Научная статья
УДК 519.872
https://doi.org/10.31854/1813-324X-2025-11-4-107-117
EDN:OELOOW

Математическая модель сигнатурного анализа сетевого трафика и экспериментальная оценка эффективности ее функционирования

Александр Александрович Браницкий[1] ✉, branickii1.aa@sut.ru
Наталия Алексеевна Браницкая[2], nataliya_petrova@mail.ru

[1]Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, Санкт-Петербург, 193232, Российская Федерация
[2]Санкт-Петербургский государственный университет, Санкт-Петербург, 199034, Российская Федерация

Аннотация

Актуальность. *В современных системах обнаружения атак алгоритмы сигнатурного анализа являются ключевыми составляющими в процессе исследования сетевого трафика. Их широкая распространенность при реализации правил обнаружения атак обусловлена не только простотой их настройки и быстротой поиска, но также и возможностью обнаружения атак с нулевым уровнем ложных срабатываний. Это достигается за счет задания таких наборов специальных правил (сигнатурных правил), которые однозначно идентифицируют конкретный тип атаки. Разработка и оптимизация моделей и алгоритмов построения таких правил является актуальной задачей, поскольку ее решение позволяет увеличить уровень защищенности сетевых ресурсов от атак.*

Цель исследования *заключается в повышении эффективности функционирования систем обнаружения атак, построенных на основе сигнатурного анализа.*

Используемые методы. *Исследование базируется на применении положений из теорий множеств и поиска информации, а также приемов параллельного и сетевого программирования. Предмет исследования – модели и алгоритмы сигнатурного поиска атак в сетевом трафике, объект – сигнатурные системы обнаружения атак.*

Новизна. *В статье представлена математическая модель сигнатурного анализа сетевого трафика, которая отличается от известных аналогов универсальностью представления сигнатурных правил и поддержкой многоуровневой обработки как отдельных пакетов, так и сетевых потоков данных; выполнена оценка эффективности программной реализации данной модели. Универсальность представления сигнатурных правил достигается за счет возможности их расширения новыми правилами независимо от их внутренней реализации и без необходимости перестроения исходной модели. Многоуровневая обработка пакетов и сетевых потоков данных сигнатурными правилами обеспечивается за счет разработанных и встроенных в модель алгоритмов IP-дефрагментации и TCP-реассемблирования.*

Практическая значимость. *Результат проведенного эксперимента показывает, что разработанный анализатор сетевого трафика демонстрирует производительность, в 1,5 раза превосходящую в терминах оперативности и ресурсопотребления другие системы обнаружения атак с открытым исходным кодом. Тем самым разработанная модель может быть использована при построении эффективной системы обнаружения атак.*

Ключевые слова: *сигнатурный анализ, сигнатурные правила, сетевой трафик, сетевые пакеты, сетевые атаки, системы обнаружения атак, компьютерные сети, конечные автоматы, сети Петри, поиск подстрок, регулярные выражения*

Ссылка для цитирования: Браницкий А.А., Браницкая Н.А. Математическая модель сигнатурного анализа сетевого трафика и экспериментальная оценка эффективности ее функционирования // Труды учебных заведений связи. 2025. Т. 11. № 4. С. 107–117. DOI:10.31854/1813-324X-2025-11-4-107-117. EDN:OELOOW

Original research
https://doi.org/10.31854/1813-324X-2025-11-4-107-117
EDN:OELOOW

Signature Analysis Mathematical Model of Network Traffic and Experimental Evaluation of Its Functioning Efficiency

Alexander A. Branitskiy[1] ✉, branickii1.aa@sut.ru
Natalia A. Branitskaya[2], nataliya_petrova@mail.ru

[1]The Bonch-Bruevich Saint Petersburg State University of Telecommunications,
St. Petersburg, 193232, Russian Federation
[2]The Saint Petersburg State University,
St. Petersburg, 199034, Russian Federation

Annotation

Relevance. *In modern attack detection systems (ADSs), signature analysis algorithms are key components in the process of analyzing network traffic. Their widespread use in implementation of attack detection rules is due not only to the ease of their configuration and search speed, but also to the ability to detect attacks with zero false positives. This is achieved by specifying such sets of special rules (signature rules) that uniquely identify a specific type of attack. The development and optimization of models and algorithms for constructing such rules is an urgent task, since its solution allows increasing the level of protection of network resources from attacks.*
The purpose of the research *is to improve the operating efficiency of ADSs built on the basis of signature analysis.*
Methods used. *The research is based on the application of provisions from the theories of sets and information retrieval, as well as parallel and network programming techniques. The subject is models and algorithms for signature search of attacks in network traffic, the object is signature ADSs.*
Novelty. *The paper presents a mathematical model of signature analysis of network traffic, which differs from known analogs in the universality of the representation of signature rules and support for multi-level processing of both individual packets and network data flows; an assessment of the effectiveness of the software implementation of this model is performed. The universality of the representation of signature rules is achieved due to the possibility of their expansion with new rules regardless of their internal implementation and without the need to reconstruct the original model. Multi-level processing of packets and network data flows by signature rules is ensured by the IP defragmentation and TCP reassembly algorithms developed and integrated into the model.*
Practical significance. *The result of the experiment shows that the developed network traffic analyzer demonstrates performance that is 1,5 times superior in terms of promptness and resource consumption to other open source ADSs. Thus, the developed model can be used in constructing an effective ADS.*

Keywords: *signature analysis, signature rules, network traffic, network packets, network attacks, attack detection systems, computer networks, finite state machines, Petri nets, substring search, regular expressions*

For citation: Branitskiy A.A., Branitskaya N.A. Signature Analysis Mathematical Model of Network Traffic and Experimental Evaluation of Its Functioning Efficiency. *Proceedings of Telecommunication Universities.* 2025; 11(4):107–117. (in Russ.) DOI:10.31854/1813-324X-2025-11-4-107-117. EDN:OELOOW

Введение

Под сигнатурным анализом понимается набор моделей и алгоритмов обнаружения атак, основанных на построении экспертных правил, сопоставлении с образцом, использовании конечных автоматов. Данный механизм является одним из наиболее распространенных подходов при реализации систем, предназначенных для обнаружения злоупотреблений в компьютерных сетях [1]. Популярность такого решения объясняется тем, что с помощью задания сигнатур, представляющих собой специальные ключевые слова или продукци-

онные правила, становится возможным описывать явным образом классы обнаруживаемых атак. Кроме того, системы обнаружения атак (СОА), построенные на основе сигнатурного анализа, характеризуются высокой скоростью функционирования и отсутствием ложных срабатываний (ошибок первого рода). Использование таких систем позволяет сократить объем ручного труда, выполняемого аналитиками безопасности при поиске возможных нарушений в сетевых и хостовых журналах регистрации событий, ввиду понятной человеку интерпретируемости полученных результатов. Во многом благодаря именно этим свойствам и коммерческие, и открытые СОА имеют в основе своего функционирования механизмы сигнатурного поиска вредоносной деятельности, которые включают как (I) алгоритмы поиска подстрок и регулярные выражения, так и (II) более сложные управляющие последовательности (например, системные команды, написанные на некотором языке программирования).

В случае I поиск сигнатур является примитивной операцией с точки зрения задания таких сигнатур администратором безопасности. От разработчиков подобных правил требуется лишь выбрать алгоритм, который является наименее ресурсоемким и обладает наименьшей временно́й сложностью [2]. К СОА, построенным с использованием этого подхода, можно отнести СОА Snort, в которой сигнатуры представляются как набор атрибутов и их значений.

В случае II могут быть заданы некоторые формальные выражения, например продукционные правила, или математический аппарат, например конечный автомат, сеть Петри и контекстно-свободные грамматики (КС-грамматики). Среди СОА этого типа можно перечислить Bro и Suricata (с поддержкой опции luajit).

Построение перспективных сигнатурных СОА – актуальное направление в области информационной безопасности, развитие которого позволит поддерживать требуемый уровень защищенности сетевых ресурсов и минимизировать риски, связанные с нарушением конфиденциальности, целостности и доступности критически важной информации. Тем самым обеспечение безопасности сетевых ресурсов должно неразрывно сопровождаться совершенствованием существующих подходов и поиском новых решений для анализа сетевого трафика.

Разработанная в статье математическая модель является многосоставной, описывает процессы сборки пакетов в сетевые потоки данных и содержит представление сигнатур. Она характеризуется наличием алгоритмов восстановления сетевых потоков данных, что позволяет анализировать содержимое, передаваемое в нескольких пакетах, а также возможностью описания сигнатурных правил без привязки к их реализации.

Классификация методов сигнатурного анализа сетевого трафика

На рисунке 1 представлена классификация методов сигнатурного анализа сетевого трафика.

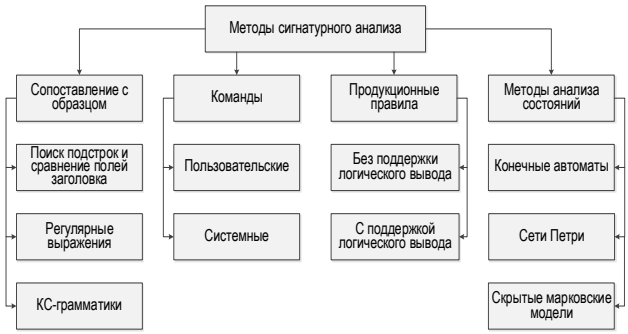

Рис. 1. Классификация методов сигнатурного анализа сетевого трафика

Fig. 1. Classification of Signature Analysis Methods of Network Traffic

Сигнатурный анализ включает в свой состав широкий класс методов, таких, например, как сопоставление с образцом, методы анализа состояний, команды (функции с аргументами), продукционные правила.

Сигнатурное правило R, построенное на основе сопоставления с образцом, представляется следующим образом: $R(b) = G_0(b) \wedge \ldots \wedge G_k(b)$, где предикаты $G_0, \ldots, G_k$ задают некоторое булево соответствие между заранее заданными ключевыми словами $\{w_0, \ldots, w_k\}$ и значениями атрибутов $\{f_0, \ldots, f_k\}$, полученных в результате парсинга (parse) пакета $b \in B^+$, где B^+ – всевозможные байтовые цепочки:

$$B^+ = \{0, 1, \ldots, N, 00, 01, \ldots, 0N, 10, 11, \ldots, 1N, \ldots\}, N = 255;$$

$f_i = \text{parse}(b,i)$; результат применения предиката $G_i(b)$ – это проверка наличия поля f_i в структуре пакета b и последующее сопоставление его значения с элементами w_i.

В случае поиска подстрок и сравнения полей заголовка каждое из ключевых слов w_i представляется как множество возможных значений атрибута f_i. Так, предикат G_0 выполняет проверку вхождения (issub) каждого из элементов w_0 внутрь содержимого f_0 анализируемого пакета b: $\forall e \in w_0$ issub(e,f_0), а предикат G_i ($i = 1, \ldots, k$) выполняет проверку принадлежности значения атрибута f_i заданному набору значений: $f_i \in w_i$. В качестве примера рассмотрим правило СОА Snort с идентификатором sid = 1762 (рисунок 2а), указанное в файле /etc/snort/rules/server-webapp.rules. В данном случае выполняется поиск подстрок «/phf» и «QALIAS» (т. е. вычисляется предикат $\forall e \in$ {«/phf», «QALIAS»} issub(e, f_0)), которые указываются в качестве значе-

ния опции content, внутри содержимого TCP-пакетов, отправляемых на HTTP-порт ($f_i \in \{80, 8080, \ldots\}$).

Регулярные выражения, как и поиск подстрок, являются распространенным способом обнаружения сетевых атак, причем для их создания не требуется каких-либо узкоспециализированных знаний. В этом случае ключевое слово w_i представляется как набор искомых подстрок и метасимволов, определяющих позицию и состав этих подстрок внутри анализируемого значения атрибута f_i, а предикат G_i проверяет f_i на соответствие регулярному выражению w_i. На рисунке 2b представлен пример сигнатуры, построенной на основе регулярных выражений, для COA Snort.

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"SERVER-WEBAPP phf arbitrary command execution attem
pt"; flow:to_server,established; content:"/phf"; fast_pattern; nocase; http_uri; content:"QALIAS"; nocase; content
:"%0a"; nocase; metadata:policy max-detect-ips drop, ruleset community, service http; reference:bugtraq,629; refer
ence:cve,1999-0067; classtype:web-application-attack; sid:1762; rev:25;)
```

a)

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"SQL union select - possible sql injection attempt -
 GET parameter"; flow:to_server,established; content:"union"; fast_pattern; nocase; http_uri; content:"select"; no
case; http_uri; pcre:"/union\s+(all\s+)?select\s+/Ui"; metadata:policy max-detect-ips drop, policy security-ips dr
op, service http; reference:bugtraq,21227; reference:cve,2006-2835; reference:cve,2006-6268;-reference:cve,2007-10
21; reference:cve,2007-2824; reference:cve,2011-1667; reference:url,www.securityfocus.com/archive/1/452259; classt
ype:misc-attack; sid:13990; rev:22;)
```

b)

Рис. 2. Сигнатурное правило COA Snort с идентификатором: a) sid = 1762; b) sid = 13990

Fig. 2. ADS Snort Signature Rule with sid = 1762 (a) and sid = 13990 (b)

В данном сигнатурном правиле предполагается, что злоумышленник пытается выполнить атаку SQL-инъекции. Планируемый им к исполнению SQL-запрос передается в качестве GET-аргумента Web-серверу, взаимодействующему с SQL-базой данных. Признаком атаки считается положительное срабатывание регулярного выражения:

«/union\s+(all\s+)?select\s+/Ui»

над входным параметром, представленным как содержимое TCP-пакета, адресованного Web-серверу. В отличие от поиска подстрок, регулярные выражения направлены на охват существенно более широкого диапазона вариаций одной и той же атаки. В частности, за счет наличия модификатора *i*, указанного после символа прямого слеша, выполняется регистронезависимый поиск. Кроме того, такие последовательности, как «*A*\s+*B*» и «*C*?», позволяют игнорировать вставку незначащих пробелов между SQL-операторами, представленными как *A* и *B*, и задавать число непрерывных вхождений строки *C* как 0 или 1. Указанные особенности регулярных выражений позволяют покрывать сразу несколько SQL-команд, синтаксически реализованных по-разному, но отличий между которыми нет с точки зрения их конечного результата исполнения.

КС-грамматики являются надстройкой над классом регулярных выражений, что позволяет при помощи первых описывать рекурсивно вложенные друг в друга последовательности. В частности, использование такого механизма позволяет, как и в случаях поиска подстрок и регулярных выражений, проверять вхождения определенных символов, а также дополнительно исследовать окружение этих символов (например, проверять корректность парных скобок, обрамляющих эти символы).

В рассматриваемом случае ключевые слова $\{w_0, \ldots, w_k\}$ представляются как последовательность правил вывода, определенных КС-грамматикой, а предикаты $\{G_0, \ldots, G_k\}$ осуществляют проверку того, порождаются ли значения атрибутов $\{f_0, \ldots, f_k\}$, извлеченных из сетевых пакетов, системой правил $\{w_0, \ldots, w_k\}$. Например, в [3] для обнаружения каждой *j*-й сетевой атаки и описания шаблона нормального трафика приводится набор правил КС-грамматики Γ_j. Предварительно вычисленные значения атрибутов $\{f_0, \ldots, f_k\}$ сетевого соединения считаются принадлежащими *j*-му классу сетевой атаки, если последовательность $\{f_0, \ldots, f_k\}$ является одной из терминальных цепочек языка, порождаемого грамматикой Γ_j.

С увеличением числа сигнатурных правил снижается производительность систем, построенных на их основе. Сокращение времени поиска сигнатурного правила и исключение дублирующих или противоречивых правил может быть достигнуто за счет перехода к древовидной модели представления сигнатурных правил или к методам анализа конечных состояний. Так, предикаты G_i, повторяющиеся в разных правилах, обозначаются единой последовательностью ребер от корневой вершины строящегося дерева правил. Остальные предикаты создают уникальный путь до терминального листа, достижение которого рассматривается как положительное срабатывание соответствующего правила [4, 5].

В модели, построенной на основе анализа конечных состояний, становится возможным описывать многоуровневые атаки. За счет представления сигнатурного правила *R* в виде набора пар {((cur_state, trans_cond) → new_state)} сохраняется история изменения значений наблюдаемых атрибутов $\{f_0, \ldots, f_k\}$. При успешном выполнении усло-

вия trans_cond конечный автомат осуществляет переход в следующее состояние new_state, зависящее от его текущего состояния cur_state и значения функции смены состояния. В качестве систем, в которых был реализован подобный подход, стоит назвать систему USTAT [6], предназначенную для обнаружения атак на UNIX-хосты.

Другим примером является COA IDIOT [1, 7], в которой раскрашенные сети Петри применяются в качестве модели для обнаружения злоупотреблений в компьютерной сети. Последовательность действий злоумышленника представляется в виде ориентированного графа состояний, конечная вершина в котором представляет собой цель атакующего. Предлагаемая в [1, 7] раскрашенная сеть Петри характеризуется наличием не более одного направленного ребра между состоянием (узлом сети) и дальнейшим переходом. В ней присутствует ровно одно конечное состояние, при этом на количество начальных состояний такого ограничения нет.

Переход в новое состояние выполняется, если одновременно удовлетворяются оба следующих условия. Во-первых, логическое условие, представляющее собой результат определенного пользователем предиката, принимает истинное значение. Во-вторых, во всех состояниях, являющихся входными для данного перехода, размещается хотя бы один маркер. Далее при удовлетворении этих двух условий выполняется унификация формальных переменных, указанных в маркере, с реальными значениями атрибутов зафиксированного события. Указанным образом осуществляется движение маркера по сети Петри. Атака детектируется при перемещении маркера в терминальный узел. Определение вероятности перехода системы в скомпрометированное атакой состояние возможно при помощи другой модели, а именно скрытой марковской модели [8].

В случае команд значение атрибута f_i рассматривается как выходной результат исполнения некоторой пользовательской или системной функции. Это значение в дальнейшем используется для сравнения с заранее заданной или динамически вычисляемой пороговой величиной w_i. Здесь становится возможным аккумулировать характерные для сетевых соединений статистические показатели, например, частоту появления пакетов с флагами, указывающими на признак сканирования IP-хостов или портов, или количество открытых TCP-соединений на определенном порту сервера. Для добавления ответных реакций act1 и / или act2 на случай срабатывания и / или пропуска сигнатурного правила $R(b)$ применяются экспертные системы, например, IDES [9] и EMERALD [10, 11], с продукционными правилами вида IF ⟨$R(b)$⟩ THEN ⟨act1⟩ [ELSE ⟨act2⟩]. Такие системы характеризуются наличием базы знаний с механизмом обработки ее содержимого на основе методов вывода, например, прямого логического вывода и правила modus ponens [10–12], байесовского вывода [13] или нечеткого вывода [14].

Математическая модель сигнатурного анализа сетевого трафика

Математическая модель сигнатурного анализа сетевого трафика описывается в виде выражения (1) и состоит из модели восстановления сетевых потоков данных (2) и моделей представления сигнатур (3). Рассмотрим эти модели более детально.

Разработанная модель M_1 сигнатурного анализа сетевого трафика представляется как кортеж следующего вида:

$$M_1 = \langle E, V, H, M_2, \{M_3\}\rangle, \quad (1)$$

где E – множество направленных дуг, представляющих собой связи между сетевыми узлами; V – множество вершин, представляющих собой узлы внутри компьютерной сети H: $E \to V \times V$ – отображение, сопоставляющее каждой дуге упорядоченную пару вершин и приписывающее наличие односторонней связи между соответствующими сетевыми узлами; M_2 – модель восстановления сетевых потоков данных; $\{M_3\}$ – модели представления сигнатур.

Запишем модель M_2 восстановления сетевых потоков данных в виде следующего кортежа:

$$M_2 = \langle P, G, C, T\rangle, \quad (2)$$

где $P = B^{[I:J]} \subset B^+$ – сетевые пакеты; $B^{[I:J]}$ – байтовые цепочки длины не меньше I и не больше J; G – функция извлечения типа протокола сетевого пакета: $G : P \to \{\text{IP, TCP, UDP, ICMP}\}$; C – предикат проверки корректности заголовка и содержимого сетевого пакета (проверки контрольных сумм IP-, TCP-, UDP- и ICMP-пакетов): $C : B^+ \to \{0, 1\}$, T – отображение, выполняющее склейку содержимого нескольких IP-фрагментов и реассемблирование нескольких TCP-сегментов: $T : 2^P \to B^+$.

Функционирование модели M_2 описывается следующим образом. Каждый сетевой пакет представляет собой байтовую цепочку, длина которой не меньше I и не превосходит J. Для сетевого протокола IP эти величины принимают следующие значения: $I_{IP} = 20$; $J_{IP} = 65535$. На практике значение J_{IP} сверху ограничено порогом максимальной единицы передачи (MTU, *аббр. от англ.* Maximum Transmission Unit), которая, в свою очередь, чаще всего установлена в значение 1500 байт (для ОС Linux и Windows). Для транспортного протокола TCP эти величины равны следующим значениям: $I_{TCP} = 20$ (наименьший размер TCP-заголовка); $J_{TCP} = J_{IP} - I_{IP} = 65515$. Для транспортного протокола

UDP – $I_{UDP} = 8$ (размер UDP-заголовка); $J_{UDP} = J_{IP} –$ $– I_{IP} = 65515$. Для сетевого протокола ICMP – $I_{ICMP} = 8$ (размер ICMP-заголовка); $J_{ICMP} = J_{IP} – I_{IP} = 65515$. Перед выполнением процедуры склейки, свойственной для IP- и TCP-пакетов, выполняются функции определения типа протокола G и проверки контрольных сумм C.

Рассмотрим функцию G извлечения сетевого протокола. Для проверки того, что сырой пакет (т. е. байтовый поток данных, захваченных на сетевом интерфейсе и еще не обработанных сетевым стеком ОС) является действительно IP-пакетом, выполняется сдвиг на 12 байт, занятых под размещение MAC-адресов получателя и отправителя. Следующие за ними 2 байта предназначены для идентификации типа протокола нижележащего уровня, инкапсулированного в Ethernet-кадр. Если извлеченная двухбайтовая последовательность равна 2048 (0×0800), то пакет считается принадлежащим уровню IPv4, если 34525 (0×86DD) – IPv6. Для определения типа протокола пакета, вложенного в IP-фрагмент, выполняется сдвиг на 9 байт, и извлекается следующий за ними 10-й байт, равенство которого 1 означает принадлежность к ICMP, 6 – TCP, 17 – UDP.

Функционирование предиката C оформлено в виде алгоритма, выполняющего проверку контрольной суммы сетевых пакетов на уровне протокола IP согласно спецификации RFC 1071 [15]. Разработанный алгоритм был расширен таким образом, чтобы осуществлялась проверка корректности также TCP-, UDP- и ICMP-пакетов посредством введения структуры псевдозаголовка [16].

Для описания отображения T рассмотрим алгоритмы IP-дефрагментации и TCP-реассемблирования, представленные на рисунках 3 и 4. В первом алгоритме выполняется объединение IP-сегментов, фрагментированных на уровне IP, в единый IP-пакет. Во втором алгоритме выполняется сборка содержимого, передаваемого в нескольких TCP-пакетах, в единую TCP-сессию.

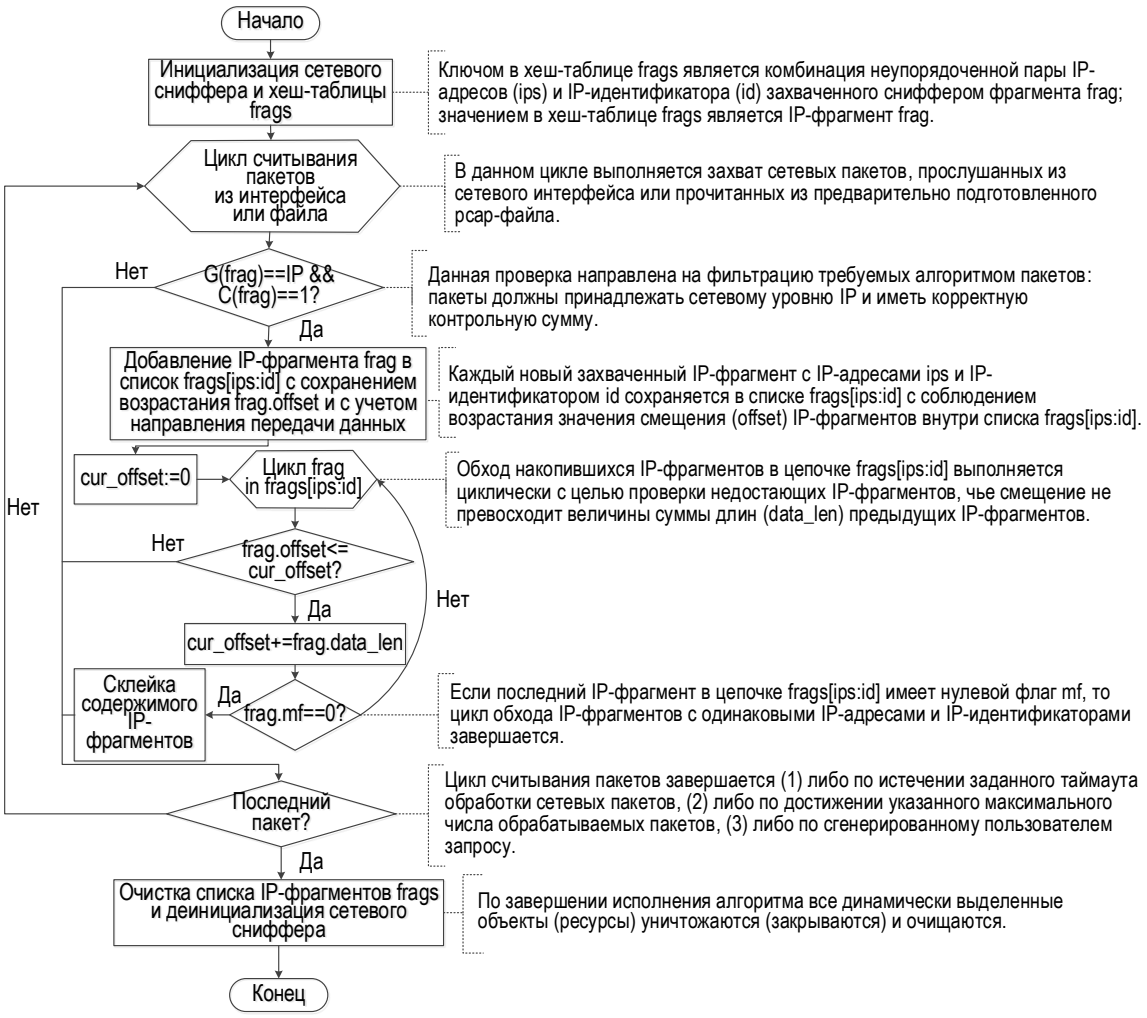

Рис. 3. Алгоритм дефрагментации IP-пакетов

Fig. 3. IP-Packet Defragmentation Algorithm

В алгоритме, блок-схема которого изображена на рисунке 3, для каждого IP-пакета осуществляется проверка его контрольной суммы. В случае ее некорректности такой пакет отбрасывается. В противном случае пакет добавляется в список, элементы которого состоят из IP-фрагментов, имеющих одинаковые пары IP-адресов и IP-идентификаторы (двухбайтовое поле identification в IP-заголовке). Во вложенном цикле выполняется обход накопившихся IP-фрагментов в порядке возрастания значения их поля смещения (fragment offset). Если присутствует недостающий IP-фрагмент, т. е. величина смещения очередного IP-фрагмента больше суммы величины смещения предыдущего и его длины, то сборка таких фрагментов прерывается. В противном случае выполняется склейка IP-фрагментов в единый IP-пакет при условии, что в последнем фрагменте бит MF (more_fragment) установлен в нулевое значение.

При сборке TCP-пакетов в сессии (рисунок 4) формируется список, элементы которого идентифицируются при помощи двух сокетных пар (IP-адреса и порта источника, IP-адреса и порта назначения). Обход такого списка выполняется в порядке возрастания позиционных номеров (sequence numbers). Объединение данных, накопившихся в нескольких TCP-пакетах, выполняется в одном из следующих случаев, когда:

– инициируется сетевое соединение, т. е. в момент захвата на сетевом интерфейсе пакета с установленным флагом SYN;

– проталкиваются данные в буфер пользовательских приложений, т. е. при получении пакета с установленным флагом PSH;

– терминируется сетевое соединение, т. е. в момент захвата на сетевом интерфейсе пакета с установленным флагом FIN или RST.

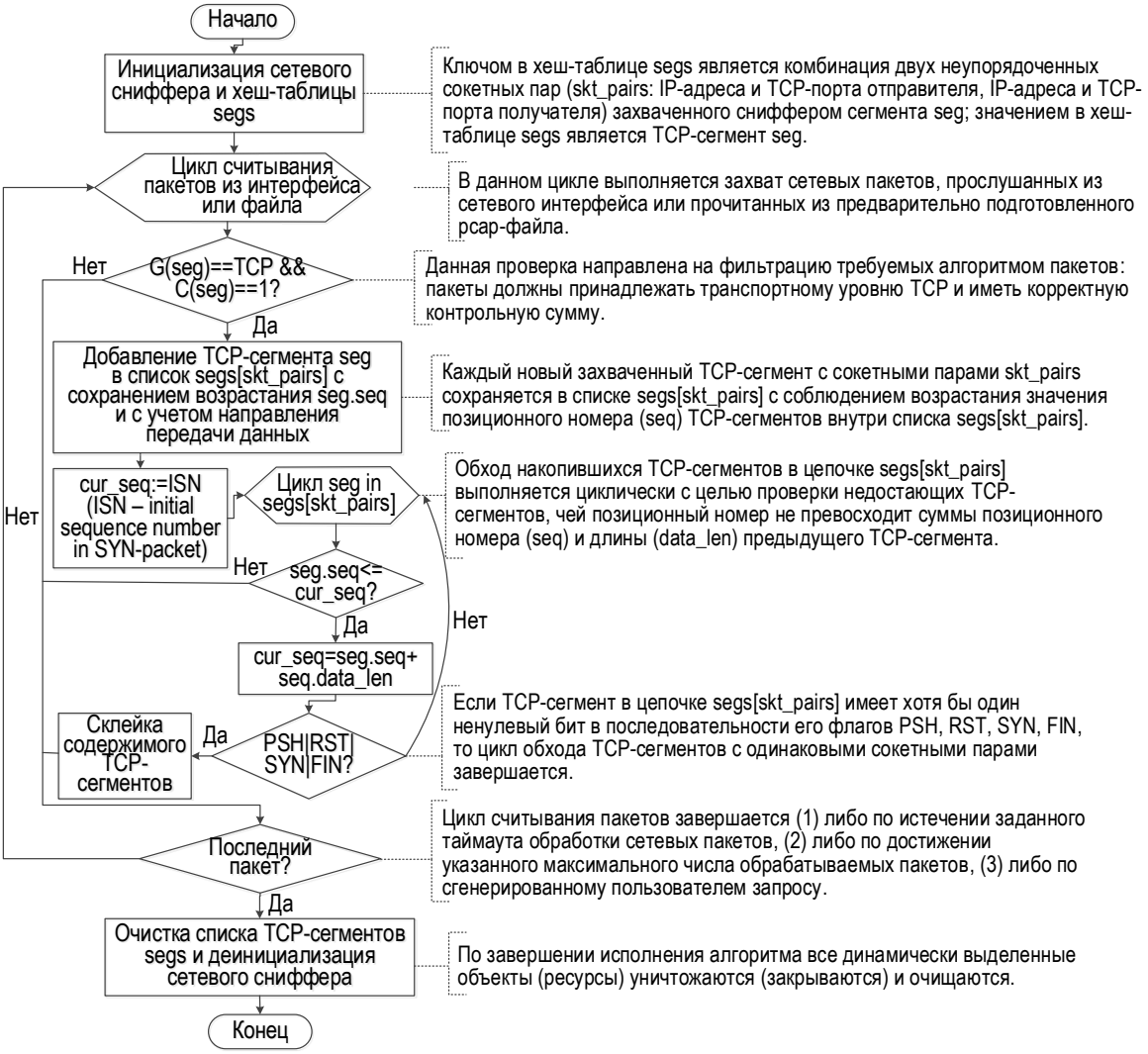

Рис. 4. Алгоритм реассемблирования данных, содержащихся в TCP-пакетах

Fig. 4. Algorithm for Reassembling Data Contained in TCP-Packets

После выполнения процедуры склейки при помощи отображения T анализируется контент, передаваемый в нескольких сетевых пакетах.

Для этой цели применяется модель M_3 представления сигнатур, которая описывается следующим образом:

$$M_3 = \langle F, R, A \rangle, \quad (3)$$

где F: $B^+ \to \{0, 1\}$ – булево правило фильтрации сетевых пакетов, выполняемое перед применением сигнатурного правила; R: $B^+ \to \{0, 1\}$ – сигнатурное правило; $A = \{a_1, a_2, ..., a_n\}$ – действие, выполняемое при положительном срабатывании сигнатурного правила (т. е. если $F(b) = 1 \wedge R(b) = 1$, где $b \in B^+$ – обрабатываемый поток данных).

Правило фильтрации применяется к полям отдельных сетевых пакетов и выполняется перед IP-дефрагментацией и TCP-реассемблированием. Это позволяет исключить из рассмотрения пакеты с неправильной контрольной суммой с противоречащими друг другу флагами в заголовках пакетов, а также с неизвестным типом протокола, инкапсулированного в IP-фрагмент. Сигнатурное правило представляет собой отображение, выполняющее поиск характерных признаков, однозначно указывающих на наличие вредоносности, внутри отдельных пакетов и сетевых потоков данных $b \in B^+$, таких что $F(b) = 1$.

Особенность данной модели заключается в универсальности представления сигнатурных правил, что позволяет без изменения структуры и содержания самой модели обогащать ее функционирование посредством добавления новых правил независимо от их конкретной реализации. Так, для эффективной реализации сигнатурных правил R могут использоваться конечные автоматы, расширенные посредством введения проактивного механизма определения первичных признаков проведения многошаговой атаки [17], сети Петри, дополненные нечеткими продукционными правилами [18], алгоритмы параллельного поиска подстрок, реализованные на графических процессорах [19], регулярные выражения, вычисляемые с поддержкой аппаратного ускорения [20].

Действие A включает в себя следующие шаги:

– запись в журнал регистрации системных событий (a_1);

– генерация уведомления через его отправку на электронную почту (a_2);

– игнорирование события ввиду его низкой приоритетности (a_3);

– добавление нового правила межсетевого экрана (a_4);

– принудительное терминирование сетевого соединения (a_5);

– вызов другого (уточняющего) сигнатурного правила (a_6).

Эксперимент

В рамках проведенного эксперимента было выполнено сравнение разработанного сетевого анализатора (Sensor), в котором реализована представленная математическая модель сигнатурного анализа сетевого трафика, с такими СОА, как Snort 2.9.8.2, Suricata 3.0.1 и Bro 2.4.1, при этом СОА Suricata запускалась в двух режимах: однопоточном (Suricata^S) и многопоточном (Suricata^A). В таблице 1 приведены показатели ресурсопотребления и затрат по времени функционирования рассматриваемых СОА.

ТАБЛИЦА 1. Показатели ресурсопотребления и затрат по времени функционирования СОА

TABLE 1. Resource Consumption and Operating Time Indicators of ADSs

	$T^{(real)}$	$T^{(proc)}$	$V^{(kpkts)}$	$V^{(data)}$	$L^{(CPU)}$	$C^{(virt)}$	$C^{(resd)}$	$C^{(real)}$
Sensor	30,7	30,1	324,8	18949,1	83,6	233,4	201,6	198,5
Snort	65,7	51,3	190,2	11098,8	88,8	705,7	422,2	416,9
Suricata^S	53,1	46,6	209,5	12223,1	141,9	617,4	303,8	296,7
Suricata^A	102,7	96,2	101,5	5923,7	193,3	914,1	318,7	311,6
Bro	184,7	160,8	60,7	3543,6	116,0	972,2	634,4	625,4

При вычислении указанных показателей использовался pcap-файл объемом 709 МБ, содержащий ровно 10 млн сетевых пакетов.

Среди вычисляемых показателей были выбраны следующие восемь критериев:

1) $T^{(real)}$ – общее время функционирования (сек.);

2) $T^{(proc)}$ – время обработки пакетов (общее время функционирования без учета времени инициализации и освобождения ресурсов сниффера), измеряемое в сек.;

3) $V^{(kpkts)}$ – количество обрабатываемых килопакетов за единицу времени (сек.$^{-1}$);

4) $V^{(data)}$ – количество обрабатываемых килобайтов за единицу времени (сек.$^{-1}$);

5) $L^{(CPU)}$ – суммарная загрузка виртуальных ядер центрального процессора (%);

6) $C^{(virt)}$ – размер потребляемой виртуальной памяти (МБ);

7) $C^{(resd)}$ – размер потребляемой резидентной памяти (МБ);

8) $C^{(real)}$ – размер потребляемой реальной памяти (МБ).

Запуск каждой СОА осуществлялся 100 раз для получения усредненной оценки выбранных показателей. При проведении эксперимента использовалось оборудование, включающее виртульную машину на базе ОС Debian Jessie 8.4 со следующей конфигурацией: размер оперативной памяти 1536 MB, ЦПУ Intel Core i5-2400 3140.366 MHz, размер кэша третьего уровня 6144 KB, число виртуальных процессоров 3.

Разработанный анализатор демонстрирует наилучшие показатели среди остальных сравни-

ваемых СОА. Средняя скорость обработки сетевых пакетов при помощи данного программного инструмента составляет 324,8×1024 сек.$^{-1}$, что более чем в 1,5 раза превышает аналогичный показатель, демонстрируемый Suricata в режиме single, при этом размер потребляемой анализатором реальной памяти почти в 1,5 раза ниже, чем у указанной СОА. Столь существенный выигрыш оказался возможным благодаря разработанным в [21] механизмам fnv-хеширования сетевых соединений и преаллокации памяти для хранения записей об активных сетевых соединениях.

Заключение

Значимость полученных результатов – экспериментальной и сравнительной оценки разработанной модели, а также формализованного представления процесса сигнатурного анализа сетевого трафика в виде модельно-алгоритмического аппарата – состоит в следующем.

Предложенная математическая модель сигнатурного анализа сетевого трафика базируется на теориях множеств и поиска информации, приемах параллельного и сетевого программирования, обеспечивает высокую скорость выявления сетевых атак и позволяет выполнять поиск нарушений в компьютерной сети с заданными связями между узлами.

Разработанная модель восстановления сетевых потоков данных позволяет осуществлять поиск вредоносной последовательности команд, передаваемой в нескольких отдельных пакетах и TCP-сессиях. Это достигается за счет применения разработанных алгоритмов дефрагментации IP-пакетов и реассемблирования TCP-пакетов.

Спроектированная модель представления сигнатур обеспечивает полное покрытие известных шаблонов сетевых атак за счет универсальности представления сигнатурного правила, которое может функционировать как конечный автомат, сеть Петри, алгоритм поиска подстрок или регулярное выражение.

В результате проведенного эксперимента было достигнуто улучшение показателей оперативности и ресурсопотребления реализованного программного анализатора в 1,5 раза по сравнению с аналогичными показателями, вычисленными для СОА Snort, Suricata и Bro.

Направление дальнейших исследований включает разработку и оптимизацию алгоритмов автоматической генерации сигнатурных правил с использованием деревьев решений [22] и приемов генетического программирования [23].

Список источников

1. Kumar S., Spafford E.H. A Pattern Matching Model for Misuse Intrusion Detection // Proceedings of the 17th National Computer Security Conference (NIST, Baltimore, USA, 11–14 October 1994). 1994. Vol. 1. PP. 11–21.

2. Браницкий А.А. Программные способы повышения эффективности функционирования сетевых сигнатурных систем обнаружения атак // VII Международная научно-техническая и научно-методическая конференция «Актуальные проблемы инфотелекоммуникаций в науке и образовании» (АПИНО, Санкт-Петербург, Российская Федерация, 28 февраля – 01 марта 2018 г.). СПб.: СПбГУТ, 2018. Т. 1. С. 118–123. EDN:XSUFGX

3. Gowrison G., Ramar K., Muneeswaran K., Revathi T. Efficient context-free grammar intrusion detection system // International Journal of Innovative Computing Information and Control. 2011. Vol. 7. Iss. 8. PP. 4779–4788.

4. Kazachkin D.S., Gamayunov D.Y. Network traffic analysis optimization for signature-based intrusion detection systems // Proceedings of the Spring/Summer Young Researchers' Colloquium on Software Engineering. 2008. Iss. 2. DOI:10.15514/SYRCOSE-2008-2-5

5. Kruegel C., Toth T. Using Decision Trees to Improve Signature-Based Intrusion Detection // Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection (RAID 2003, Pittsburgh, USA, 8–10 September 2003). Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2003. Vol. 2820. PP. 173–191. DOI:10.1007/978-3-540-45248-5_10

6. Ilgun K., Kemmerer R.A., Porras P.A. State transition analysis: A rule-based intrusion detection approach // IEEE Transactions on Software Engineering. 1995. Vol. 21. Iss. 3. PP. 181–199. DOI:10.1109/32.372146

7. Kumar S., Spafford E.H. A Software Architecture to Support Misuse Intrusion Detection // Proceedings of the 18th National Information Security Conference (NIST, Baltimore, USA, 10–13 October 1995). 1995. Vol. 1. PP. 194–204.

8. Zhang X., Wu T., Zheng Q., Zhai L., Hu H., Yin W., et al. Multi-Step Attack Detection Based on Pre-Trained Hidden Markov Models // Sensors. 2022. Vol. 22. Iss. 8. P. 2874. DOI:10.3390/s22082874

9. Lunt T.F., Tamaru A., Gillham F. A Real-Time Intrusion Detection Expert System (IDES). Final Technical Report for SRI Project 6784. 1992.

10. Lindqvist U., Porras P.A. eXpert-BSM: A host-based intrusion detection solution for Sun Solaris // Seventeenth Annual Computer Security Applications Conference (New Orleans, USA, 10–14 December 2001). IEEE, 2001. PP. 240–251. DOI:10.1109/ACSAC.2001.991540

11. Lindqvist U., Porras P.A. Detecting computer and network misuse through the production-based expert system toolset (P-BEST) // Proceedings of the 1999 IEEE Symposium on Security and Privacy (Oakland, USA, 14–14 May 1999). IEEE, 1999. PP. 146–161. DOI:10.1109/SECPRI.1999.766911

12. Kim H.J., Choi J. Recommendations for Responding to System Security Incidents Using Knowledge Graph Embedding // Electronics. 2023. Vol. 13. Iss. 1. P. 171. DOI:10.3390/electronics13010171

13. Wang Y., Jere S., Banerjee S., Liu L., Shetty S., Dayekh S. Anonymous Jamming Detection in 5G with Bayesian Network Model Based Inference Analysis // Proceedings of the 23rd International Conference on High Performance Switching and Routing (HPSR, Taicang, China, 06–08 June 2022). IEEE, 2022. PP. 151–156. DOI:10.1109/HPSR54439.2022.9831286

14. Almseidin M., Al-Sawwa J., Alkasassbeh M., Alweshah M. On detecting distributed denial of service attacks using fuzzy inference system // Cluster Computing. 2023. Vol. 26. Iss. 2. PP. 1337–1351. DOI:10.1007/s10586-022-03657-5

15. Braden R., Borman D., Partridge C. RFC 1071: Computing the Internet Checksum. 1988. URL: https://dl.acm.org/doi/pdf/10.17487/RFC1071 (Accessed 22.08.2025)

16. Fall K.R., Stevens W.R. TCP/IP illustrated. Addison-Wesley Professional, 2012. Vol. 1. 1008 p.

17. Laraba A., François J., Chrisment I., Chowdhury S.R., Boutaba R. Detecting Multi-Step Attacks: A Modular Approach for Programmable Data Plane // NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium (Budapest, Hungary, 25–29 April 2022). IEEE Press, 2022. PP. 1–9. DOI:10.1109/NOMS54207.2022.978993

18. Madhloom J.K., Noori Z.H., Ebis S.K., Hassen O.A., Darwish S.M. An Information Security Engineering Framework for Modeling Packet Filtering Firewall Using Neutrosophic Petri Nets // Computers. 2023. Vol. 12. Iss. 10. P. 202. DOI:10.3390/computers12100202

19. Браницкий А.А. Алгоритмы параллельного поиска шаблонных подстрок при реализации сигнатурных правил СОА // Региональная информатика и информационная безопасность: сборник трудов конференции (Санкт-Петербург, Российская Федерация, 01–03 ноября 2017 г.). СПб.: Санкт-Петербургское Общество информатики, вычислительной техники, систем связи и управления, 2017. Т. 4. С. 210–212. EDN:XTONUL

20. Thota K.K., Raj R.J.R. Efficient Regular Expression Matching and Hardware-Accelerated Finite Automata Pattern Recognition in NIDS // Proceedings of the 6th International Conference on Recent Trends in Advance Computing (ICRTAC, Chennai, India, 14–15 December 2023). IEEE, 2023. PP. 349–353. DOI:10.1109/ICRTAC59277.2023.10480760

21. Браницкий А.А. Обнаружение аномальных сетевых соединений на основе гибридизации методов вычислительного интеллекта. Дис. … канд. тех. наук. СПб.: Санкт-Петербургский Федеральный исследовательский центр РАН, 2018. 305 с. EDN:OQGUHC

22. Febrita R.E., Hakim L., Utomo A.P. The Implementation of Machine Learning for Optimizing Network-Based Intrusion Detection in the Snort Application // Proceedings of the 6th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI, Batam, Indonesia, 11–12 December 2023). 2023. PP. 141–147. DOI:10.1109/ISRITI60336.2023.10467566

23. Makanju A., LaRoche P., Zincir-Heywood A.N. A Comparison Between Signature and Machine Learning Based Detectors. URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=16e92def7fca8e41894e875f2eb9d4118a4d09df (Accessed 22.08.2025)

References

1. Kumar S., Spafford E.H. A Pattern Matching Model for Misuse Intrusion Detection. *Proceedings of the 17th National Computer Security Conference, NIST, 11–14 October 1994, Baltimore, USA, vol.1*. 1994. p.11–21.

2. Branitskiy A. Software Ways for Enhancing the Effectiveness of the Network-Based Signature Attack Detection Systems. *Proceedings of the VIIth International Conference on Infotelecommunications in Science and Education, 28 February – 1 March 2018, St. Petersburg, Russian Federation, vol.1.* St. Petersburg: Saint-Petersburg State University of Telecommunications Publ.; 2018. p.118–123. (in Russ.) EDN:XSUFGX

3. Gowrison G., Ramar K., Muneeswaran K., Revathi T. Efficient context-free grammar intrusion detection system. *International Journal of Innovative Computing Information and Control.* 2011;7(8):4779–4788.

4. Kazachkin D. S., Gamayunov D. Y. Network traffic analysis optimization for signature-based intrusion detection systems. *Proceedings of the Spring/Summer Young Researchers' Colloquium on Software Engineering.* 2008;2. DOI:10.15514/SYRCOSE-2008-2-5

5. Kruegel C., Toth T. Using Decision Trees to Improve Signature-Based Intrusion Detection. *Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection, RAID 2003, 8–10 September 2003, Pittsburgh, USA. Lecture Notes in Computer Science, vol.2820*. Berlin, Heidelberg: Springer; 2003. p.173–191. DOI:10.1007/978-3-540-45248-5_10

6. Ilgun K., Kemmerer R. A., Porras P. A. State transition analysis: A rule-based intrusion detection approach. *IEEE Transactions on Software Engineering.* 1995;21(3):181–199. DOI:10.1109/32.372146

7. Kumar S., Spafford E.H. A Software Architecture to Support Misuse Intrusion Detection. *Proceedings of the 18th National Information Security Conference, NIST, 10–13 October 1995, Baltimore, USA, vol.1.* 1995. p.194–204.

8. Zhang X., Wu T., Zheng Q., Zhai L., Hu H., Yin W., et al. Multi-Step Attack Detection Based on Pre-Trained Hidden Markov Models. *Sensors*. 2022;22(8):2874. DOI:10.3390/s22082874

9. Lunt T.F., Tamaru A., Gillham F. *A Real-Time Intrusion Detection Expert System (IDES). Final Technical Report for SRI Project 6784*. 1992.

10. Lindqvist U., Porras P.A. eXpert-BSM: A host-based intrusion detection solution for Sun Solaris. *Seventeenth Annual Computer Security Applications Conference, 10–14 December 2001, New Orleans, USA*. IEEE; 2001. p.240–251. DOI:10.1109/ACSAC.2001.991540

11. Lindqvist U., Porras P.A. Detecting computer and network misuse through the production-based expert system toolset (P-BEST). *Proceedings of the 1999 IEEE Symposium on Security and Privacy, 14–14 May 1999, Oakland, USA*. IEEE; 1999. p.146–161. DOI:10.1109/SECPRI.1999.766911

12. Kim H.J., Choi J. Recommendations for Responding to System Security Incidents Using Knowledge Graph Embedding. *Electronics.* 2023;13(1):171. DOI:10.3390/electronics13010171

13. Wang Y., Jere S., Banerjee S., Liu L., Shetty S., Dayekh S. Anonymous Jamming Detection in 5G with Bayesian Network Model Based Inference Analysis. *Proceedings of the 23rd International Conference on High Performance Switching and Routing, HPSR, 06–08 June 2022, Taicang, China*. IEEE; 2022. p.151–156. DOI:10.1109/HPSR54439.2022.9831286

14. Almseidin M., Al-Sawwa J., Alkasassbeh M., Alweshah M. On detecting distributed denial of service attacks using fuzzy inference system. *Cluster Computing.* 2023;26(2):1337–1351. DOI:10.1007/s10586-022-03657-5

15. Braden R., Borman D., Partridge C. *RFC 1071: Computing the Internet Checksum*. 1988. URL: https://dl.acm.org/doi/pdf/10.17487/RFC1071 [Accessed 22.08.2025]

16. Fall K.R., Stevens W.R. *TCP/IP illustrated, vol.1.* Addison-Wesley Professional, 2012. 1008 p.

17. Laraba A., François J., Chrisment I., Chowdhury S.R., Boutaba R. Detecting Multi-Step Attacks: A Modular Approach for Programmable Data Plane. *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, 25–29 April 2022, Budapest, Hungary*. IEEE Press; 2022. p.1–9. DOI:10.1109/NOMS54207.2022.978993.

18. Madhloom J.K., Noori Z.H., Ebis S.K., Hassen O.A., Darwish S.M. An Information Security Engineering Framework for Modeling Packet Filtering Firewall Using Neutrosophic Petri Nets. *Computers*. 2023;12(10):202. DOI:10.3390/computers12100202

19. Branitskiy A.A. Algorithms for parallel search of template substrings in the implementation of signature rules of ADSs. *Regional Informatics and Information Security: Conference Proceedings, 1–3 November 2017, St. Petersburg, Russian Federation, vol.4.* St. Petersburg: Saint-Petersburg Society for Informatics, Computer Engineering, Communications and Control Systems Publ.; 2017. p.210–212. EDN:XTONUL

20. Thota K.K., Raj R.J.R. Efficient Regular Expression Matching and Hardware-Accelerated Finite Automata Pattern Recognition in NIDS. *Proceedings of the 6th International Conference on Recent Trends in Advance Computing, 14–15 December 2023, ICRTAC, Chennai, India*. IEEE; 2023. p.349–353. DOI:10.1109/ICRTAC59277.2023.10480760

21. Branitskiy A.A. *Detection of Anomalous Network Connections Based on Hybridization of Computational Intelligence Methods*. Ph.D. Dissertation. St. Petersburg: St. Petersburg Federal Research Center of the Russian Academy of Sciences Publ.; 2018. 305 p. (in Russ.) EDN:OQGUHC

22. Febrita R.E., Hakim L., Utomo A.P. The Implementation of Machine Learning for Optimizing Network-Based Intrusion Detection in the Snort Application. *Proceedings of the 6th International Seminar on Research of Information Technology and Intelligent Systems, ISRITI, 11–12 December 2023, Batam, Indonesia*. 2023. p.141–147. DOI:10.1109/ISRITI60336.2023.10467566

23. Makanju A., LaRoche P., Zincir-Heywood A.N. *A Comparison Between Signature and Machine Learning Based Detectors*. URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=16e92def7fca8e41894e875f2eb9d4118a4d09df (Accessed 22.08.2025)

Статья поступила в редакцию 14.03.2025; одобрена после рецензирования 18.07.2025; принята к публикации 18.07.2025.
The article was submitted 14.03.2025; approved after reviewing 18.07.2025; accepted for publication 18.07.2025.

Информация об авторах:

БРАНИЦКИЙ Александр Александрович
кандидат технических наук, доцент, доцент кафедры защищенных систем связи Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича
https://orcid.org/0000-0003-3104-0622

БРАНИЦКАЯ Наталия Алексеевна
заместитель начальника отдела международного образовательного сотрудничества Санкт-Петербургского государственного университета
https://orcid.org/0009-0001-5629-6715

Авторы сообщают об отсутствии конфликтов интересов.
The authors declare no conflicts of interests.