уязвимости и атаки, на которые может быть подвержена ROS, и предлагает меры по защите системы. Рассмотрены такие уязвимости, как атаки через каналы коммуникации, атаки на службы авторизации и аутентификации, атаки на файловую систему и другие.

Для защиты ROS авторы статьи рекомендуют использование следующих мер:

- Использование шифрования данных при передаче по сети для защиты от атак [3] через каналы коммуникации;

- Использование сложных паролей и механизмов авторизации и аутентификации, чтобы защититься от атак на службы авторизации;

- Ограничение доступа к файловым системам и управляющим устройствам, чтобы предотвратить атаки на файловую систему;

- Использование механизмов логирования и мониторинга системы, чтобы обнаружить атаки на ранней стадии.

Также авторы статьи отмечают, что многие из этих мер могут быть реализованы с помощью сторонних инструментов и библиотек, таких как TLS или SSH.

В целом, статья представляет собой полезный обзор проблем безопасности ROS и предоставляет рекомендации по защите системы. Однако, как и любая другая операционная система, безопасность ROS требует постоянного мониторинга [4] и обновления мер безопасности для защиты от новых угроз.

Постановка задачи

Необходимо разработать механизм защиты ROS в мультиагентной системе на базе Turtlebot3 от возможных атак и утечек данных. Для этого необходимо выполнить следующие шаги:

1. Изучить основные уязвимости ROS и способы их эксплуатации.

2. Проанализировать архитектуру мультиагентной системы [5] на базе Turtlebot3 и выявить возможные уязвимости.

3. Определить наиболее эффективные методы защиты ROS и выбрать наиболее подходящие для данной системы.

4. Реализовать механизмы защиты в рамках мультиагентной системы на базе Turtlebot3.

5. Протестировать работоспособность механизмов защиты и оптимизировать их в случае необходимости.

Целью данного проекта является обеспечение безопасности мультиагентной системы на базе Turtlebot3 и защиты ее от возможных угроз. Решение этой проблемы позволит использовать систему в более широком диапазоне, в том числе в критических приложениях, где безопасность является первоочередным условием.

Решение задачи

Рассмотрим возможные уязвимости в системе ROS и методы их защиты. ROS использует открытый протокол обмена сообщениями, что может сделать систему уязвимой для атак. Для защиты ROS можно использовать следующие методы:

- Шифрование: зашифровать сообщения, передаваемые между узлами ROS, чтобы предотвратить возможность перехвата и чтения сообщений.

- Аутентификация: обеспечить проверку источника сообщений и узлов, взаимодействующих с системой ROS, чтобы предотвратить возможность подделки сообщений.

- Мониторинг: настроить систему мониторинга, чтобы отслеживать поток данных и обнаруживать аномальное поведение или потенциальные угрозы.

Теперь рассмотрим, какие шаги можно предпринять для защиты мультиагентной системы на базе Turtlebot3. Некоторые основные меры безопасности включают в себя:

- Ограничение доступа: настроить систему доступа к узлам ROS и мультиагентной системе на базе Turtlebot3, чтобы ограничить доступ только к авторизованным пользователям.

- Защита паролей: установить надежные пароли для доступа к системе и обновлять их регулярно, чтобы предотвратить возможность несанкционированного доступа к системе.

- Резервное копирование данных: настроить систему резервного копирования данных, чтобы обеспечить сохранность информации, в случае поломки оборудования или других проблем.

- Анализ угроз: регулярно проверять систему на наличие потенциальных угроз и анализировать логи, чтобы быстро выявлять и реагировать на атаки.

В целом, безопасность ROS и мультиагентной системы [6, 7] на базе Turtlebot3 может быть обеспечена путем сочетания различных методов, таких как шифрование, аутентификация, мониторинг, ограничение доступа и анализ угроз.

Проведение эксперимента

Для ограничения доступа к узлам ROS и мультиагентной системе на базе Turtlebot3 можно использовать систему аутентификации и авторизации.

Для настройки системы доступа между узлами ROS Noetic и Turtlebot3, мы можем использовать инструменты аутентификации и шифрования, такие как SSH и TLS [8,9].

В первую очередь мы установим и настроим SSH.

SSH (Secure Shell) – это протокол сетевой безопасности, который позволяет защищенно подключаться к удаленному узлу и передавать данные через зашифрованный канал [10, 11].

Для установки SSH на ROS и Turtlebot3 выполняем следующие команды:

sudo apt-get update
sudo apt-get install openssh-server

Затем настроем SSH, чтобы ограничить доступ только к авторизованным пользователям. Для этого редактируем файл конфигурации SSH:

sudo nano /etc/ssh/sshd_config

Находим и изменяем следующие параметры:

PermitRootLogin no
PasswordAuthentication no

Перезапускаем SSH при помощи команды:

sudo service ssh restart

Дальше мы устанавливаем и настраиваем TLS.

TLS (Transport Layer Security) – это протокол шифрования, который обеспечивает безопасную передачу данных между узлами.

Для установки TLS на ROS и Turtlebot3, выполняем следующие команды:

sudo apt-get update
sudo apt-get install libssl-dev

Затем создаем сертификаты TLS для каждого узла ROS Noetic и Turtlebot3. Для этого выполняем следующие команды на каждом узле:

mkdir certs
cd certs
openssl req -new -x509 -nodes -out cert.pem -keyout key.pem

Затем копируем сертификаты на другой узел:

scp cert.pem user@remote:/path/to/certs
scp key.pem user@remote:/path/to/certs

где user@remote это имя Turtlebot3.

Копии сертификатов нужны для того, чтобы у ROS и у Turtlebot3 были одинаковые ключи шифрования, иначе они не смогут подключиться друг к другу, но тем самым защищая себя от сторонних подключений.

После установки SSH и TLS переходим к настройке ROS Noetic и Turtlebot3.

Для настройки ROS Noetic и Turtlebot3, редактируем файлы конфигурации ROS и Turtlebot3:

nano ~/.bashrc

Добавляем следующие строки в конец файла:

export ROS_MASTER_URI=http://<master_ip>:11311
export ROS_HOSTNAME=<node_ip>
export ROS_IP=<node_ip>
export ROS_SECURITY_ROOT_DIRECTORY=/path/to/certs
export ROS_SECURITY_ENABLE=true
export ROS_SECURITY_STRATEGY=TLS

Заменяем <master_ip> на IP-адрес ROS Noetic (другими словами это IP компьютера), <node_ip> на IP-адрес текущего узла (это IP вашего компьютера – у Turlebot3 это IP самого робота) и /path/to/certs на путь к папке с сертификатами TLS.

После настроек всех файлов конфигурации и установок всех необходимых протоколов, приступаем к запуску ROS и Turtlebot3

Запускаем ROS Noetic при помощи команды roscore и подключаемся к Turtlebot3 удаленно при помощи команды:

roslaunch turtlebot3_bringup turtlebot3_robot.launch

Заключение

В ходе проведения данного эксперимента можно сделать вывод, что данный метод, который был показан, позволяет операционной системе ROS и роботам Turtlebot3 обмениваться данными друг с другом через зашифрованный канал без сторонних подключений.

Литература

1. Зиктратов И.А., Козлова Е.В., Зикратова Т.В. Анализ уязвимостей робототехнических комплексов с роевым интеллектом // Научно-технический вестник информационных технологии, 2013, №5 (87), с. 149-154.

2. Dieber Bernhard, Breiling Benjamin, Taurer Sebastian, Kacianka Severin, Rass Stefan, Schartner Peter Security for the Robot Operating System // Robotics and Autonomous Systems, 2017, pp. 1-29.

3. Казанцев В.С., Мельников А.О., Русаков А.М., Филатов В.В., Долженков С.С. Применение универсальных состязательных атак в задачах повышения эффективности систем защиты от роботов и спама // Инженерный вестник Дона, 2023, № 6. URL: ivdon.ru/ru/magazine/archive/n6y2023/8460.

4. Эшанкулов Х.И. Многоагентные системы для информационного мониторинга и управления в реальном времени // Universum: технические науки: электрон. научн. журн. 2021. 3(84). URL: cyberleninka.ru/article/n/mnogoagentnye-sistemy-dlya-informatsionnyy-monitoringa-i-upravleniya-v-realnom-vremeni

5. Zaytsev E.I., Khalabiya R.F., Stepanova I.V., Bunina L.V. Advances in Intelligent Systems and Computing. 2020. V. 1156. pp. 131-141.

6. Sehrawat Deepthi. International Journal of Research Aspects of Engineering and Management ISSN: 2348–6627, Vol. 1, Issue 2, June 2014, pp. 95–98.

7. Астанин С.В., Жуковская Н.К. Адаптивное поведение агента: акцептор результатов действий и эфферентный синтез // Инженерный вестник Дона, 2014, №1. URL: ivdon.ru/ru/magazine/archive/n1y2014/2284.

8. Quigley Morgan, Gerkey Brian, Smart William D. Programming Robots with ROS: A Practical Introduction to the Robot Operating System, 2015, p. 417.

9. Grimmett Richard. Raspberry Pi Robotic Projects, 2016, p. 238.

10. Viega John, Messier Matt, Chandra Pravir Network Security with OpenSSL: Cryptography for Secure Communications, 2002, p.386.

11. Lucas Michael W. SSH Mastery: Openssh, Putty, Tunnels and Keys, 2nd Edition, 2018, p. 242.

References

1. Zikratov I.A., Kozlova E.V., Zikratova T.V. Nauchno-tekhnicheskij vestnik informacionnyh tekhnologii, 2013, No. 5 (87), pp. 149-154.

2. Dieber Bernhard, Breiling Benjamin, Taurer Sebastian, Kacianka Severin, Rass Stefan, Schartner Peter Robotics and Autonomous Systems, 2017, pp. 1-29.

3. Kazancev V.S., Mel'nikov A.O., Rusakov A.M., Filatov V.V., Inzhenernyj vestnik Dona, 2023, № 6. URL: ivdon.ru/ru/magazine/archive/n6y2023/8460.

4. Eshankulov H.I. Universum: tekhnicheskie nauki: elektron. nauchn. zhurn. 2021. 3(84). URL: cyberleninka.ru/article/n/mnogoagentnye-sistemy-dlya-informatsionnyy-monitoringa-i-upravleniya-v-realnom-vremeni

5. Zaytsev E.I., Khalabiya R.F., Stepanova I.V., Bunina L.V. Advances in Intelligent Systems and Computing. 2020. V. 1156. pp. 131-141.

6. Sehrawat Deepthi. Simulating Multi-Agent Systems with AnyLogic system: Review. International Journal of Research Aspects of Engineering and Management ISSN: 2348–6627, Vol. 1, Issue 2, June 2014, pp. 95–98.

7. Astanin S.V., Zhukovskaya N.K. Inzhenernyj vestnik Dona, 2014, №1. URL: ivdon.ru/ru/magazine/archive/n1y2014/2284.

8. Quigley Morgan, Gerkey Brian, Smart William D. Programming Robots with ROS: A Practical Introduction to the Robot Operating System, 2015, pp. 417.

9. Grimmett Richard. Raspberry Pi Robotic Projects, 2016, p. 238.

10. Viega John, Messier Matt, Chandra Pravir Network Security with OpenSSL: Cryptography for Secure Communications, 2002, p.386.

11. Lucas Michael W. SSH Mastery: Openssh, Putty, Tunnels and Keys, 2nd Edition, 2018, p. 242.