ВЛИЯНИЕ РАНЖИРОВАНИЯ ИНДИКАТОРОВ АТАК НА КАЧЕСТВО МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ В АГЕНТНЫХ СИСТЕМАХ НЕПРЕРЫВНОЙ АУТЕНТИФИКАЦИИ

DOI: 10.36724/2072-8735-2023-17-8-45-55

Manuscript received 30 May 2023;
Accepted 02 July 2023

Фомичева Светлана Григорьевна,
Санкт-Петербургский государственный университет аэрокосмического приборостроения, г. Санкт-Петербург, Россия, ***levikha@mail.ru***

Ключевые слова: *индикатор компрометации, индикатор атаки, ранжирование индикаторов, методы объясняемого машинного обучения, деревья решений*

Агенты безопасности систем аутентификации функционируют в автоматическом режиме и контролируют поведение субъектов, анализируя их динамику с помощью как традиционных (статистических) методов, так и методов на базе машинного обучения. Расширение парадигмы тканей кибербезопасности актуализирует совершенствование адаптивных объясняемых методов и моделей машинного обучения для систем непрерывной аутентификации. Целью исследования является оценка влияния методов ранжирования индикаторов компрометации, индикаторов атак и иных признаков на точность выявления аномалий сетевого трафика, как части ткани безопасности при непрерывной аутентификации пользователей и сущностей. Использовались вероятностные и объясняемые методы бинарной классификации, а также нелинейные регрессоры на базе деревьев решений. Результаты исследования показали, что методы предварительного ранжирования повышают точность и скорость функционирования у контролируемых ML-моделей в среднем на 7%. У неконтролируемых моделей предварительное ранжирование существенно не влияет на время обучения, но повышает F1-Score на 2-10 %, что обосновывает их целесообразность в агентных системах непрерывной аутентификации. Разработанные в работе модели обосновывают целесообразность механизмов предварительного ранжирования индикаторов компрометации и атак, позволяя создавать прототипы паттернов индикаторов атак в автоматическом режиме. В целом неконтролируемые модели не столь точны, как контролируемые, что актуализирует совершенствование либо объясняемых неконтролируемых подходов к выявлению аномалий, либо подходов на базе методов с подкреплением.

Информация об авторе:
Фомичева Светлана Григорьевна, *к.т.н., профессор, профессор Санкт-Петербургского государственного университета аэрокосмического приборостроения, г. Санкт-Петербург, Россия*

Для цитирования:
Фомичева С.Г. Влияние ранжирования индикаторов атак на качество моделей машинного обучения в агентных системах непрерывной аутентификации // T-Comm: Телекоммуникации и транспорт. 2023. Том 17. №8. С. 45-55.

For citation:
Fomicheva S.G. (2023) Influence of attack indicator ranking on the quality of machine learning models in agent-based continuous authentication systems. *T-Comm*, vol. 17, no. 8, pp. 45-55. (*in Russian*)

Введение

Смена традиционной парадигмы обеспечения безопасности, базирующейся на защите периметра организации, на парадигму нулевого доверия (Zero Trust) привела к кардинальным изменениям в осуществлении процессов аутентификации и верификации пользователей, устройств и программных сущностей. Архитектура нулевого доверия основана на принципе «никогда не доверять, непрерывно проверять и оценивать риски». По данным https://owasp.org/www-project-top-ten/ OWASP (Open Worldwide Application Security Project) угрозы, связанные с нарушением контроля доступа (Broken Access Control) и нарушением идентификации и аутентификации (Broken Authentication) продолжают возглавлять список наиболее опасных нарушений в сфере информационной безопасности (табл. 1).

Таблица 1

Рейтинг угроз безопасности для веб-приложений (динамика с 2017 по 2021 год)

№	Рейтинг атак на 2017 год	Динамика	Рейтинг атак на 2021 год
1	A02:2017 – Injection (Инъекции)	↓ (-2)	A01:2021 – Broken Access Control (Нарушение контроля доступа)
2	A02:2017 – Broken Authentication (Нарушение идентификации и аутентификации)	↓ (-6)	A02:2021 – Cryptographic Failures (Криптографические сбои)
3	A03:2017 – Sensitive Data Exposure (Раскрытие конфиденциальных данных)	↑ (+1)	A03:2021 – Injection (Инъекции)
4	A04:2017 – XML External Entities (Внешние объекты)	new	A04:2021 – Insecure Design (Небезопасный дизайн)
5	A05:2017 – Broken Access Control Нарушение контроля доступа)	↑ (+4)	A05:2021 – Secure Misconfiguration (Неверная конфигурация безопасности)
6	A06:2017 – Secure Misconfiguration (Неверная конфигурация безопасности)	↑ (+1)	A06:2021 – Vulnerable and Outdated Components (Уязвимые и устаревшие компоненты)
7	A07:2017 – Cross-site scripting (XSS) (Межсайтовый скриптинг)	↑ (+3)	A07:2021 – Broken Authentication (Нарушение идентификации и аутентификации)
8	A08:2017 – Insecure Deserialization (Небезопасная десериализация)	new	A08:2021 – Soft and Data Integrity Failures (Ошибки целостности программного обеспечения и данных)
9	A09:2017 – Using Components with Known Vulnerabilities (Использование компонентов с известными уязвимостями)	↑ (+3)	A08:2021 – Secure Logging & Monitoring Failures (Сбои регистрации и мониторинга безопасности)
10	A10:2017 – Insufficient Logging & Monitoring (Некачественность регистрации и мониторинга)	new	A10:2021 – Server-Site Request Forgery Подделка запросов на стороне сервера

Ключевым аспектом архитектуры нулевого доверия является то, что вся деятельность должна регистрироваться и отслеживаться, а любые аномалии, включая подозрительное боковое движение, немедленно отмечаться. Для этих целей инфраструктура безопасности все чаще использует агентный подход организации межкомпонентного взаимодействия, где функционал по обеспечению безопасности реализуется с помощью специализированных программных решений, называемых агентами безопасности. Агенты безопасности функционируют в автоматическом режиме и контролируют поведение субъектов, анализируя их динамику с помощью как традиционных (статистических) методов, так и методов на базе машинного обучения (ML-методов) [1]. Специфика ML-методов для агентов безопасности заключается в обязательном использовании методов объясняемого искусственного интеллекта (XAI), а также необходимостью их самоадаптации к изменению внешнего окружения при сохранении способности выполнять свою основную целевую задачу [2].

В данной работе приведены результаты авторских экспериментов по оценке влияния методов ранжирования индикаторов компрометации (*IoC* – Indicators of Compromise), индикаторов атак (*IoA* – Indicators of Attack) [3] и иных признаков на точность выявления аномалий сетевого трафика, как части ткани безопасности при непрерывной аутентификации пользователей и сущностей.

С этой целью исследовались методы на базе деревьев решений и вероятностные ML-модели при выявлении аномалий, связанных с тремя типами атак: FTP – Patator, SSH-Patator, XSS (Cross-Site Scripting) [https://kali.tools/?p=269]. Эксперименты проводились при использовании отрытых наборов данных (датасетов) в их исторической ретроспективе – NSL – KDD 2009, ISCX 2012. CICIDS2017, Loghub2021 [4], на основе которых после очистки и дополнительной предобработки и разметки был создан авторский композитный датасет для выявления аномалий в системах аутентификации.

Индикаторы угроз безопасности и методы их ранжирования

Персонализированные характеристики поведения субъектов систем безопасности подразумевают регистрацию и последующий динамический анализ последовательностей многомерного наблюдения, которые генерируются окружением субъекта. В частности, среди активно исследуемых подходов в Zero Trust системах рассматривается непрерывная аутентификация (CA – continuous authentication), в ходе которой проводится потоковый майнинг данных (DSM – data stream mining) с целью выявления индикаторов компрометации (*IoC* – Indicators of Compromise), индикаторов атак (*IoA* – Indicators of Attack) и иных признаков аномального поведения субъектов при получении доступа к доверительным средам. Обычно индикаторы компрометации поставляются в виде так называемых потоков или фидов угроз (threat feed) – структурированного списка данных об угрозах. Фиды, как правило, интегрируются в средства мониторинга, анализа и реагирования, например, такие как SIEM (Security Information and Event Management), UEBA (User and Entity Behavioral Analytics) и SOAR (System of Orchestration, Automation and Response). Зачастую индикатор компрометации состоит из типа источника, значения и контекста.

Типом источника может быть IP-адрес, e-mail-адрес, DNS-адрес, URL-адрес, хэши вредоносных файлов по стандартам MD5 или SHA256, процессы, ключи реестра и т.д.

В свою очередь, *IoA* – это набор векторов данных, который дает релевантную информацию для соответствующей атаки. Структуры взаимосвязанных *IoC* через последовательности событий, которые атакующий должен совершить, чтобы добиться успеха при атаке, формируют *IoA* и интерпретируются, как правила баз знаний системы реагирования на инциденты безопасности.

Сравнение особенностей использования *IoC*и *IoA* отражено в таблице 2.

Таблица 2

Особенности использования *IoC*и *IoA*

№	*IoA*	*C*:
1	Обнаруживают известные и неизвестные атаки, для которых могут отсутствовать индикаторы компрометации	Фиксируют случившийся факт нарушения безопасности
2	Постоянное обнаружение в режиме реального времени	Точечное обнаружение, во времени. *IoC* устаревают
3	Абстрагированы от конкретных наблюдаемых случаев и отдельных конкретных инцидентов	Представляют информацию о конкретной угрозе

На рисунке 1 структурно представлена схема формирования *IoA*. В качестве примера можно привести один из наиболее распространенных индикаторов атаки *IoA* – связь между внутренними и внешними хостами с использованием необычных портов (отличных от обычных 80 и 443, которые используются для внешнего трафика) может указывать на злоумышленников, использующих эти порты для связи с вредоносными программами.

Поставщики фидов угроз не выставляют приоритет *IoC* и *IoA*, а могут лишь добавить степень уверенности во вредоносности той или иной сущности, Выставление приоритета событий выявления индикаторов –это задача потребителя услуг безопасности, решение которой является сложным и должно учитывать специфику бизнес-процессов организации. Агрегация, обогащение, очистка и ранжирование подобных индикаторов невозможна без средств автоматизации на базе методов машинного обучения.

Это объясняется:

- Колоссальным объемом регистрируемых уникальных *IoC* (за год, регистрируется более семи миллионов уникальных *IoC*)
- Ограниченном временем жизни *IoC* [https://www.pvsm.ru/informatsionnaya-bezopasnost/324071]
- Отсутствием фиксированной структуры фидов (например, для описания атомарных сетевых и хостовых индикаторов целесообразны плоские структуры в формате txt- или csv-фид, без каких-либо иерархических вложений, а для *IoA* уже требуется фиксировать связки *IoC*-индикаторов, поведенческие паттерны вредоносных программ, техники и тактики (TTP - (Tactics, Techniques and Procedures)) хакерских группировок или схема атаки (Attack pattern)). Для их описания используют форматы json, yaml, xml и такие композитные форматы как STIX 2 [http://docs.oasis-open.org/cti/stix/v2.1/stix-v2.1.html /] и. MISP [https://tools.ietf.org/html/draft-dulaunoy-misp-core-format-13].

Поскольку в данной работе рассматривается влияние методов ранжирования на точность выявления аномалий в системах аутентификации, рассмотрим существующие методы. Среди основных современных методов ранжирования (приоритизация) индикаторов компрометации выделяют:

1) Методы «взвешенных» метрик [3], среди них такие, как «Амстердамская» модель», адаптации амстердамской модели (Jet CSIRT), Decaying Indicators of Compromise Центра реагирования на компьютерные инциденты Люксембурга (CIRCL) https://news.myseldon.com/ru/news/index/213462261.
2) Методы машинного обучения [5-17].
3) Байесовские сети [18].

Амстердамская модель [3] ранжирования *IoC* оперирует четкими коэффициентами, опираясь на «пирамиду боли» (Pyramid Of Pain), которые используются при расчете:

- Экстенсивности – меры количества взаимосвязей между индикаторами компрометации и другими индикаторами, и контекстами.
- Своевременности – меры скорости, с которой источник предоставляет данные по сравнению с другими источниками.
- Полноты – меры полноты данных в источнике относительно общего набора данных всех источников.

К этим мерам добавлены коэффициент для учета включения *IoC* в списки известных невредоносных ресурсов, коэффициент затухания для корректировки скорости устаревания рейтинга и веса коэффициентов. Ядром данной модели ранжирования являются метрики:

- динамика поступления индикаторов компрометации;
- среднее временя жизни индикаторов компрометации;

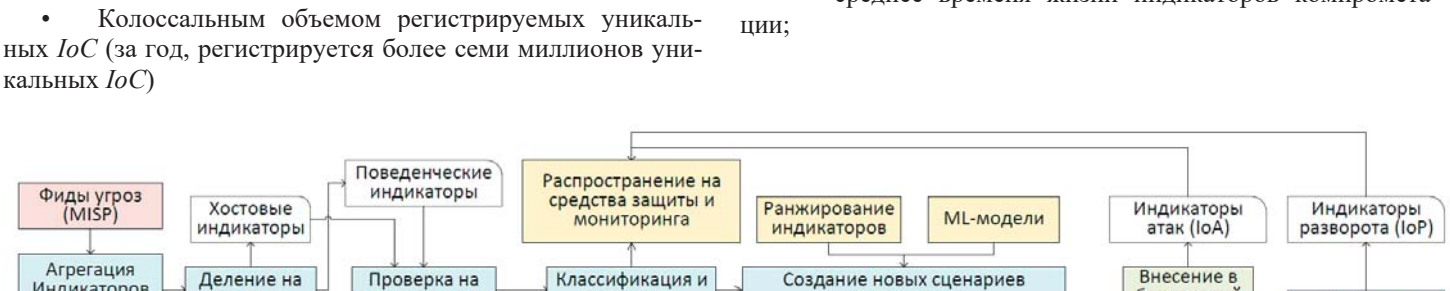


Рис. 1. Жизненный цикл индикаторов атак

- распределение данных по типам объектов, которые предоставляют фиды;
- распределение типов индикаторов компрометации по фидам;
- среднее время обновления индикатора;
- возраст индикаторов компрометации в фиде;
- уникальность данных в различных источниках.

Принцип адаптации амстердамской модели заключается в том, что все индикаторы компрометации, рассматриваются в привязке к принадлежности доменам систем и доменам данных контролируемой IT-инфраструктуры.

Авторы из компании Jet CSIRT исходят из предположения, что домены данных находятся в категориях [https://habr.com/ru/companies/jetinfosystems/articles/459674/]:

- *Real-Time Activity* - активность на источнике (то, что обнаруживается средствами анализа событий безопасности в реальном времени). Например, запускаемые процессы, изменение ключей реестра, создание файлов; сетевая активность, активные подключения и т.п. При обнаружении индикатора данной категории время на реакцию со стороны служб ИБ – минимально, следовательно, весовой коэффициент индикатора большой.
- *Historical Activity* - историческая активность (то, что обнаруживается при ретроспективных проверках): исторические логи; телеметрия; •сработавшие алерты. При обнаружении индикатора данной категории время на реакцию со стороны служб ИБ –допустимо ограничено, следовательно, весовой коэффициент индикатора средний.
- *Data at Rest* - данные, находящиеся в покое (то, что обнаруживается в рамках ретроспективных проверок давно неиспользуемых источников): файлы, которые давно хранятся на источнике; ключи реестра; другие не использующиеся объекты. При обнаружении индикатора данной категории время на реакцию со сторон служб ИБ ограничивается длительностью проведения полного расследования инцидента, следовательно, весовой коэффициент индикатора низкий.

Домены систем определяют принадлежность источника индикатора компрометации к одной из подсистем инфраструктуры:

- *Рабочие станции.* Источники, используемые непосредственно пользователем для выполнения повседневной работы: АРМ, ноутбуки, планшеты, смартфоны, терминалы (VoIP, ВКС, IM), прикладные программы (CRM, ERP, etc.).
- *Серверы*. Здесь имеются в виду остальные устройства, обслуживающие (serve) инфраструктуру, т.е. устройства, обеспечивающие работу ИТ-комплекса: СЗИ (FW, IDS/IPS, AV, EDR, DLP), сетевые устройства, файловые/веб/прокси-серверы, системы СХД, СКУД, контроль окр. среды и т.д.

Комбинируя принадлежность источников доменам данных и систем с составом признаков индикатора компрометации (атомарный *IoC* или композитный), в зависимости от допустимого времени реакции, формируется приоритет инцидента при его детектировании.

CIRCL-подход предполагает [https://news.myseldon.com/ru/news/index/21346226], что приоритет и время жизни некоторых индикаторов не является гомогенным и могут меняться. Время жизни индикаторов задается функцией, характеризующей скорость снижения веса индикатора со временем [3].

Ключевым недостатком вышеперечисленных моделей является необходимость знания особенностей архитектуры анализируемой IT-инфраструктуры. При ее изменении приходится переконфигурировать и систему ранжирования индикаторов. Эту проблему пытаются решить с использованием методов машинного обучения (ML-методов).

В качестве ML-методов при обнаружении вторжений на основе индикаторов компрометации исследовали оценивали эффективность деревьев решений [9], машины опорных векторов [10], метода *k*-ближайших соседей [11], ансамбли данных методов [12] и методы глубокого обучения [13 -15,19]. При этом итоговое дерево решений или обученная модель интерпретируются как прототип паттерна *IoA* (после верификации экспертом- аналитиком паттерн фиксируется в базе знаний как *IoA* (рис.1)).

В данной работе акцент при ранжировании индикаторов сделан на исследовании ML-методов на основе XAI (объясняемого искусственного интеллекта) с целью повысить точность выявления аномалий в агентных системах аутентификации. В исследовании, в силу ограниченности объема статьи. не отражены перспективные ML-методы на базе иерархических нейро-нечетких сущностей, предложенные автором данной работы. Однако, теоретическое обоснование возможностей применения иерархических нейро-нечетких сущностей автором представлено в работах [1, 2, 18, 20], а структура и принципы построения таких интеллектуальных классификаторов для систем аутентификации описано в [21].

Фиды угроз безопасности и датасеты для систем аутентификации

Качество ML-моделей напрямую зависит от полноты и непротиворечивости используемых для обучения и тестирования наборов данных. Данное исследовании проводилось, исходя из следующей эвристики.

Пусть *Features* - множество признаков некоторого датасета, а *Feeds* – множество признаков фида. Тогда исходим из предположения, что:

$$\begin{cases} Features \cap Feeds = \varnothing \\ Features_j^A \cap Feeds_j \approx Feeds_j = \bigcup\limits_i IoC_i^j, \\ Ranking\left(Features_j^A\right) \cap Feeds_j \rightarrow IoA_j \end{cases} \quad (1)$$

где $Features^A{}_j$ - подмножество признаков датасета, свойственных аномалии при зафиксированной атаке *j*-того типа, а IoA_j- индикатор атаки *j*-того типа, $IoC^j{}_i$ – совокупность индикаторов компрометации, релевантных атаке *j*-того типа.

Предположение (1) требует наличия датасетов, содержащих признаки, релевантные индикаторам компрометации и атак соответствующего типа. Кроме того, поскольку индикаторы компрометации имеют ограниченный жизненный цикл, было принято решение учесть возможность деградации индикаторов за счет использования нескольких открытых датасетов (табл.3), применяемых при выявлении аномалий, в их исторической ретроспективе с шагом создания 3-4 года – NSL-KDD 2009, ISCX 2012. CICIDS 2017, Loghub 2021.

Таблица 3

Характеристики используемых датасетов

Характеристики датасета	NSL-KDD 2009	ISCX 2012	CICIDS 2017	Loghub 2021
Объём данных	~5 ГБ	~5 ГБ	~12 ГБ	>75 Гб
Разделение на обучающую и тестовую выборку	Да	Да	Нет	Нет
Маркировка сетевых атак	Да	Да	Да	Частично
Используемые протоколы HTTP, HTTPS, FTP, SSH	Да	Да	Да	Да
Способ создания трафика	Реальный	Реальный	Реальный	Реальный
Потребность в очистке и предобработке	Высокая	Высокая	Средняя	Низкая

Набор данных NSL-KDD 2009 [https://www.kaggle.com/datasets/hassan06/nslkdd] в оригинальном виде содержит большое количество ошибок разметки классов атак, устаревшие типы атак и уязвимостей, ограниченное число типов атак. Это потребовало существенной работы по верификации меток и пополнению датасета контекстом фидов соответствующего года, взятом с платформы MISP. Устаревшие типы атак не удалялись.

Датасет ISCX 2012 [https://gitlab.com/santhisenan/ids_iscx_2012_dataset] создан с использованием реальных устройств, реальных вредоносных и нормальных потоков, включая FTP, HTTP, IMAP, POP3, SMTP и SSH протоколы. Все данные корректно промаркированы. Однако, набор данных ISCX 2012 не включает в себя SSL/TLS, которые сегодня составляют большую часть интернет-трафика

Набор данных CICIDS2017 [https://www.kaggle.com/datasets/cicdataset/cicids2017] подготовлен по результатам анализа сетевого трафика в изолированной среде, в которой моделировались действия 25 легальных пользователей, а также вредоносные действия нарушителей. Набор объединяет более 50 Гб «сырых» данных в формате PCAP и включает 8 предобработанных файлов в формате .csv, содержащих размеченные сессии с выделенными признаками в разные дни наблюдения. Он состоит из пятидневного потока данных сети, использующей с операционными системами, Windows Vista / 7 / 8.1 / 10, Mac, Ubuntu 12/16 и Kali. В наборе данных имеется 15 различных типов меток и 85 признаков. Одна метка (Benign) соответствует штатному режиму без атаки, другие 14 представляют собой режим атаки. Из этих 14 видов атак, лишь пять направленны непосредственно на аутентификацию. Запись без использования атаки, формируемая с использованием почтовых сервисов, протоколов SSH, FTP, HTTP и HTTPS, представляет собой безопасный/обычный поток данных в сети, созданный путем имитации реальных пользовательских данных.

Loghub [https://zenodo.org/record/3227177#.ZFQ2_nZByx0] – большая коллекция наборов данных log-журналов, включая распределенные системы, суперкомпьютеры, операционные системы, мобильные системы, серверные приложения и автономное программное обеспечение.

Общий объем всех этих журналов составляет 77 ГБ. В составе Loghub две категории наборов данных журналов: пять размеченных и 12 неразмеченных.

В частности, два размеченных набора журналов созданы для распределенной файловой системы HDFS: HDFS-1 и HDFS-2. HDFS-1 генерируется в 203-узловой HDFS с использованием эталонных рабочих нагрузок и размечен вручную с помощью также разработанных вручную правил для выявления аномалий. Кроме того, HDFS-1 также предоставляет информацию о конкретном типе аномалии и позволяет проводить исследования по выявлению повторяющихся проблем. HDFS-2 собран путем агрегирования журналов из кластера HDFS в лаборатории создателей датасета CUHK, который включает в себя один *namenode* и 32 узла данных.

Журналы агрегируются на уровне узла и имеют огромный размер (более 16 ГБ) и предоставляются как есть без дальнейших изменений. Кроме этого, в датасете есть данные журналов операционных систем Widows, Linux, Mac, мобильных систем и серверных приложений. Основные характеристики использованных в данной работе датасетов из коллекции Loghub приведены в таблице 4.

Таблица 4

Характеристики использованных датасетов из коллекции Loghub 2021

Система	Описание log-журнала	Период наблюдения	Количество записей	Объем датасета	Наличие разметки
Распределенные системы					
HDFS	Hadoop distributed file system log	38,7 часов	11175629	1,47 Гб	Да
HDFS	Hadoop distributed file system log	Не фиксирован	71118073	16,06 Гб	Нет
Операционные и мобильные системы					
Windows	Windows event log	226,7 дней	114608388	26,09 Гб	Нет
Linux	Linux system log	263,9 дней	25567	2,25 Мб	Нет
Mac	Mac OS log	7 дней	117283	16,09 Мб	Нет
Android	Android framework log	Не фиксирован	1555005	183,37 Мб	Нет
Серверы					
Apache	Apache web-server error log	263,9 дней	56481	4,90 Мб	Нет
OpenSSH	OpenSSH server log	28,4 дней	655146	70,02 Мб	Нет

Поскольку все рассмотренные выше датасеты содержат записи различных типов атак, с целью сокращения анализируемых экземпляров дни, в которых использовались атаки: DoS, PortScan, Infiltration, Bot, DDoS были исключены в силу того, что они напрямую не влияют на процесс аутентификации. В качестве магистральной была выбрана структура признаков датасета CICIDS2017, а в качестве магистрального метода ранжирования признаков (включая IoC), выбрана регрессия случайного леса (Random Forest Regression). Данный выбор обоснован тем, что он способен обрабатывать большие объемы данных и выявлять нелинейные зависимости между переменными. Random Forest позволяет работать с данными разных типов (например, числовыми, категориальными, бинарными) и обнаруживать скрытые взаимосвязи между ними.

Кроме того, деревья решений относят кия к объясняемым методам XAI. В качестве инструмента использован Scikit-learn – библиотека машинного обучения на Python.

Очищенный композитный набор данных, из которого исключены данные на прямую не относящиеся к аутентификации, содержит 1 146 193 записей с 85 признаками, которые определяют свойства потока, такие как идентификатор потока, IP-адрес источника, порт источника и другие. Записей с FTP-Patator атаками – 47938, SSH-Patator атаками – 35897, Web Attack – Brute Force атаками – 21507, Web Attack – XSS атаками – 6652, Web Attack – Sql Injection атаками – 921. Проблема несбалансированности набора данных решалась сокращением числа штатных записей при формировании как обучающей, так и тестовых выборок.

Задача обнаружения аномалий моделировалась как проблема бинарной классификации. В качестве основной оценочной метрики использована *F1-Score* в силу типичной для алгоритмов классификации при оценке показателя точности. Метрика *F1-Score* оценивает баланс между *Precision* и *Recall*, вычисляя их среднее гармоническое. Если *F1 Score* =1, это указывает на идеальную точность и полноту:

$$F1\_Score = 2 \cdot \frac{Recall \times Precision}{Recall + Precision} = \frac{2TP}{2TP + FP + FN}. \quad (2)$$

где *Precision* —основная оценочная метрика при работе с несбалансированными данными, определяемая выражением:

$$Precision = \frac{количество\ истинно\ положительных\ прогнозов}{количество\ правильных\ прогнозов} = \frac{TP}{TP + TN}. \quad (3)$$

Recall определяется выражением (4) и указывает на пропущенные положительные прогнозы, в отличие от метрики *Precision* (3):

$$Recall = \frac{TP}{TP + FN}. \quad (4)$$

Чем ближе *Recall* к 1, тем лучше модель, поскольку она не пропускает истинно положительные результаты. Гораздо хуже, если модель верифицирует, например, некоторых авторизованных пользователей как неавторизованных и откажет им в доступе к системе. Метрика *Recall* в этом случае работает лучше, чем *Precision*.

Метрика *Accuracy* интуитивно понятна и проста в реализации: (5), варьируется от 0 до 1 и используется для простых моделей и сбалансированных датасетов:

$$Accuracy = \frac{количество\ правильных\ прогнозов}{общее\ количество\ прогнозов} = \frac{TP + TN}{TP + FP + TN + FN}. \quad (5)$$

Типы анализируемых атак

Во всех используемых датасетах запись без использования атаки, формируемая с использованием почтовых сервисов, протоколов SSH, FTP, HTTP и HTTPS, представляет собой безопасный/обычный поток данных в сети, созданный путем реальных пользовательских данных.

Атака FTP-Patator – это атака грубой силы, направленная на захват действительного имени пользователя и пароля при использовании сетевого протокола FTP, который обеспечивает передачу файлов между клиентом и сервером в сети. Большое количество неудачных попыток входа за короткий промежуток времени является характерной особенностью для атак грубой силы. Поэтому можно наблюдать плотный поток пакетов во время атаки. Кроме того, неудачные попытки входа не содержат файлов большого размера, поэтому потребление полосы пропускания и количество байтов низкие.

Атака SSH-Patator. SSH (Secure Shell) – это криптографический протокол, который позволяет безопасно работать с различными сетевыми сервисами по сети в незащищенной среде. Наиболее распространенным использование SSH – удаленный доступ к системе. Соответственно, цель данной атаки получить удаленный доступ. Атака SSH-Patator включает три шага:

1) Этап сканирования. Целью этого этапа является попытка узнать сведения о целевой системе. Атакующий пытается найти узел, использующий SSH, путем выполнения сканирования определенного номера порта для IP-блоков в сети или подсети.

2) Этап грубой силы: на этом этапе атакующий пытается войти в систему, которую он обнаружил на этапе сканирования, используя большое количество комбинаций имени пользователя и пароля.

3) Этап вымирания: Атакующий получает полномочия легитимного пользователя, успешно войдя в систему.

Если на первом этапе этой атаки наблюдается большое количество незавершенных TCP-пакетов с SYN флагами, на втором этапе (перебор) наблюдается небольшое количество завершенных TCP-пакетов небольшого размера. Количество пакетов в потоке велико, но размеры пакетов малы.

Атаки SSH-Patator и FTP-Patator часто входят в состав более сложных атак, и, в частности, перед или в процессе реализации web-атак.

Атака XSS (Cross-Site Scripting) – тип веб-атаки, которая реализуется путем внедрения кода в скрипты веб-страниц. При просмотре скомпрометированной веб-страницы, внедренный фрагмент вредоносного кода может привести к нежелательным результатам, таким как кража данных, захват сеанса, выполнение специального кода и т.п.

Атака SQL-инъекций реализуется при попытке доступа к базе данных, используя соединения между веб-приложением и базой данных, что грозит возможностью кражи, удаления или изменения данных, отправив вредоносные sql-запросы на сервер базы данных.

В процессе обучения применялось два сценария – обучение ML-моделей на конкретном типе атак и со всеми вышеперечисленными совместно. Для первого подхода создавался отдельный csv-файл для каждого типа атаки. Этот файл содержит все записи, маркированные конкретным типом атаки, и записи, выбранные из штатного потока с учетом интервала атаки (до и после атаки).

Обнаружение аномалий

Поскольку основу объясняемых ML-методов составляет класс моделей на базе деревьев решений, представим процесс построения дерева решений в формальном виде.

Пусть датасет задан в виде векторов обучения, $x_i \in X^m$, $i=1,\ldots,l$ и векторов соответствующих меток $y_i \in X^l$. Дерево решений рекурсивно разбивает пространство признаков таким образом, что образцы с одинаковыми метками или сходными целевыми значениями группируются вместе.

Пусть данные в узле n дерева решений представлены множеством Q_n с m_n образцами (samples). Кандидат на точку разбиения обозначим через $\theta(j,t_n)$ где j – некоторая функция, а t_n пороговое значение для разбиения данных на $Q_n^{left}(\theta)$ и $Q_n^{right}(\theta)$ подмножества.

Тогда правило разбиения описывается следующим выражением:

$$\begin{cases} Q_n^{left}(\theta) = \{(x,y) | x_i < t_n\} \\ Q_n^{right}(\theta) = Q_n \setminus Q_n^{left}(\theta) \end{cases}, \quad (6)$$

Качество возможного разбиения $G(Q_n,\theta)$ узла n вычисляется с использованием H-функции потерь (Loss-функция), выбор которой зависит от решаемой задачи (классификация или регрессия):

$$G(Q_n,\theta) = \frac{m_n^{left}}{m_n} H\left(Q_n^{left}(\theta)\right) + \frac{m_n^{right}}{m_n} H\left(Q_n^{right}(\theta)\right). \quad (7)$$

В итоге выбирается параметр θ^*, который минимизирует функцию потерь:

$$\theta^* = \arg\min_\theta \left(G\left(Q_n,\theta\right)\right). \quad (8)$$

Рекурсия указанных выше разбиений (6) – (8) выполняется для подмножеств $Q_n^{left}(\theta)$ и $Q_n^{right}(\theta)$ до тех пор, пока не будет достигнута максимально допустимая глубина m_n<min или же m_n=1.

Функция потерь $H(Q_n)$ для задачи классификации задается выражением (9), а для задачи регрессии – выражением (10):

$$H(Q_n) = -\sum_k p_{nk} \log(p_{nk}), \quad (9)$$

где k – количество классов классификации, а p_{nk} – пропорция наблюдений класса k в узле n.

Если целью является непрерывное значение (решается задача регрессии), то для узла n общими критериями для минимизации при определении местоположений для будущих разбиений являются среднеквадратическая ошибка (MSE или ошибка $L2$), отклонение Пуассона, а также средняя абсолютная ошибка (MAE или ошибка $L1$):

$$H(Q_n) = \frac{1}{m_n} \sum_{y \in Q_n} \left(y - \overline{y_n}\right)^2, \quad (10)$$

где $\overline{y_n} = \frac{1}{m_n} \sum_{y \in Q_n} y$ – среднеквадратическая ошибка, y – численное значение метки.

Среди основных современных алгоритмов обнаружения аномалий (типичная задача бинарной классификации) выделяют [14] «случайный лес» (Random Forest), «сокращение измерений» ;(Dimension Reduction, например, алгоритм PCA), «изолированный лес» (Isolation Forest). PCA является неконтролируемым методом, а Random Forest – контролируемым, Isolation Forest применим как к контролируемым, так и неконтролируемым подходам к обучению.

Класс Random Forest Regressor из Sklearn использован при вычислении весов важности признаков (включая *IoC*). Гиперпараметры Random Forest Regressor были установлены в следующие значения *n_estimators*=100, *max_depth*=90, *min_samples_split*=10, *min_samples_leaf*=3. В результирующем лесу решений каждому признаку присвоен вес в зависимости от того, насколько он полезен при построении дерева решений. По завершении процесса эти веса важности признаков отсортированы (рис. 2, 4-6). Сумма весов важности всех свойств дает общий вес дерева решений. Доля в процентах любого признака от веса всего дерева, взятого за 100%, дает информацию о важности этого признака в дереве решений. На рисунке 4 продемонстрирована динамика доминирующих признаков при FTP-Patator атаке.

Анализатор на базе Isolation Forest предварительно обрабатывает все файлы лог-журналов и сохраняет журналы действий пользователей. Для каждого пользователя система извлекает набор характеризующих их признаков (feturias – фичей) и строит базовую модель пользователя (base line tree), создавая набор расширенных функций – лес деревьев изоляции (Isolated Forest). Когда в лог-журнале появляется новая запись о действиях пользователя, она сопоставляется с каждым из этих деревьев изолированного леса и вычисляется оценка аномалий. Если полученная оценка аномалии ниже априорно заданного порогового значения, она считается нормальной, в противном случае регистрируется аномальное поведение, и этот пользователь/сущность помечается как аномалия. Изолированный лес обладает рядом преимуществ в качестве алгоритма обнаружения аномалий в агентных системах аутентификации:

- Для получения функции обнаружения аномалий требуются относительно небольшие выборки из больших наборов данных. Это делает его быстрым и масштабируемым.
- Не требуются примеры аномалий в наборе обучающих данных.
- Его пороговое значение расстояния для определения аномалий основано на глубине дерева, которая не зависит от масштабирования размеров набора данных.

Поскольку для формирования индикаторов атаки следует учитывать прежде всего индикаторы компроментации, соответствующие оранжевому и красному уровню «пирамиды боли», из 85 признаков были исключены признаки, присущие зеленым уровням (Flow ID, Source IP, Source Port, Destination IP, Destination Port, Protocol, Timestamp, External IP), в силу того, что они легко атакующим маскируюися или меняются. Оставшиеся признаки ранжировались. Полученные в результате ранжирования профили (рис. 2, 4, 5) образуют прототип паттерна атак (*IoA*). Результат ранжирования признаков атак SSH-Patator и FTP-Patator указывает на доминирующий признак Fwd Packet Length Max, имеющий наибольшой вес. Это связанно с похожими схемами проведения атак этих типов.

В свою очередь, существенные признаки веб-атак отличаются по составу доминирующих признаков.

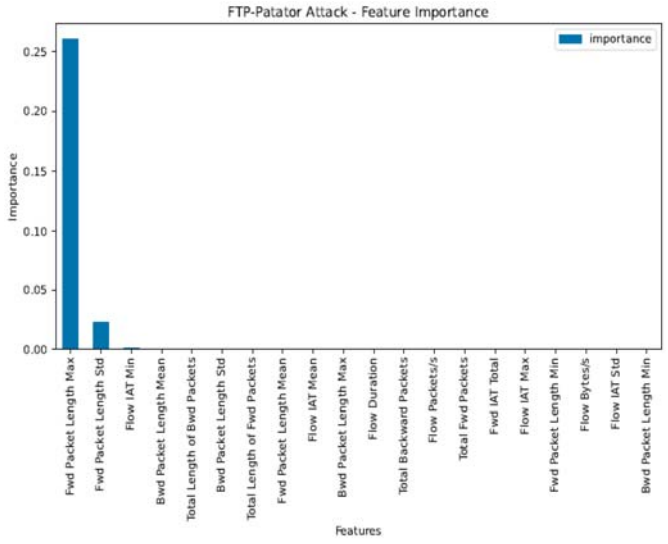
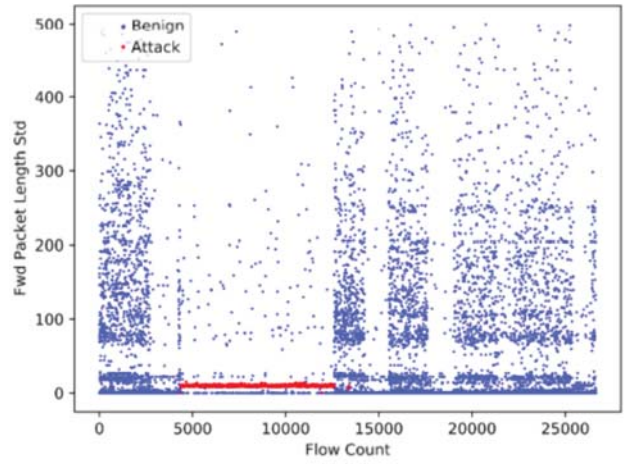
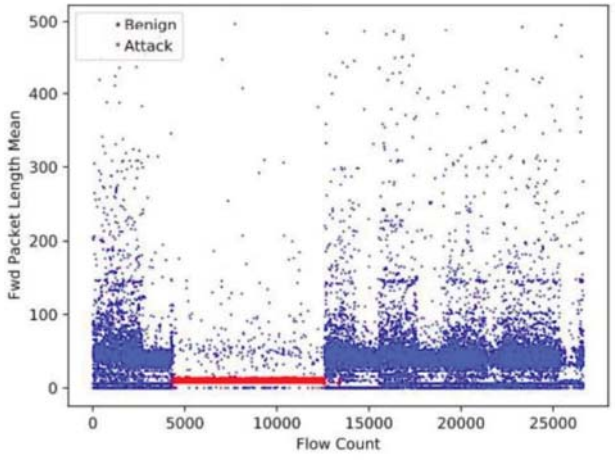


Рис. 2. Результат ранжирования признаков атаки FTP-Patator



а) Динамика "Fwd Packet Length Std" при FTP-Patator атаке



б) Динамика "Fwd Packet Length Mean" при FTP-Patator атаке

Рис. 3. Динамика доминирующих признаков при FTP-Patator атаке (фрагмент траффика)

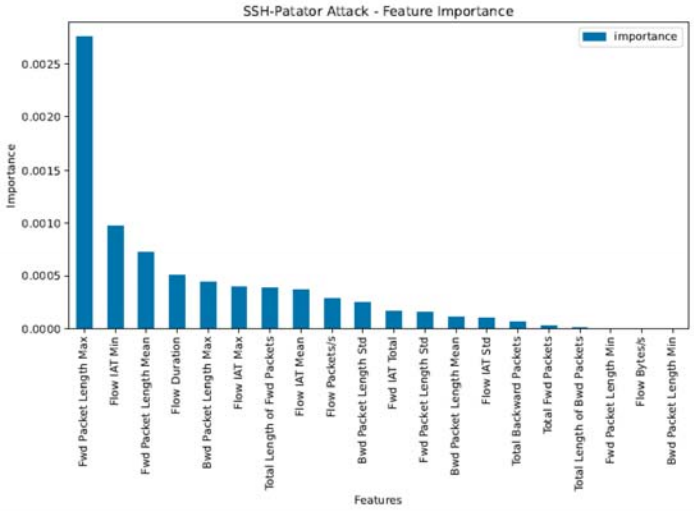


Рис. 4. Результат ранжирования признаков атаки SSH-Patator

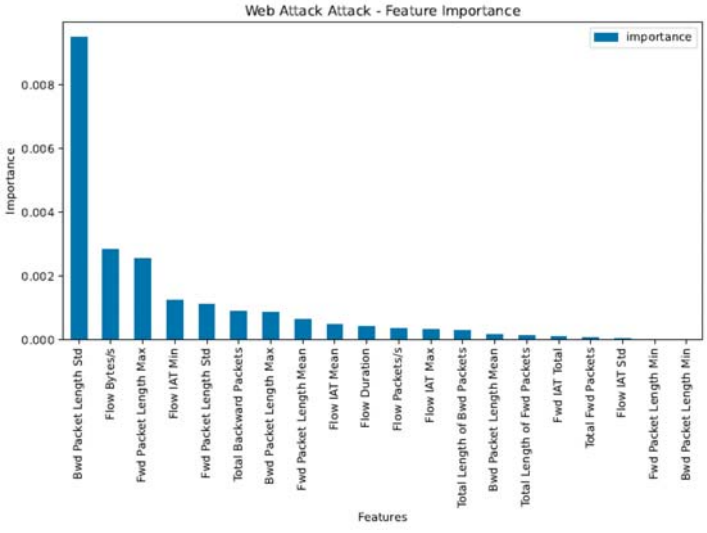


Рис. 5. Результат ранжирования признаков Web-атаки

Профиль рейтинга для случая совмещенных атак (рис. 6) нельзя интерпретировать как прототип паттерна конкрентого *IoA*, однако он полезен для выявления доминирующиз фичей систем аутентификации в целом и может быть проинтерпретирована как индикатор разворота IoPivot (*IoP*), характеризующий боковое движение сложной атаки:

$$IoP(t) = \varphi\left(\bigcup_j IoA_j \otimes \bigcup_i IoC_i(t)\right) \quad (11)$$

где φ – функциональная зависимость, выявленная в ходе оюучения ML-модели, t – дискретное время наблюдений трафика, j – количество выявленных паттернов для IoA, i – число домонирующих признаков индикаторов компроментации IoC.

При установике для (11) порогового значения веса признака равным 1% от общей суммы весов признаков, выделены семь доминирующих признаков, покрывающих 95,8% общего веса признаков. Список семи доминирующих признаков приведен в таблице 5.

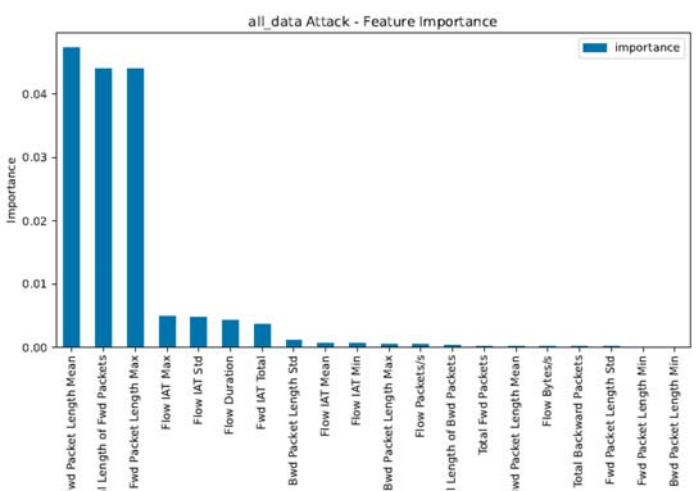


Рис. 6. Результат ранжирования признаков при совмещении атак

Таблица 4

Веса признаков для сценария со совмещенными атаками

Наименование признака	Вес признака	Наименование признака	Вес признака
Fwd Packet Length Mean	0.047260	Bwd Packet Length Max	0.000562
Total Length of Fwd Packets	0.043968	Flow Packets/s	0.000493
Fwd Packet Length Max	0.043883	Total Length of Bwd Packets	0.000318
Flow IAT Max	0.004879	Total Fwd Packets	0.000243
Flow IAT Std	0.004756	Bwd Packet Length Mean	0.000157
Flow Duration	0.004255	Flow Bytes/s	0.000156
Fwd IAT Total	0.003606	Total Backward Packets	0.000124
Bwd Packet Length Std	0.001099	Fwd Packet Length Std	0.000111
Flow IAT Mean	0.000696	Fwd Packet Length Min	0.000054
Flow IAT Min	0.000616	Bwd Packet Length Min	0.000012

Таблица 5

Доминирующие признаки при аутентификации

Наименование признака	Значение веса	Весовая доля, %
Fwd Packet Length Mean	0,047260	29.68
Total Length of Fwd Packets	0,043968	27.63
Fwd Packet Length Max	0,043883	27.57
Flow IAT Max	0,004879	3.06
Flow IAT Std	0,004756	2.99
Flow Duration	0,004255	2.67
Fwd IAT Total	0,003606	2.26

Результаты исследования ML-моделей для выявления аномалий с учетом проранжированных признаков отражены в таблице 6.

Хотя контролируемые методы усчтойчиво демострируют высокие значения метрик точности, однако неконтролируемые методы более применимы в реальной производственной среде, поскольку, во-первых. аномалии значительно реже встречаются в реальных системах и, во-вторых. маркировка данных требует много времени.

Для агентных систем непрерывной аутентификации в подавляющем большинстве используются неконтролируемые методы, реже – методы с подкреплением.

Таблица 6

Оценка качества обученных ML-иоделей при выявлении аномалий в системах непрерывной аутентификации

ML-модель	Тип решаемой задачи	Принцип обучения	Метрика. (без ранжирования/ с ранжированием)				Время обучения
			Accuracy	Precision	Recall	F1-score	Time, c
Random Forest (Случайный лес)	**Classification**	**Supervised**	**0.99**	**1.0**	**0.75**	**0.83**	**6.6472**
			0.99	**1.0**	**0.89**	**0.99**	**4.7321**
Naive Bayes (Наивный Байесовский метод)	Classification	Supervised	0.31	0.51	0.64	0.25	0.725
			0.16	0.5	0.53	0.15	0.628
QDA (Квадратичный дискриминантный анализ)	Classification	Supervised	0.73	0.52	0.85	0.47	1.0098
			0.69	0.52	0.83	0.45	0.8507
ID3 – Iterative Dichotomiser 3 (Итеративный дихотомизатор-3)	Classification	Supervised	1.0	0.99	0.84	0.9	4.6815
			1.0	0.99	0.83	0.89	4.7226
KNN (*k*-ближайших соседей)	Classification	Unsupervised	0.99	0.91	0.83	0.87	38.5388
			1.0	0.97	0.89	0.93	31.9922
One-Class SVM (Одноклассовая машина опорных векторов)	Classification	Unsupervised	0.99	1.0	0.22	0.37	19.5771
			0.99	1.0	0.25	0.42	20.2327
Isolation Forest (Изолирующий лес)	**Proper Binary Tree**	**Unsupervised**	**0.99**	**0.85**	**0.78**	**0.81**	**4.9117**
			0.99	**0.93**	**0.81**	**0.83**	**3.9366**

Методы предварительного ранжирования повышают точность и скорость функционирования у контролируемых ML-моделей в среднем на 7%. У неконтролируемых ML-моделей предварительное ранжирование существенно не влияет на время обучения, но повышает F1-Score на 2-10%, что обосновывает его целесообразность,

Методы Naive Bayes и QDA демонстрируют очень высокую скорость обучения, но склонны к переобучению модели и существенно проигрывают по точности классификации: F1-score имеет значение 0,25 и 0,47 соответственно. KNN – простой и эффективный метод классификации, он не является оптимальным решением для агентных систем аутентификации в силу низких скоростей адаптации, при работе с большими объемами данных. Кроме того, при наличии выбросов в данных, KNN может давать ошибочные результаты. ID3 эффективен для наборов данных с большим количеством признаков, не требует масштабирование данных и имеет хорошую производительность (ID3 имеет линейную сложность, следовательно, время работы алгоритма прямо пропорционально количеству данных). Однако, ID3 не способен обрабатывать пропущенные значения, имеет ограниченную поддержку для данных с числовыми значениями и склонность к переобучению на некоторых типах данных.

Random Forest обеспечивает наивысшую *F1*-меру (*F1-Score* = 0,99) и отклик (*Recall*=0,89) при использовании предварительного ранжирования признаков.

В целом, Random Forest имеет ряд преимуществ по сравнению с другими ML-алгоритмами, включая высокую точность, хорошую устойчивость к шуму и выбросам, а также возможность обработки больших объемов данных. При трансляции кода модели с интерпретируемого в нативный *C* время работы метода, как правило, можно сократить в 3-5 раз (до 1 сек, в данном случае). Однако используемый принцип обучения (Supervised) ограничивает его применение в агентных системах аутентификации. Интерпретация результатов классификации у Random Forest сложнее, чем у Isolation Forest. Хотя One-Class SVM обеспечивают высокое значение Accuracy (0,99), его отклик (Recall) слишком низок (0,25) даже после предварительного ранжирования, что приводит к низкому показателю *F1-Score*. Isolation Forest обеспечивает наивысшую *F1*-меру (*F1-Score* 0,99) и отклик (Recall =0,89) при использовании предварительного ранжирования признаков, что отдает в его пользу предпочтение при выборе магистральной ML-модели классификации в агентных системах непрерывной аутентификации.

Заключение

Проведенные исследования показали, что методы предварительного ранжирования повышают точность и скорость функционирования у контролируемых ML-моделей в среднем на 7%. У неконтролируемых ML-моделей предварительное ранжирование существенно не влияет на время обучения, но повышает F1-Score на 2-10%, что обосновывает его целесообразность. В целом неконтролируемые подходы не столь точны, как контролируемые подходы, что актуализирует совершенствование объясняемых неконтролируемых подходов к выявлению аномалий, либо подходов на базе методов с подкреплением [20, 21].

На практике количество аномалий намного меньше, чем штатных случаев. В промышленных системах может появиться только 2-5 аномалий в год, но способных привести к нарушению функционирования критически важной инфраструктуры с серьезными последствиями. Таким образом, разработка подходов к обнаружению аномалий, которые не требует исторических аномальных случаев, остается важной и сложной проблемой. Кроме того, все современные подходы к обнаружению аномалий основаны на лог-последовательностях, что оказывает ограниченную помощь разработчикам в дальнейшей диагностике, такой как анализ первопричин.

Литература

1. *Беззатеев С.В., Фомичева С.Г., Супрун А.Ф.* Повышение эффективности мультиагентных систем информационной безопасности методами постквантовой криптографии // Проблемы информационной безопасности. Компьютерные системы. 2022. № 4 (52). С. 71-88. DOI: 10.48612/jisp/75dp-p3ed-hrmk

2. *Fomicheva, S. and Bezzateev, S.* Modification of the Berlekamp-Massey algorithm for explicable knowledge extraction by SIEM-agents // Journal of Physics: Conference Series, 2022, 2373(5), 052033. DOI:10.1088/1742-6596/2373/5/052033

3. *Mokaddem S., et al.* Taxonomy driven indicator scoring in MISP threat intelligence platforms // URL: https://arxiv.org/pdf/1902.03914.pdf (дата обращения – 22.04.2923)

4. *Shilin He and Jieming Zhu*, Loghub: A Large Collection of System Log Datasets towards Automated Log Analytics // URL: https://arxiv.org/pdf/2008.06448v1.pdf (дата обращения – 22.04.2923)

5. *Kazato, Y.; Nakagawa, Y.; Nakatani, Y.* Improving Maliciousness Estimation of Indicator of Compromise Using Graph Convolutional Networks. // In Proceedings of the 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 10–13 January 2020, pp. 1-7. DOI:10.1109/CCNC46108.2020.9045113

6. *Wu Y., Huang C., Zhang X., Zhou H.* GroupTracer: Automatic Attacker TTP Profile Extraction and Group Cluster in Internet of Things // *Secur. Commun. Netw*. 2020, URL: https://www.researchgate.net/deref/http%3A%2F%2Fdx.doi.org%2F10.1155%2F2020%2F8842539. DOI:10.1155/2020/8842539 (дата обращения – 22.04.2923)

7. *Ramsdale A., Shiaeles S., Kolokotronis N*. A Comparative Analysis of Cyber-Threat Intelligence Sources, Formats and Languages // Electronics 2020, 9 (5) :824. URL: https://www.researchgate.net/deref/http%3A%2F%2Fdx.doi.org%2F10.3390%2Felectronics9050824. DOI:10.3390/electronics9050824 (дата обращения – 22.04.2923)

8. *Gong, S., Lee, C.* Cyber Threat Intelligence Framework for Incident Response in an Energy Cloud Platform. // Electronics 2021, 10 (3) 239. URL https://www.researchgate.net/deref/http%3A%2F%2Fdx.doi.org%2F10.3390%2Felectronics10030239. DOI:10.3390/electronics10030239

9. *Kumar M., Hanumanthappa M., Kumar T.V.S.* Intrusion Detection System using decision tree algorithm // Proceedings of the 2012 IEEE 14th International Conference on Communication Technology, Chengdu,China, 9-11 November 2012, pp. 629-634. DOI:10.1109/ICCT.2012.6511281.

10. *Li Y., Xia J., Zhang S., Yan J., Ai X., Dai K*. An efficient intrusion detection system based onsupport vector machines and gradually feature removal method // Expert Syst. Appl.2012,39, pp. 424-430. DOI:10.1016/j.eswa.2011.07.032.

11. *Lin W.C., Ke S.W., Tsai C.F.* CANN: An intrusion detection system based on combining cluster centers and nearest neighbors // Knowl.-Based Syst. 2015,78, pp. 13-21. DOI:10.1016/j.knosys.2015.01.009.

12. *Aburomman A.A., Ibne Reaz M.B.* A novel SVM-kNN-PSO ensemble method for intrusion detectionsystem // Appl. Soft Comput. 2016, 38, Pp. 360–372. DOI:10.1016/j.asoc.2015.10.011.

13. *Yin C., Zhu Y., Fei J., He X.* A Deep Learning Approach for Intrusion Detection Using Recurrent NeuralNetworks // IEEE Access 2017, 5, pp. 21954-21961. DOI:10.1109/ACCESS.2017.2762418.

14. *Buczak A. L., Guven E*. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection // IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153-1176, Secondquarter 2016, DOI 10.1109/COMST.2015.2494502.

15. *Liu H., Lang B.* Machine learning and deep learning methods for intrusion detection systems: A survey // Applied Sciences 9(20):4396 DOI:10.3390/app9204396

16. *Shone N., et al.* Deep learning for anomaly detection: A review // ACM Comput. Surv., Vol. 1, No. 1, Article 1. 2020. https://arxiv.org/pdf/2007.02500.pdf (дата обращения – 22.04.2923)

17. *Chalapathy, R., and Sanjay, H.* Deep anomaly detection: A survey // In book: Machine Learning and Knowledge Discovery in Databases. 2017, pp. 36-51. DOI:10.1007/978-3-319-71249-9_3

18. *Javid K., et al* Compromise-free Bayesian neural networks // URL: https://arxiv.org/pdf/2004.12211.pdf (дата обращения – 22.04.2923).

19. *Фомичева С.Г., Конев А.В.* Адаптивная система управления содержания оксида кремния в шлаках при переработке медно-никелевых руд // Программные продукты и системы. 2014. № 3. С. 131-140. DOI:10.15827/0236-235X.107.131-140.

20. *Фомичева С.Г.* Теоретические аспекты квантования баз знаний в мультиагентных системах // Информационно-управляющие системы. 2017. № 3 (88). С. 2-10. DOI: 10.15217/issnl684-8853.2017.3.2

21. *Bezzateev S., Fomicheva S.* Soft multi-factor authentication // 2020 Wave Electronics and its Application in Information and Telecommunication Systems, WECONF 2020. 2020. С. 9131537. DOI: 10.1109/WECONF48837.2020.9131537

INFLUENCE OF ATTACK INDICATOR RANKING ON THE QUALITY OF MACHINE LEARNING MODELS IN AGENT-BASED CONTINUOUS AUTHENTICATION SYSTEMS

Svetlana G. Fomicheva, *Petersburg University of Aerospace Instrumentations, St-Petersburg, Russia,* ***levikha@mail.ru***

Abstract

Security agents of authentication systems function in automatic mode and control the behavior of subjects, analyzing their dynamics using both traditional (statistical) methods and methods based on machine learning. The expansion of the cybersecurity fabric paradigm actualizes the improvement of adaptive explicable methods and machine learning models. Purpose: the purpose of the study was to assess the impact of ranking methods at compromise indicators, attacks indicators and other futures on the quality of detecting network traffic anomalies as part of the security fabric with continuous authentication. Probabilistic and explicable methods of binary classification were used, as well as nonlinear regressors based on decision trees. The results of the study showed that the methods of preliminary ranking increase the F1-Score and functioning speed for supervised ML-models by an average of 7%. In unsupervised models, preliminary ranking does not significantly affect the training time, but increases the by 2-10%, which justifies their expediency in agent-based systems of continuous authentication. Practical relevance: the models developed in the work substantiate the feasibility of mechanisms for preliminary ranking of compromise and attacks indicators, creating patterns prototypes of attack indicators in automatic mode. In general, uncontrolled models are not as accurate as controlled ones, which actualizes the improvement of either explicable uncontrolled approaches to detecting anomalies, or approaches based on methods with reinforcement.

Keywords: *compromise indicator; attack indicator; ranking of indicators; methods of explained machine learning; decision trees.*

References

1. S.V. Bezzateev, S.G. Fomicheva, A.F Suprun (2022). Improving the efficiency of multi-agent information security systems using post-quantum cryptography. *Problemy informacionnoj bezopasnosti. Komp'juternye sistemy* [Problems of information security. Computer systems], no. 4 (52), pp. 71-88. DOI: 10.48612/jisp/75dp-p3ed-hrmk

2. S. Fomicheva, and S. Bezzateev (2022). Modification of the Berlekamp-Massey algorithm for explicable knowledge extraction by SIEM-agents. *Journal of Physics: Conference Series*, 2373(5), 052033. DOI:10.1088/1742-6596/2373/5/052033

3. S. Mokaddem, et al. (2023). Taxonomy driven indicator scoring in MISP threat intelligence platforms. URL: https://arxiv.org/pdf/1902.03914.pdf (date of access – 22.04.2023)

4. Shilin He and Jieming Zhu (2023). Loghub: A Large Collection of System Log Datasets towards Automated Log Analytics. URL: https://arxiv.org/pdf/2008.06448v1.pdf (date of access - 22.04.2023)

5. Y. Kazato, Y. Nakagawa, Y. Nakatani (2020). Improving Maliciousness Estimation of Indicator of Compromise Using Graph Convolutional Networks. *Proceedings of the 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, 10-13 January 2020, pp. 1-7.13. DOI:10.1109/CCNC46108.2020.9045113

6. Y. Wu, C. Huang, X. Zhang, H. Zhou (2020). GroupTracer: Automatic Attacker TTP Pro?le Extraction and Group Cluster in Internet of Things. *Secur. Commun. Netw.* URL: https://www.researchgate.net/deref/http%3A%2F%2Fdx.doi.org%2F10.1155%2F2020%2F8842539. DOI:10.1155/2020/8842539 (date of access – 22.04.2023)

7. A. Ramsdale, S. Shiaeles, N. Kolokotronis (2023). A Comparative Analysis of Cyber-Threat Intelligence Sources, /Formats and Languages. Electronics 2020, 9 (5) :824. URL: https://www.researchgate.net/deref/http%3A%2F%2Fdx.doi.org%2F10.3390%2Felectronics9050824. DOI:10.3390/electronics9050824 (date of access – 22.04.2023)

8. S. Gong, C. Lee (2021). Cyber Threat Intelligence Framework for Incident Response in an Energy Cloud Platform. /Electronics 2021, 10 (3) 239. URL https://www.researchgate.net/deref/http%3A%2F%2Fdx.doi.org%2F10.3390%2Felectronics10030239 DOI:10.3390/electronics10030239 (date of access – 22.04.2023)

9. M. Kumar, M. Hanumanthappa, T.V.S. Kumar (2012). Intrusion Detection System using decision tree algorithm. *Proceedings of the 2012 IEEE 14th International Conference on Communication Technology*, Chengdu, China, 9-11 November 2012, pp. 629-634. DOI:10.1109/ICCT.2012.6511281.

10. Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, K. Dai (2012). An eficient intrusion detection system based onsupport vector machines and gradually feature removal method. Expert Syst. Appl. 2012, no. 39, pp. 424-430. DOI:10.1016/j.eswa.2011.07.032.

11. W.C. Lin, S.W. Ke, C.F. Tsai (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowl.-Based Syst.*, no. 78, pp. 13-21. DOI:10.1016/j.knosys.2015.01.009.

12. A.A. Aburomman, M.B. Ibne Reaz (2016). A novel SVM-kNN-PSO ensemble method for intrusion detectionsystem. *Appl. Soft Comput.*, no. 38, pp. 360-372. DOI:10.1016/j.asoc.2015.10.011.

13. C. Yin, Y. Zhu, J. Fei, X. He (2017). A Deep Learning Approach for Intrusion Detection Using Recurrent NeuralNetworks. *IEEE Access*, no. 5, pp. 21954-21961. DOI:10.1109/ACCESS.2017.2762418.

14. A.L. Buczak, and E. Guven (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection, *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153-1176, Secondquarter 2016, DOI 10.1109/COMST.2015.2494502.

15. H. Liu, B. Lang. Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences,* no. 9(20), pp. 4396 DOI:10.3390/app9204396

16. N. Shone, et al. (2020). Deep learning for anomaly detection: A review. *ACM Comput. Surv.*, vol. 1, No. 1, Article 1, https://arxiv.org/pdf/2007.02500.pdf (date of access - 22.04.2023)

17. Chalapathy, R., and Sanjay, H. Deep anomaly detection: A survey./In book: Machine Learning and Knowledge Discovery in Databases^ 2017. pp. 36-51. DOI:10.1007/978-3-319-71249-9_3

18. K. Javid, et al. (2023). Compromise-free Bayesian neural networks. URL: https://arxiv.org/pdf/2004.12211.pdf (date of access – 22.04.2023)

19.S.G. Fomicheva, A.V. Konev (2014). Adaptive control system for silicon oxide concentration in slags at processing copper-nickel ores. *Programmnye produkty i sistemy* [SOFTWARE & SYSTEMS], no. 3, pp. 131-140. DOI:10.15827/0236-235X.107.131-140

20. S.G. Fomicheva (2017). Theoretical aspects of knowledge base quantization in multi-agent systems. *Informacionno-upravljajushhie sistemy* [Information and control systems], no. 3 (88), pp. 2-10. DOI: 10.15217/issnl684-8853.2017.3.2

21. S. Bezzateev, S. Fomicheva (2020). Soft multi-factor authentication. *2020 Wave Electronics and its Application in Information and Telecommunication Systems, WECONF 2020*, pp. 9131537. DOI: 10.1109/WECONF48837.2020.9131537

Information about author:

Svetlana G. Fomicheva, *PhD, Full Professor, Professor at the Department of Information Security, St. Petersburg University of Aerospace Instrumentations, St. Petersburg, Russia*