УДК 004.056 DOI 10.18522/2311-3103-2024-5-38-47

И.В. Котенко, М.В. Мельник

ПРИМЕНЕНИЕ ГИБРИДНОЙ НЕЙРОННОЙ СЕТИ AE-LSTM ДЛЯ ОБНАРУЖЕНИЯ АНОМАЛИЙ В КОНТЕЙНЕРНЫХ СИСТЕМАХ

Популярность контейнерных систем привлекает внимание многих исследователей в области информационных технологий. Технология контейнеризации позволяет сократить расходы вычислительных ресурсов при разворачивании и поддержке сложных инфраструктурных решений. Обеспечение безопасности контейнерных систем и контейнеризации в целом, а также применение злоумышленниками методов реализации "умных" атак на основе искусственного интеллекта, является серьезной проблемой на пути безопасного и устойчивого функционирования контейнерных систем. В статье предлагается подход к обнаружению не только ранее неизвестных отдельных аномальных процессов, но и аномальных последовательностей процессов в контейнерных системах. Предлагаемый подход и его реализация на основе платформы Docker основываются на трассировке системных вызовов, построении гистограмм выполняемых процессов и использовании нейронной сети AE-LSTM. Процесс построения гистограмм базируется на учете количества выполненных системных вызовов для каждого отдельного процесса. Это решение предоставляет возможность не только идентифицировать любой процесс в системе, но и эффективно обнаруживать аномальные последовательности процессов с высокой степенью точности. Созданные последовательности используются в качестве входных данных для нейронной сети. После завершения процесса обучения, нейронная сеть приобретает способность обнаруживать аномальные последовательности, сравнивая заданный порог ошибки реконструкции с фактическим уровнем ошибки входного вектора данных. Когда нейронная сеть сталкивается с новым входным вектором данных, она вычисляет уровень ошибки реконструкции — разницу между ожидаемым и фактическим значением. Если эта ошибка превышает заранее установленный порог, система сигнализирует о наличии аномалии в последовательности. Эксперименты показывают, что предложенный подход демонстрирует достаточно высокую точность обнаружения аномальных процессов при низком уровне ложноположительных результатов обнаружения. Такие результаты подтверждают эффективность предложенного подхода. Затраты вычислительных ресурсов на обучение модели нейронной сети находятся на достаточно низком уровне. Это позволяет использовать менее мощные аппаратные средства без значительных потерь в производительности. Разработанный прототип может быть обучен и внедрен в новую инфраструктуру в достаточно сжатые сроки.

Обнаружение аномалий; системные вызовы; контейнерные системы; нейронные сети; автоэнкодер; долговременная кратковременная память.

I.V. Kotenko, M.V. Melnik

APPLICATION OF A HYBRID NEURAL NETWORK AE-LSTM FOR ANOMALIES DETECTION IN CONTAINER SYSTEMS

The popularity of container systems attracts the attention of many researchers in the field of information technology. Containerization technology allows to reduce the cost of computing resources when deploying and supporting complex infrastructure solutions. Ensuring the security of container systems and containerization in general, as well as the use of smart attacks based on artificial intelligence by malefactors, is a serious problem on the way to the safe and stable operation of container systems. This article proposes an approach for detecting not only previously unknown individual anomalous processes, but also anomalous process sequences in container systems. The proposed approach and its implementation based on the Docker platform are based on tracing system calls, constructing histograms of running processes, and using the AE-LSTM neural network. The process of constructing histograms is based on accounting of the number of executed system calls for each individual process. This solution provides the ability not only to accurately identify any process in the system, but also to effectively detect anomalous process sequences with a high degree of accuracy. The generated sequences are used as input data for the neural network. After completing the training process, the neural network acquires the ability to detect anomalous sequences by comparing a given threshold of reconstruction error with the actual error level of the input data vector. When the neural network encounters a new input data vector, it calculates the reconstruction error level - the difference between the expected and actual value. If this error exceeds a predetermined

threshold, the system signals the presence of an anomaly in the sequence. Experiments show that the proposed approach demonstrates high accuracy in detecting anomalous processes with a low level of false positive detection results. Such results confirm the effectiveness of the proposed approach. Also, the computational costs of training the neural network model are quite low. This allows using less powerful hardware without significant performance losses. Such a solution can be trained and implemented in a new infrastructure in a fairly short time.

Anomaly detection; system calls; container systems; neural networks; autoencoder; long short-term memory.

Введение. В последние годы, использование контейнерных систем позволило повысить удобство в эксплуатации и масштабируемость, а также снизить затраты вычислительных ресурсов, и в настоящее время широко используются при внедрении сложных и высоконагруженных вычислительных систем. Такие преимущества привлекают разработчиков программного обеспечения. Но проблемы безопасности и активное применение методов искусственного интеллекта в проведении атак [1], является серьезным препятствием на пути внедрения технологии контейнеризации.

Только в 2024 году в системах виртуализации было обнаружено несколько серьезных уязвимостей, например, CVE-2024-21626, CVE-2024-23651, CVE-2024-23652 и CVE-2024-23653 [2]. Данные уязвимости позволяют злоумышленнику покинуть пределы изолированных контейнерных систем и получить несанкционированный доступ к данным хостовой операционной системы. Также дополнительные проблем несут возможные нарушения конфигурации, уязвимые образы, пренебрежение механизмами изоляции контейнерных систем и запуск контейнеров с правами администратора (root) [3].

Киберпреступники часто усовершенствуют свои инструменты, тактики и техники, используемые при атаках на информационные системы, что ставит под сомнение использование средств защиты на основе сигнатурного подхода [4]. Решения на основе анализа аномалий [5], которые предполагают использование статистических показателей для обнаружения аномальной активности, таких как системные вызовы (syscalls), потребление вычислительных ресурсов оперативной памяти (RAM), графических процессоров (GPU), центральных процессоров (CPU), быстрых запросов ввода-вывода (Fast I/O request) [6], тоже недостаточно эффективны, поскольку злоумышленник при выполнении, например, сканирования открытых портов с использованием утилиты NMAP [7], может использовать параметры, задающие медленное сканирование (-T0 или -T1).

Поэтому в данной статье представлена методика и программный прототип для обнаружения аномалий в контейнерных системах на основе профилирования. Данный прототип основан на подходе поведенческого анализа, а в роли наблюдаемого объекта выступают контейнерные системы. Каждому контейнеру соответствует эталонный профиль поведения, который построен на основе последовательностей выполняемых процессов в контейнере. Обнаружение аномалий выполняется путем предсказания следующего шага на основе текущего. Если предсказанный шаг отличается от фактического следующего шага – регистрируется аномалия.

Новым элементом предлагаемого решения является создание гистограмм процессов поведения на основе трассировки системных вызовов и использования неконтролируемой гибридной модели глубокого обучения AE-LSTM, комбинирующей автокодировщик (Autoencoder, AE) и долгую краткосрочную память (Long short-term memory, LSTM). Для решения проблемы обработки большого количества данных выполняется построение гистограммы процессов, что позволят сократить количество данных для обучения модели. Гистограммы процессов определяются как вектор данных для обучения модели и последующего обнаружения. После обучения модели осуществляется обнаружение аномалий. Таким образом становится возможным обнаружить ранее неизвестные, а также аномальные последовательности выполняемых процессов в контейнерных системах.

Статья организована следующим образом. Вначале рассматриваются релевантные работы, далее излагается предлагаемый подход к обнаружению аномалий, и представляются особенности обучения и обнаружения. В следующем разделе демонстрируются

предлагаемая реализация, испытательный стенд, используемые наборы данных и результаты экспериментов. В заключении делаются основные выводы и определяются направления будущих исследований.

1. Релевантные работы. Большинство текущих исследований в области аномалий в контейнерных системах основаны на применении трассировки системных вызовов и различных методов глубокого машинного обучения. Они позволяют расширить существующие подходы на основе профилирования, увеличить скорость и точность обнаружения. В [8] предлагается использовать нейронную сеть LSTM для обнаружения аномального поведения путем вычисления значения разности предсказанной последующей последовательности системных вызовов на основе текущей по отношению к действительной последующей последовательности. Аномалия будет зарегистрирована при условии, если порог несовпадения будет превышать допустимый.

Работа [9] основывается на учете шаблонного потребления вычислительных ресурсов во время работы контейнерных систем. Основу предлагаемого решения составляет гибридная нейронная сеть LSTM-BLSTM, использующая комбинацию LSTM и двунаправленной сети долгой краткосрочной памяти (bidirectional LSTM, BLSTM). Предлагаемое решение работает в два этапа. Первый этап, как и в предыдущем исследовании, основывается на прогнозировании будущего значения потребления вычислительных ресурсов и производительности. Второй этап предполагает выполнение классификации ожидаемого следующего значения как нормального или аномального. По экспериментальным результатам отмечено, что предложенная модель работает лучше, чем другие рассматриваемые модели, такие как Марковские модели, рекурентные нейронные сети (Recurrent Neural Network, RNN), управляемые рекуррентные блоки (Gated Recurrent Unit, GRU), LSTM и BLSTM. Высокие результаты прогнозирования достигаются благодаря тому, что модель позволяет интегрировать не только прямые, но и обратные зависимости.

В [10] предлагается система Kubernetes Anomaly Detector (KAD). Отличительной чертой данной системы является использование различных методов машинного обучения для обнаружения различных типов аномалий. Выбор модели осуществляется автоматических из перечная доступных.

Методы глубокого машинного обучения также применяются и в задачах обнаружения вредоносного программного обеспечения. Исследование [11] направлено на обнаружение аномальных подов (pods) кластера Kubernetes, содержащих активный процесс майнингового программного обеспечения, путем анализа системных вызовов. Результаты, полученные экспериментальным путем, демонстрируют точность модели AE-LSTM - 78.9%, а точность ANN - 79,7%, соответственно. В [12] рассматривается система DockerWatch, которая выполняет обнаружение вредоносных файлов. Процесс обнаружения разбит на два этапа: первый этап предполагает быстрый анализ дизассемблированного кода и исходного двоичного файла с использованием сверточной нейронной сети (Convolutional Neural Network, CNN). Второй этап - более медленный и предполагает анализ данных последовательностей вызовов программного интерфейса приложения (application programming interface, API) с использованием гибридной нейронной сети CNN-LSTM.

Следует также выделить работы [13, 14], в которых авторы придерживаются подхода на основе трассировки системных вызовов и использовании автокодировщиков. В [13] собранная последовательность системных вызовов разделяется на несколько отдельных последовательностей. Дальнейшим шагом является построение вектора признаков путем преобразования каждой последовательности в граф. Вектор передается в модель нейронной сети автокодировщика для классификации. Классификация выполняется путем вычисления ошибки реконструкции. Если ошибка реконструкции входного вектора превышает допустимый порог, то это свидетельствует об аномалии. В [14] представлена система KubAnomaly. Как и в [13], собранная последовательность системных вызовов разделяется на небольшие временные отрезки. Для решения проблемы обработки большого количества данных принято решение собирать только четыре категории системных вызовов (файловый ввод-вывод, сетевой ввод-вывод, операции планировщики и памяти). Такое

решение позволило оптимизировать время обучения модели. Особенностью системы KubAnomaly является использование компонента "Планировщик", который управляет всеми остальными компонентами предложенной системы. Точность предложенной модели составляет 96%.

В [15] нейронная сеть AE использована для обнаружения аномальных, ранее неизвестных процессов. Предлагаемый подход основан на ошибке реконструкции, если ошибка превышает заданный порог – регистрируется аномалия.

В [16] исследуется влияние размера окна захвата системных вызовов в качестве входных данных для методов машинного обучения. В том числе исследуются и методы глубокого машинного обучения. В [17] применен метод скользящего окна с использованием гибридной нейронной сети на основе AE и LSTM.

Рассмотренные решения являются ресурсозатратными, сложными в создании и не эффективны для обнаружения атак на контейнерные системы. Следует отметить, что использование нейронных сетей, подобных LSTM и BLSTM, в совокупности с трассировкой системных вызовов без какой-либо оптимизации значительно влияет на скорость обучения модели. Таким образом, подобные решения не могут быть эффективны при внедрении в новые инфраструктуры, поскольку каждая архитектура процессора использует свой набор системных вызовов, а процессы, циркулирующие в инфраструктуре, в которую внедряется решение, могут отличаться от той, на которой проводилось обучение.

В настоящей работе предлагается решение на основе профилирования для реализации обнаружения аномального поведения с использованием гибридной нейронной сети AE-LSTM. Предложенный подход на основе профилирования позволяет значительно сократить время и затраты вычислительных ресурсов на обучение моделей нейронной сети. Также предложенное решение демонстрирует высокие показатели обнаружения при низком уровне ложных срабатываний.

2. Предлагаемый подход. Предлагаемое решение опирается на подход к профилированию поведения контейнерных систем и основано на трассировке системных вызовов и использовании неконтролируемой модели обучения нейронной сети AE-LSTM. Данное решение включает три основных этапа: сбор данных; нормализация; обучение и обнаружение аномалий. В рамках данной работы, под аномалией подразумевается любое отклонение от легитимной активности.

2.1. Сбор данных. На первом этапе выполняется сбор данных об активности контейнерной системы. Сбор системных вызовов и их аргументов может в значительной степени отразиться на скорости обработки и нормализации данных. Поэтому, было решено игнорировать все аргументы системных вызовов.

Сбор данных осуществляется посредством Falco Security [18] с использованием модуля ядра eBPF [19], поскольку данный инструмент обладает широким набором правил, с помощью которых можно гибко настраивать процессы мониторинга контейнерных систем и наблюдать за всеми процессами, происходящими в системе. Преимуществом eBPF является то, что данная технология не требует много ресурсов, поскольку eBPF работает в пространстве ядра и данные не копируются в пространство пользователя.

В сформированном наборе данных содержатся следующие данные: метка времени, имя пользователя, идентификатор контейнера, имя контейнера, имя процесса, идентификатор процесса и системный вызов.

2.2. Нормализация. Следующим этапом выступает обработка и нормализация собранной последовательности данных. Поскольку контейнер в системе не один, следует собрать последовательности системных вызовов характерных для каждого контейнера и отделить их друг от друга. Для этого необходим идентификатор процесса и контейнера.

Процесс нормализации выполняется за пять шагов: (1) модуль отвечающий за нормализацию “Normalizer” читает файл falcoDatasets.csv с последовательностями системных вызовов, построчно, опираясь на идентификатор процесса и идентификатор контейнера; (2) как только встречается идентификатор процесса, отличающийся от текущего, процесс останавливается, формируется строка с гистограммой процесса и записывается в файл характерный для контейнера (trainDatasetsContainerA.csv); имя файла формируется

на основе имени или идентификатора контейнера; (3) так продожается до тех порт пока нормализатор не встретит другой идентификатор контейнера; (4) после того, как нормализатор встретит дургой идентификатор контейнера, отличающий от текущего, все собранные последовательности буду записаны в файл, характеризующий контейнер (trainDatasetsContainerA.csv); (5) модуль нормализатора продолжает работу до тех пор, пока не встретит последнюю строку.

Таким образом, каждому контейнеру будет соответствовать CSV-файл с последовательностью выполненных процессов легитимного поведения.

Особое внимание следует уделить формированию гитограммы процесса (шаг 2 в процессе нормализации). Формирование гитограммы процесса включает два этапа.

На первом этапе выполняется оптимизация конструкций системных вызовов. Количество системных вызовов, которые поддерживаются в Falco, насчитывает 413. Для построения гистограммы необходимо создать массив размером 414 элементов, где каждый элемент этого массива по умолчанию будет равен 0, поскольку учитываются все исполняемые и неисполняемые системные вызовы. Массив размером 414 создается для того, чтобы модель нейронной сети смогла принять оптимальные параметры. Поскольку, слой, который находится между входным и выходным слоем, содержит вдвое меньше нейронов, текущий этап предполагает преобразование в числовое представление собранных системных вызовов и их подсчет.

Например, после выполнения системного вызова write (номер 4 в словаре) дважды и open (номер 1 в словаре) один раз, будет получена гистограмма (вектор), которая представлена в лист. № 1.

Листинг № 1

Гистограмма (вектор) процесса

```
[ 1, 0, 0, 2, 0, 0, 0, 0, 0, … ]
```

На втором этапе, как только гистограмма сформирована, она будет занесена в файл, и все последующие гистограммы выполненных процессов в контейнере также будут занесены в файл.

Таким образом, файл хранит в себе последовательность гистограмм процессов.

Данные, не используемые в процессе построения гистограмм и формирования файлов последовательностей выполненных процессов, необходимы для того, чтобы идентифицировать контейнер и дать оценку выполняемых процессов.

2.3. Обучение и обнаружение. Одним из главных компонентов, реализующих предлагаемый подход, является нейронная сеть AE-LSTM. Цель данной сети заключается в обнаружении аномалий во временных рядах или последовательных данных. С этой целью сеть обучается на легитимных данных и используется для предсказания следующего шага временного ряда или последовательности данных. Предсказанное значение сравнивается со следующим фактическим значением, и вычисляется ошибка предсказания. В случае, если ошибка предсказания существенно отличается от ожидаемой, это указывает на наличие аномалии в данных. В качестве ошибки предсказания выступает пороговое значение. Если ошибка предсказания превышает заданный порог, это означает наличие аномалии в данных. Таким образом, нейронная сеть AE-LSTM обнаруживает аномалии путем анализа изменений в данных и сравнения их с ожидаемыми шаблонами.

3. Реализация. Сбор данных осуществляется с помощью инструмента аудита Falco Security с модулем ядра eBPF. Данные собираются с помощью правил Falco Security. Правила прописаны в файле /etc/falco/falco_rules.yaml. С помощью правил Falco можно собрать практически любую информацию о том, что происходит в системе. Но в этом нет необходимости, поскольку чем больше информации будет собираться, тем больше будет накладных расходов на вычислительные ресурсы.

После того, как набор данных сформирован, он передается в модель нейронной сети AE-LSTM для обучения.

Результатом исходного кода, представленного выше, является модель нейронной сети с использованием LSTM слоев. Сначала создается последовательная модель с несколькими слоями LSTM. Первый LSTM слой с 414 нейронами и функцией активацией ReLU принимает входные данные заданной формы (timesteps, rows) и возвращает последовательности. Следующий LSTM слой с 207 нейронами также имеет функцию активации ReLU и не возвращает последовательности. Затем добавляется слой RepeatVector для повторения входных данных timesteps еще раз. Последующие два LSTM слоя с 207 и 414 нейронами имеют функцию активацию ReLU и возвращают последовательности. Наконец, добавляется слой TimeDistributed Dense для предсказания выходных данных заданной формы. Модель компилируется с оптимизатором "adam" и функцией потерь "mse" (mean squared error). Затем модель обучается на данных X в течение 100 эпох с размером пакета 414. Эта модель предназначена для прогнозирования последовательных данных с использованием LSTM слоев и оптимизатора Adam. Представленный прототип реализован на языке Python 3 с использованием библиотек numpy, pandas, tensorflow и keras.

4. Эксперименты

4.1. Испытательный стенд. Для проведения экспериментов создан испытательный стенд, а также наборы данных легитимной, аномальной и вредоносной активности.

Испытательной стенд включает три контейнера приложения T-Pot [20]. T-Pot - это платформа, которая поддерживает более 20 "ловушек" (honeypots) и графический интерфейс для визуализации атак в режиме реального времени. Контейнеры, задействованные в эксперименте, представлены в табл. 1.

Таблица 1

Описание контейнеров

Контейнер	Описание
cowrie	Cowrie - honeypot SSH / Telnet, предназначен для регистрации атак методом Brute-force
mailoney	Mailoney - SMTP honeypot
elasticsearch	Elasticsearch - honeypot, имитирующий уязвимый сервер Elasticsearch

4.2. Наборы данных. Для создания наборов данных, было использовано несколько контейнеров с имитацией легитимной, аномальной и вредоносной нагрузки. Испытательный стенд включал контейнеры с различным взаимодействием, в том числе контейнеры отправки почтовых сообщений, мониторинга системы с помощью графического интерфейса и прикладного программного обеспечения, авторизации с помощью ssh и работы в контейнере с помощью различных команд.

Информация о собранных наборах данных отображена в табл. 2.

Таблица 2

Наборы данных

Группа наборов данных	Набор данных	Описание
A	cowrie-A	Работа в контейнере с помощью команд (sudo, ls, chmod, pwd, nano, sh, bash, touch, rm, mkdir, cd, mv), авторизация с помощью ssh
	mailoney-A	Отправка почтовых сообщений
	elasticsearch-A	Мониторинг системы, работа с графическим интерфейсом T-pot (просмотр / изменение веб – страниц)
B	cowrie-B	Создание запланированной задачи в контейнере, загрузка файлов с удаленного сервера
	mailoney-B	Сканирование портов, сканирование на наличие уязвимостей
	elasticsearch-B	Сканирование портов

Окончание табл. 2

C	cowrie-C	BackConnect, Шифровальщик, Mining, Brute-force
	mailoney-C	DDOS
	elasticsearch-C	DDOS
D	cowrie-D	Работа в контейнере с помощью команд (sudo, ls, chmod, pwd, nano, sh, bash, touch, rm, mkdir, cd, mv), авторизация с помощью ssh, создание запланированной задачи в контейнере, загрузка файлов с удаленного сервера, BackConnect, Шифровальщик, Mining, Brute-force
	mailoney-D	Отправка почтовых сообщений, сканирование портов, сканирование на наличие уязвимостей, DDOS
	elasticsearch-D	Мониторинг системы, работа с графическим интерфейсом T-pot (просмотр / изменение веб – страниц), сканирование портов, DDOS

Наборы данных группы “A” содержат данные легитимной активности, группы “B” – аномальной активности, “C” – вредоносной активности, а группы “D” – смешанной активности.

Каждый набор данных был сформирован за период 10 минут. В каждом наборе данных была представлена активность 10-ти пользователей. Действия пользователей были организованы в хаотичном порядке с разным интервалом между событиями, чтобы получить максимально правдоподобные наборы данных.

4.3. Обучение моделей. Для обучения моделей, использовались наборы данных легитимной активности, которые представлены в табл. 2. Каждая модель соответствует профилю контейнера. Например, модель нейронной сети, скомпилированная на базе набора данных cowrie-A, соответствует профилю контейнера cowrie. Модели обучены в течении 100 эпох с размером пакета 414 и скомпилированы с оптимизатором "adam" и функцией потерь "mse".

Следует отметить, что в среднем обучение каждой модели длилось 5 минут. Обучение проводилось на испытательном стенде, который включает следующие характеристики: ОС - Cent OS 9; ОЗУ - 32 ГБ; ЦП - AMD Ryzen 5 5600X 6-Core Processor 3.70 GHz.

4.4. Результаты. Обнаружение аномалий выглядит следующим образом: входной вектор последовательности гистограмм процессов подается в обученную модель нейронной сети AE-LSTM. Размер каждой последовательности составляет 10 последовательных гистограмм. На основе этих данных предсказывается следующая последовательность, если предсказанная последовательность отличается от следующей фактической последовательности, то регистрируется аномалия.

Предлагаемое решение позволяет обнаружить не только аномальные процессы, но и аномальных последовательностей процессов, что необходимо экспериментально подтвердить на разных наборах данных. Для проведения экспериментов использованы наборы данных группы “B” с аномальными данными, группы “C” с вредоносными данными и группы “D”, в которую входят легитимная, аномальная и вредоносная активность.

В результате эксперимента любая атака, аномальное или вредоносное поведение обнаруживалось как аномалия, поскольку класс атаки не определялся.

С целью экспериментального подтверждения эффективности предлагаемого решения, была проведена оценка созданных моделей на трех наборах данных B, C и D.

В каждом наборе данных содержится более 500 тысяч записей об активности на испытательном стенде. Для оценки эффективности предлагаемого решения в табл. 3 представлены результаты экспериментов на основе определения элементов матрицы ошибок (Confusion Matrix). В рамках проведенного эксперимента определялось два класса поведения: легитимное и аномалия.

Таблица 3

Результаты проведенного эксперимента

Группа	Набор данных	True		False	
		Positive	Negative	Positive	Negative
B	cowrie-B	**6953**	**0**	**0**	**48**
	mailoney-B	**6953**	**0**	**0**	**147**
	elasticsearch-B	**6999**	**0**	**0**	**18**
C	cowrie-C	**6688**	**0**	**0**	**151**
	mailoney-C	**6743**	**0**	**0**	**113**
	elasticsearch-C	**6914**	**0**	**0**	**99**
D	cowrie-D	**6085**	**1519**	**202**	**291**
	mailoney-D	**5386**	**1805**	**80**	**213**
	elasticsearch-D	**2337**	**1831**	**148**	**116**

Результаты эксперимента показывают высокую точность в обнаружении аномальных последовательностей процессов и аномальных процессов в целом, а уровень ложных срабатываний достаточно низкий.

Заключение. В статье представлен подход к обнаружению аномальной активности в контейнерных системах на основе профилирования действий пользователей и процессов и продемонстрирована реализация этого подхода. Подход включает в себя этапы трассировки системных вызовов, нормализации (построения гистограмм процессов), а также создания профилей легитимного поведения с помощью моделей нейронной сети AE-LSTM. Предложенный подход позволяет проводить обнаружение ранее неизвестных процессов и аномальных последовательностей. Его отличительной особенностью является построение последовательностей гистограмм процессов и использование нейронной сети AE-LSTM.

Результаты проведенных экспериментов продемонстрировали достаточно высокую точность обнаружения ранее неизвестных процессов и аномальных последовательностей процессов и низкий уровень ложных срабатываний (за счет широкого окна захвата последовательностей).

К недостаткам данного подхода можно отнести неспособность обнаруживать медленные атаки. Если атака bruteforce будет работать по таймеру (например, запрос на авторизацию будет выполняться раз в 5 секунд), подход будет не эффективным, поскольку гистограмма процесса авторизации не будет захвачена в окно последовательности. Также следует отметить, что аномальная активность, такая как, работа в нетипичное время, тоже не будет обнаружена, поскольку в рамках представленного подхода не осуществляется привязки ко времени.

Для решения задачи обнаружения аномальной активности, а именно работы в нетипичное время, планируется использовать методы, учитывающие временные ряды. Для обнаружения такой активности разрабатывается алгоритм, который отслеживает появление тех или иных гистограмм в определенный промежуток времени. Если появившаяся гистограмма не характерна для конкретного промежутка времени, будет зарегистрирована аномалия. Также в рамках будущих исследований будут протестирована эффективность обнаружения других атак (SQL-инъекции, PHP-инъекции и др.).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Котенко И.В.* Искусственный интеллект для кибербезопасности: новая стадия противоборства в киберпространстве // Искусственный интеллект и принятие решений. – 2024. – № 1. – С. 3-19.
2. *Priedhorsky R.* Charliecloud is not affected by CVE-2024-21626 or related vulnerabilities, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2024, № LA-UR-24-21089.
3. *Левшун Д.С., Веснин Д.В., Котенко И.В.* Прогнозирование категорий уязвимостей в конфигурациях устройств с помощью методов искусственного интеллекта // Вопросы кибербезопасности. – 2024. – № 3 (61). – С. 33-39.

4. *Applebaum S., Gaber T., Ahmed A.* Signature-based and machine-learning-based web application firewalls: A short survey // Procedia Computer Science. – 2021. – Vol. 189. – P. 359-367.
5. *Pang G., Shen C., Cao L., Hengel A.V.D.* Deep learning for anomaly detection: A review // ACM computing surveys (CSUR). – 2021. – Vol. 54, No. 2. – P. 1-38.
6. *Ahmed M.E., Kim H., Camtepe S., Nepal S.* Peeler: Profiling kernel-level events to detect ransomware // Computer Security–ESORICS 22: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part I 26. – Springer International Publishing, 2021. – P. 240-260.
7. *Liao S., Zhou C., Zhao Y., Zhang Z., Zhang C., Gao Y., Zhong, G.* A comprehensive detection approach of nmap: Principles, rules and experiments // 2020 international conference on cyber-enabled distributed computing and knowledge discovery (CyberC), IEEE, 2020. – P. 64-71.
8. *Snehi J., Bhandari A., Baggan V., Snehi M., Kaur H.* AIDAAS: Incident handling and remediation anomaly-based IDaaS for cloud service providers // 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), IEEE, 2021. – P. 356-360.
9. *Gupta S., Muthiyan N., Kumar S., Nigam A., Dinesh D. A.* A supervised deep learning framework for proactive anomaly detection in cloud workloads // 2017 14th IEEE India Council International Conference (INDICON), IEEE, 2017. – P. 1-6.
10. *Kosińska J., Tobiasz M.* Detection of Cluster Anomalies With ML Techniques // IEEE Access. – 2022. – Vol. 10. – P. 110742-110753.
11. *Karn, R. R., Kudva, P., Huang, H., Suneja, S., Elfadel, I. M.* Cryptomining detection in container clouds using system calls and explainable machine learning, IEEE transactions on parallel and distributed systems, 2020, Vol. 32, №. 3, pp. 674-691.
12. *Wang Y., Wang Q., Qin X., Chen X., Xin B., Yang R.* DockerWatch: a two-phase hybrid detection of malware using various static features in container cloud // Soft Computing. – 2023. – Vol. 27, No. 2. – P. 1015-1031.
13. *El Khairi A., Caselli M., Knierim C., Peter A., Continella A.* Contextualizing system calls in containers for anomaly-based intrusion detection // Proceedings of the 2022 on Cloud Computing Security Workshop. – 2022. – P. 9-21.
14. *Tien C.W., Huang T.Y., Tien C.W., Huang T.C., Kuo S.Y.* KubAnomaly: Anomaly detection for the Docker orchestration platform with neural network approaches // Engineering reports. – 2019. – Vol. 1, No. 5. P. e12080.
15. *Kotenko I., Melnik M., Abramenko G.* Anomaly detection in container systems: using normal process histograms and an autoencoder // 2024 IEEE 25th International Conference of Young Professionals in Electron Devices and Materials (EDM 2024), IEEE, 2024. – P. 1930-1934.
16. *Castanhel G.R., Heinrich T., Ceschin F., Maziero C.* Taking a peek: An evaluation of anomaly detection using system calls for containers // 2021 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2021. – P. 1-6.
17. *Cui P., Umphress D.* Towards unsupervised introspection of containerized application // Proceedings of the 2020 10th International Conference on Communication and Network Security. – 2020. – P. 42-51.
18. *Deri L., Sabella S., Mainardi S., Degano P., Zunino, R.* Combining System Visibility and Security Using eBPF // ITASEC. – 2019. – Vol. 2315. – P. 1-12.
19. *Kotenko I., Saenko I., Chechulin A., Vitkova L., Kolomeec M., Zelichenok I., Melnik M., Makrushin D., Petrevich N.* Detection of Anomalies and Attacks in Container Systems: An Integrated Approach Based on Black and White Lists // International Conference on Intelligent Information Technologies for Industry, Cham: Springer International Publishing, 2022. – P. 107-117.
20. *Subhash P., Qayyum M., Likhitha Varsha C., Mehernadh K., Sruthi J., Nithin A.* A Security Framework for the Detection of Targeted Attacks Using Honeypot // International Conference on Computer & Communication Technologies, Singapore: Springer Nature Singapore, 2023. – P. 183-192.

REFERENCES

1. *Kotenko I.V.* Iskusstvennyy intellekt dlya kiberbezopasnosti: novaya stadiya protivoborstva v kiberprostranstve [Artificial Intelligence for Cybersecurity: A New Stage of Confrontation in Cyberspace], *Iskusstvennyy intellekt i prinyatie resheniy* [Artificial Intelligence and Decision Making], 2024, No. 1, pp. 3-19.
2. *Priedhorsky R.* Charliecloud is not affected by CVE-2024-21626 or related vulnerabilities, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2024, No. LA-UR-24-21089.
3. *Levshun D.S., Vesnin D.V., Kotenko I.V.* Prognozirovanie kategoriy uyazvimostey v konfiguratsiyakh ustroystv s pomoshch'yu metodov iskusstvennogo intellekta [Forecasting vulnerability categories in device configurations using artificial intelligence methods], *Voprosy kiberbezopasnosti* [Cybersecurity Issues], 2024, No. 3 (61), pp. 33-39.

4. *Applebaum S., Gaber T., Ahmed A.* Signature-based and machine-learning-based web application firewalls: A short survey, *Procedia Computer Science*, 2021, Vol. 189, pp. 359-367.
5. *Pang G., Shen C., Cao L., Hengel A.V D.* Deep learning for anomaly detection: A review, *ACM computing surveys (CSUR),* 2021, Vol. 54, No. 2, pp. 1-38.
6. *Ahmed M.E., Kim H., Camtepe S., Nepal S.* Peeler: Profiling kernel-level events to detect ransomware, *Computer Security–ESORICS 22: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part I 26, Springer International Publishing, 2021*, pp. 240-260.
7. *Liao S., Zhou C., Zhao Y., Zhang Z., Zhang C., Gao Y., Zhong, G.* A comprehensive detection approach of nmap: Principles, rules and experiments, *2020 international conference on cyber-enabled distributed computing and knowledge discovery (CyberC), IEEE, 2020*, pp. 64-71.
8. *Snehi J., Bhandari A., Baggan V., Snehi M., Kaur H.* AIDAAS: Incident handling and remediation anomaly-based IDaaS for cloud service providers, *2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), IEEE, 2021*, pp. 356-360.
9. *Gupta S., Muthiyan N., Kumar S., Nigam A., Dinesh D.A.* A supervised deep learning framework for proactive anomaly detection in cloud workloads, *2017 14th IEEE India Council International Conference (INDICON), IEEE, 2017*, pp. 1-6.
10. *Kosińska J., Tobiasz M.* Detection of Cluster Anomalies With ML Techniques, *IEEE Access*, 2022, Vol. 10, pp. 110742-110753.
11. *Karn R.R., Kudva, P., Huang H., Suneja S., Elfadel I.M.* Cryptomining detection in container clouds using system calls and explainable machine learning, *IEEE transactions on parallel and distributed systems*, 2020, Vol. 32, No. 3, pp. 674-691.
12. *Wang Y., Wang Q., Qin X., Chen X., Xin B., Yang R.* DockerWatch: a two-phase hybrid detection of malware using various static features in container cloud, *Soft Computing*, 2023, Vol. 27, No. 2, pp. 1015-1031.
13. *El Khairi A., Caselli M., Knierim C., Peter A., Continella A.* Contextualizing system calls in containers for anomaly-based intrusion detection, *Proceedings of the 2022 on Cloud Computing Security Workshop*, 2022, pp. 9-21.
14. *Tien C.W., Huang T.Y., Tien C.W., Huang T.C., Kuo S.Y.* KubAnomaly: Anomaly detection for the Docker orchestration platform with neural network approaches, *Engineering reports*, 2019, Vol. 1, No. 5, pp. e12080.
15. *Kotenko I., Melnik M., Abramenko G.* Anomaly detection in container systems: using normal process histograms and an autoencoder, *2024 IEEE 25th International Conference of Young Professionals in Electron Devices and Materials (EDM 2024), IEEE. 2024*. pp. 1930-1934.
16. *Castanhel G. R., Heinrich T., Ceschin F., Maziero C.* Taking a peek: An evaluation of anomaly detection using system calls for containers, *2021 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2021*, pp. 1-6.
17. *Cui P., Umphress D.* Towards unsupervised introspection of containerized application, *Proceedings of the 2020 10th International Conference on Communication and Network Security*, 2020, pp. 42-51.
18. *Deri L., Sabella S., Mainardi S., Degano P., Zunino, R.* Combining System Visibility and Security Using eBPF, *ITASEC*, 2019, Vol. 2315, pp. 1-12.
19. *Kotenko I., Saenko I., Chechulin A., Vitkova L., Kolomeec M., Zelichenok I., Melnik M., Makrushin D., Petrevich N.* Detection of Anomalies and Attacks in Container Systems: An Integrated Approach Based on Black and White Lists, *International Conference on Intelligent Information Technologies for Industry, Cham: Springer International Publishing*, 2022, pp. 107-117.
20. *Subhash P., Qayyum M., Likhitha Varsha C., Mehernadh K., Sruthi J., Nithin A.* A Security Framework for the Detection of Targeted Attacks Using Honeypot, *International Conference on Computer & Communication Technologies, Singapore: Springer Nature Singapore*, 2023. pp. 183-192.

Статью рекомендовал к опубликованию д.т.н., профессор Ю.А. Кравченко.

Котенко Игорь Витальевич – Санкт-Петербургский федеральный исследовательский центр Российской академии наук; e-mail: ivkote@comsec.spb.ru; г. Санкт-Петербург; Россия; лаборатория проблем компьютерной безопасности; д.т.н.; профессор.

Мельник Максим Владимирович – e-mail: mkmxvh@gmail.com; лаборатория проблем компьютерной безопасности; аспирант.

Kotenko Igor Vitalievich – Saint Petersburg Federal Research Center of the Russian Academy of Sciences; e-mail: ivkote@comsec.spb.ru; Saint Petersburg; Russia; Laboratory of computer security problems; dr. of eng. sc.; professor.

Melnik Maxim Vladimirovich – e-mail: mkmxvh@gmail.com; Laboratory of computer security problems; postgraduate student.