DOI 10.37882/2223-2966.2024.11.28

АЛГОРИТМ ОБНАРУЖЕНИЯ ВНУТРЕННИХ УГРОЗ В ИНФОРМАЦИОННОЙ СИСТЕМЕ

ALGORITHM FOR DETECTING INTERNAL THREATS IN INFORMATION SYSTEM

D. Reshetnikov
E. Ivanov
E. Amelyutin
A. Selin

Summary. This article presents an algorithm for threat detection and prevention using neural network based on multilayer perceptron and ARIMA model. Based on the algorithm, software is developed and tested with subsequent evaluation of performance in simulation of malicious actions. The aim of the work is to demonstrate an effective algorithm that can detect malicious actions with sufficient accuracy.

Keywords: information systems, internal threats, information security, ARIMA, neural networks.

Решетников Даниил Дмитриевич
МИРЭА — Российский технологический университет
r.daniil1@outlook.com
Иванов Егор Андреевич
МИРЭА — Российский технологический университет
kmaw2@yandex.ru
Амелютин Евгений Вячеславович
Доцент, МИРЭА — Российский технологический университет amelyutin9@yandex.ru
Селин Андрей Александрович
Доцент, МИРЭА — Российский технологический университет chuknor@yandex.ru

Аннотация. В данной статье представлен алгоритм для обнаружения и предотвращения угроз с использованием нейронной сети на основе многослойного персептрона и модели ARIMA. На основе алгоритма разработано программное обеспечение и проведено тестирование с последующей оценкой работы при симуляции злоумышленных действий. Цель работы — продемонстрировать эффективный алгоритм, позволяющий определять злоумышленные действия с достаточной точностью.

Ключевые слова: информационные системы, внутренние угрозы, информационная безопасность, ARIMA, нейронные сети.

Введение

В настоящее время существует большое количество программных решений по идентификации, а также принятия решений по их устранению злоумышленных действий [1]. К ним также относят средства по мониторингу действий пользователей:

1. Отслеживание времени (Tick);
2. Мониторинг сотрудников (Apploye);
3. Обнаружение угроз (OSSEC);
4. Предотвращение угроз (Trellix IPS);
5. Предотвращение от утечек данных (СёрчИнформ КИБ);
6. Анализ поведения пользователей (NuPIC);
7. Управление событиями безопасностями (IBM QRadar).

Программы Tick и Apploye не имеют функционала, ориентированного на безопасность компании, их основная задача — повышение продуктивности сотрудников за счет их контроля на рабочем месте и фокусирования их внимания на работе, а не посторонних задачах.

Программные продукты NuPIC и OSSEC бесплатны в использовании и имеют базовый функционал, для обнаружения злоумышленных действий сотрудников. Однако, NuPIC имеет сложный алгоритм и применяется как библиотека при написании программ. OSSEC, в свою очередь, не имеет большого функционала для обнаружения угроз.

Программа Trellix IPS позволяет анализировать сетевой трафик и собирать информацию с маршрутизаторов и коммутаторов для обнаружения девиантного поведения в сети. СёрчИнформ КИБ предотвращает утечку информации из компании зашифровывая файлы, выходящие за пределы сети. Решение от компании IBM использует многоуровневую оценку рисков на основе искусственного интеллекта, упрощая поиск, поскольку уведомление будет отображено только при самых важных случаях. Такие программы требуют больших вычислительных ресурсов для корректного функционирования. В результате, актуальной задачей в плане обнаружения внутренних угроз является разработка собственного программного обеспечения.

Алгоритм обнаружения угроз

В реализованной системе мониторинга используются три модуля: первый собирает данные для анализа на стороне пользователя, второй выполняет прогнозы на их основе, а третий визуализирует полученную информацию на стороне клиента.

При реализации первого модуля использовалась PostgreSQL с целью сохранения информации в базе данных. В нём также применяется auditd для создания правил и дальнейшего их поиска в системном журнале при помощи journalctl. Цикл работы первого модуля представлен на рисунке 1:

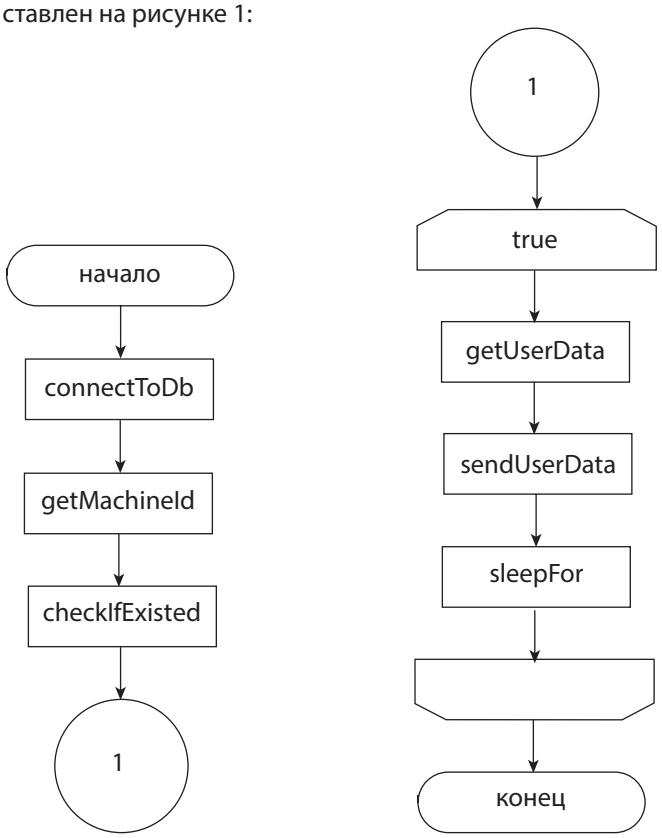

Рис. 1. Цикл работы первого модуля

В первом модуле происходит подключение к базе данных (connectToDb) и получение с компьютера пользователя machineId для его идентификации в базе данных (getMachineId). Затем, используя данный идентификатор, происходит проверка наличия пользователя в базе данных и, при его отсутствии, создает новую запись о новом устройстве (checkIfExisted). Затем в бесконечном цикле происходит сбор (getUserData) и отправление (sendUserData) информации о сетевом трафике и нарушениях из системного журнала в базу данных с заранее заданным интервалом в одну минуту (sleepFor).

При реализации второго модуля для прогнозирования данных о пользователе используются библиотеки ctsa [2] и mlpack [3]. Библиотека ctsa применяется для анализа временных рядов и моделирования на основе модели ARIMA [4]. В свою очередь mlpack позволяет быстро и интуитивно понятной для реализации нейронной сети. Цикл работы второго модуля представлен на рисунке 2.

Во втором модуле происходит подключение к базе данных (connectToDb) и проверки на существование модели на основе многослойного персептрона (isModelExists). После этого в бесконечном цикле запрашиваются данные пользователей из базы данных (getUsersData) и выполняется прогнозирование на основе полученных данных с использованием модели ARIMA (getPredictedFromArima). Затем эта информация отправляется в базу данных (sendPredictedNetworkUsage) и запрашивается текущее количество пользователей (getEmployeeAmount) для получения его действий (retrieveEmployeeActions) и прогнозирования следующего действия (getNextAction), а также получения количества нарушений (retrieveViolationsAmount). После этого следующее действие отправляется в базу данных (sendPredictedAction) и производится приостановка на одну минуту (sleepFor).

Третий модуль использует фреймворк Qt для визуализации данных. Блок-схема работы третьего модуля представлена рисунке 3.

При запуске данной программы происходит установка соединения с базой данных (initConnection). После подключения устанавливается таймер для обновления информации на экране пользователя, применяя запросы к базе данных с периодичностью в одну минуту (launchTimer). Далее одновременно происходит визуализация данных о нагрузке на сетевой трафик (updatePlot), вывод на экран круговой диаграммы о типах нарушений (updateChart), а также таблицу со списком пользователей, количество их нарушений и возможное следующее действие (updateTableData). На рисунке 4 представлено окно графического интерфейса системы мониторинга с отображением информации из тестовой базы данных:

Тестирование и оценка алгоритмов по идентификации угроз

Для проведения тестирования были созданы правила auditd. Их список представлен далее (1):

```
— w /bin/firefox -p x -k firefox_exec
— w /bin/apt -p x -k program_apt_install
— w /bin/dpkg -p x -k program_dpkg_install
— w /etc/shadow -p rwa -k shadow_actions
```
(1)

После обнаружения злоумышленных действий с использованием audtid, информация загружается в базу данных. Далее представлен SQL-запрос, который отображает список пользователей с количеством их нарушений и их возможному следующему действию (2):

```
SELECT      username, violations_amount, name AS
            pred_action
FROM        employee AS e
INNER JOIN  actions_predictions AS a ON e.id =
            a.user_id
INNER JOIN  action_patterns AS p ON predicted_
            action = p.id;
```
(2)

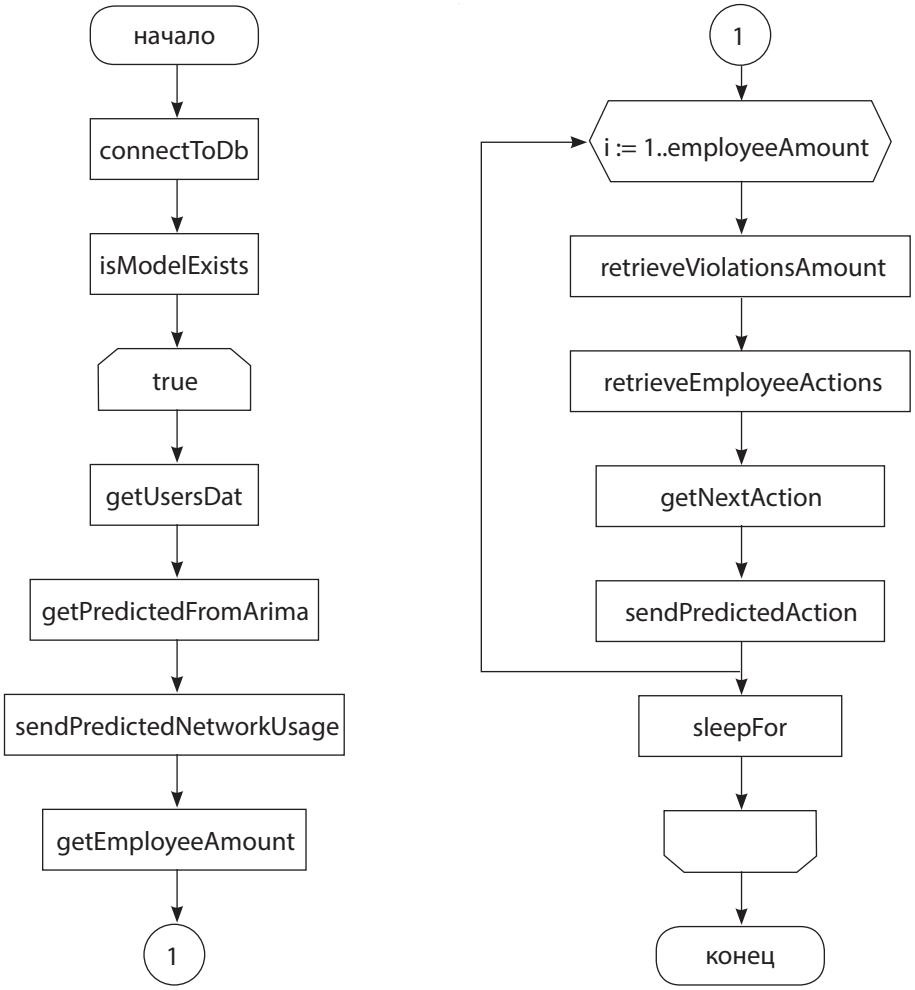

Рис. 2. Цикл работы второго модуля

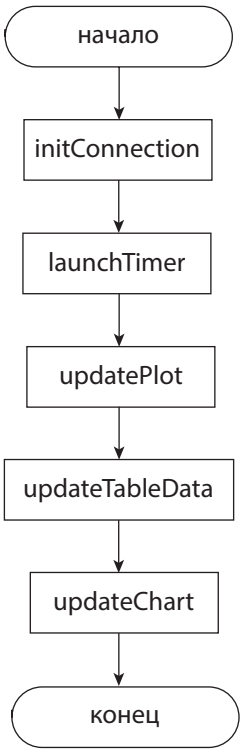

Рис. 3. Цикл работы третьего модуля

Результат выполнения SQL-запроса (2) для отображения данных о пользователей на основе тестовых данных (рисунок 5).

Эффективность работы системы мониторинга рассчитывается относительно точности используемых в ней моделей. Для её проверки используется метрика Accuracy [5]. Она представляет из себя сумму правильных предсказаний, разделённую на общее количество данных (формула 1):

$$ACC = \frac{TP + TN}{TP + TN + FN + FP} \quad (1)$$

где TP — количество истинно положительных результатов,

TN — количество истинно отрицательных результатов,

FN — количество ошибок второго рода,

FP — количество ошибок первого рода.

Для тестирования была взята информация о пользователях из базы данных, заполненная случайными

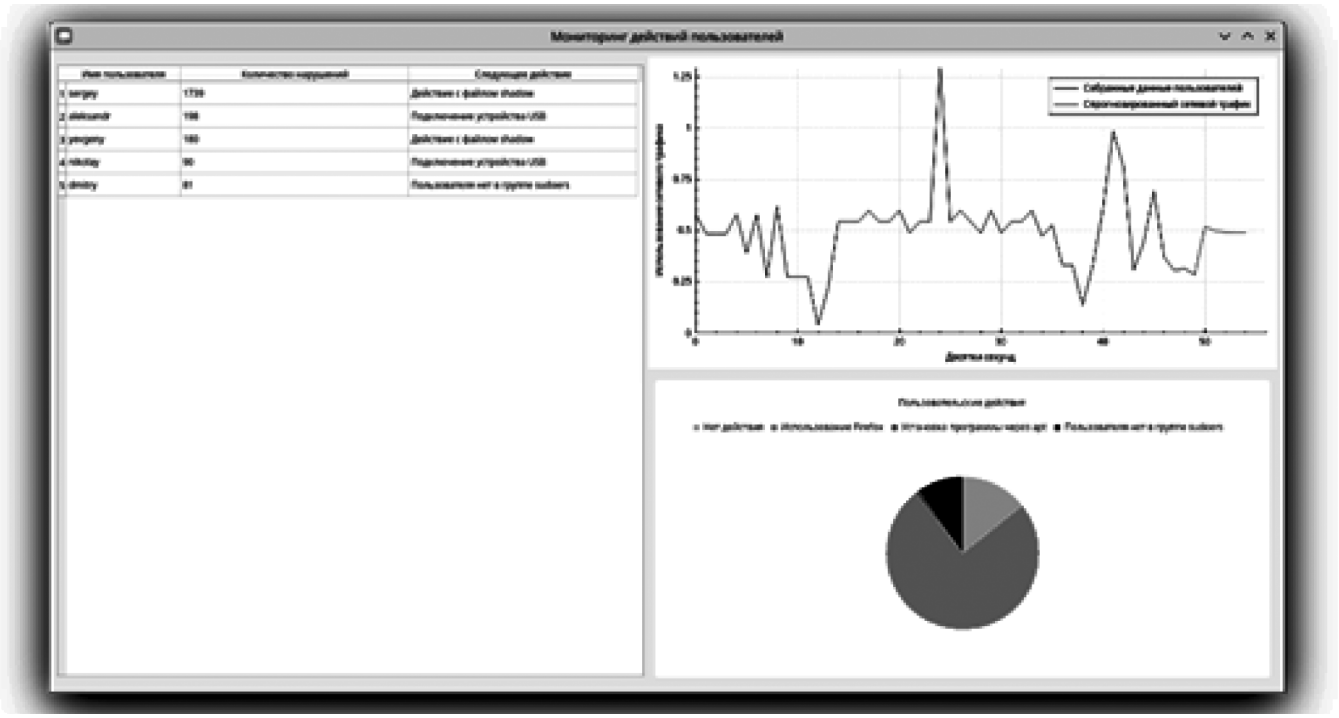

Рис. 4. Графический интерфейс системы мониторинга

	username	violations_amount	pred_action
1	nikolay	90	Подключение устройства USB
2	dmitry	81	Пользователя нет в группе sudoers
3	aleksandr	198	Подключение устройства USB
4	sergey	1739	Действие с файлом shadow
5	yevgeny	180	Действие с файлом shadow

Рис. 5. Результат выполнения SQL-запроса

данными. Полученные сведения представляют 200 различных массивов, содержащих действия пользователя. Перед использованием метрики Accuracy исходные данные для обучения модели были разделены на обучающие (150 ед.) и тестовые (50 ед.).

Формула 1 рассчитывается после обучения модели с последующей визуализацией результата. Результат применения метрики Accuracy (рисунок 6):

```
Train error: 0.946667
Test error: 0.88
```

Рис. 6. Результат расчёта метрики Accuracy

Как видно по рисунку 6, обученная модель обладает высокой точностью к прогнозированию действий пользователя: 95 % правильно спрогнозированных действиях на данных для обучения и 88 % на тестовых.

Для проверки модели ARIMA в качестве метрики используется среднеквадратичная ошибка (MSE) [6]. Формула расчёта ошибки (2):

$$\varepsilon = \hat{x} - x \tag{2}$$

где ε — ошибка модели, x — вектор наблюдений, $\hat{x}$ — прогнозы модели.

Формула MSE (3):

$$MSE(\varepsilon) = \frac{1}{n}\sum_{j=1}^{m}\varepsilon_j^2 \tag{3}$$

где ε — ошибка модели,

n — количество наблюдений.

Тестирование на основе метрики MSE осуществлено аналогично метрике Accuracy — выполнение запроса к базе данных для получения информации о использовании сетевого трафика пользователями (50 ед. вещественных значений). Формула 3 была применена после обучения модели с отображением рассчитанной ошибки. Результат показывает высокую точность модели ARIMA (рисунок 7):

```
Mean squared error ARIMA: 0.0390749
```

Рис. 7. Результат расчёта среднеквадратичной ошибки

Заключение

В статье был описан алгоритм работы для каждой программной компоненты. Кроме того, он позволяет определить злоумышленные действия заранее, за счёт применения модели ARIMA и нейронной сети на основе многослойного персептрона.

Результатом реализации алгоритма стало программное обеспечение по обнаружению злоумышленных действий. Разработанное программное обеспечение позволяет определять девиантное поведение и прогнозировать использование сетевого трафика.

ЛИТЕРАТУРА

1. Employee Monitoring Software — URL: https://www.getapp.com/hr-employee-management-software/employee-monitoring/ (дата обращения 10.06.2024)
2. CTSA: A Univariate Time Series Analysis and ARIMA Modeling Package in ANSI C. — URL: https://github.com/rafat/ctsa (дата обращения 14.07.2024)
3. mlpack: a fast, header-only machine learning library. — URL: https://github.com/mlpack/mlpack (дата обращения 14.07.2024)
4. ARIMA & SARIMA: Real-World Time Series Forecasting. — URL: https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide (дата обращения 14.07.2024)
5. Vujović Ž. et al. Classification model evaluation metrics //International Journal of Advanced Computer Science and Applications. — 2021. — Т. 12. — №. 6. — С. 599–606.
6. Hodson T.O., Over T.M., Foks S.S. Mean squared error, deconstructed //Journal of Advances in Modeling Earth Systems. — 2021. — Т. 13. — №. 12. — С. e2021MS002681.

© Решетников Даниил Дмитриевич (r.daniil1@outlook.com); Иванов Егор Андреевич (kmaw2@yandex.ru);
Амелютин Евгений Вячеславович (amelyutin9@yandex.ru); Селин Андрей Александрович (chuknor@yandex.ru)
Журнал «Современная наука: актуальные проблемы теории и практики»