www.volsu.ru

CC BY

DOI: https://doi.org/10.15688/NBIT.jvolsu.2025.1.3

УДК 004.056
ББК 32.973

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СИСТЕМ ДЕТЕКТИРОВАНИЯ ВТОРЖЕНИЯ SNORT И SURICATA

Евгений Сергеевич Семенов

Кандидат технических наук, доцент,
заведующий кафедрой телекоммуникационных систем,
Волгоградский государственный университет
semenov.evgeniy@volsu.ru
просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

Никита Романович Дмитриенко

Аспирант, кафедра телекоммуникационных систем,
Волгоградский государственный университет
ITTa-231_682251@volsu.ru
просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

Иван Федорович Гладков

Студент, кафедра телекоммуникационных систем,
Волгоградский государственный университет
ITSm-231_227876@volsu.ru
просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

Аннотация. Рассмотрены различные подходы к организации работы систем детектирования вторжений с открытым исходным кодом. В этом контексте сетевые системы обнаружения вторжений становятся важным элементом стратегической безопасности, обеспечивая надежную защиту от потенциальных угроз целостности, конфиденциальности и доступности данных в сети. В литературе предложено множество систем обнаружения вторжений, однако Snort и Suricata являются наиболее известными и распространенными среди решений с открытым исходным кодом. В статье проводится сравнение производительности этих двух систем обнаружения вторжений. Количественное исследование выполнялось в виртуализированной сетевой среде с целью оценки производительности каждой системы по таким показателям, как точность, использование памяти и процессора, скорость обработки пакетов и уровень потерь пакетов. В результате исследования было установлено, что Snort и Suricata демонстрируют хорошие результаты, но имеют некоторые ограничения, которые требуют устранения.

Ключевые слова: IPS, NIDS, HIDS, Snort, Suricata.

© Семенов Е.С., Дмитриенко Н.Р., Гладков И.Ф., 2025

Введение

С ростом зависимости от информационных технологий и Интернета вещей становится крайне важно, чтобы ИТ-специалисты стали реагировать более остро на растущее число кибератак. Это бдительность необходима, для внедрения новых стратегий предотвращения дальнейших подобных атак. Одним из наиболее проблемных элементов кибербезопасности является быстрое и постоянное развитие угроз сетевой безопасности, будь то уязвимости или новые методы взлома. Киберпреступники быстро совершенствуют свои методы, что подчеркивает необходимость развития современных систем безопасности. Следовательно, [5; 10], одним из первых действий для инициирования эффективной стратегии кибербезопасности является получение понимания угрозы. Существует достаточно большое количество реализаций систем обнаружения, предоставляемой на коммерческой основе и реализованных как в аппаратном, так и программном виде. Однако все большее распространение получает использование систем детектирования вторжения, разрабатываемых под открытой лицензией, силами сообщества.

IDS – программные или аппаратные системы, автоматизирующие процесс регистрации, логгирования возникающих в системе или сети событий, и анализируют их на предмет подозрительной активности. С увеличением количества сетевых атак возрастает роль IDS как неотъемлемое дополнение инфраструктуры безопасности.

Обнаружение вторжений – это процесс мониторинга событий в компьютерной системе или сети с целью их анализа. Вторжения определяются как попытки компрометации конфиденциальности, целостности, доступности или обхода механизмов безопасности. Вторжения могут осуществляется как из внешней сети, так и изнутри, силами пользователей системы по неосторожности, или же намеренно. Системы IDS представляют собой программные или аппаратные устройства, которые автоматизируют процесс мониторинга и анализа событий в сети или системе для обнаружения вторжений.

IDS становятся необходимым дополнением системы безопасности организации. Однако технологии обнаружения вторжений не гарантируют абсолютную безопасность системы. Несмотря на это, использование IDS имеет несколько преимуществ:

1. В первую очередь, возможность реагировать на атаки позволяет сделать атакующих ответственными за свои действия. В TCP/IP сетях сложно обеспечить идентификацию адреса источника или другие формы идентификации, но IDS все же дает возможность требовать от атакующих ответа за свои действия. Это особенно важно в системах с неэффективными механизмами аутентификации и идентификации.

2. Возможность блокирования значит, что система может распознать некоторые действия или события как атаку и принять меры по блокированию этих источников. Важно отметить, что возможность реагирования на атаки и возможность блокирования имеют отличия в применении.

Все системы обнаружения атак IDS работают по одному принципу – поиск угрозы путем анализа трафика. Отличие в принципе их работы заключается в методе обработке и анализа трафика. По этому критерию, выделяют два основных вида: сигнатурные, основанные на детектировании аномалий. Сигнатурные IDS функционируют похожим образом на антивирусное ПО. Они просматривают сигнатуры и сопоставляют их с базой данных, постоянно обновляемой для обеспечения корректной работы. Но есть определенный недостаток: если база не доступна по определенным причинам, сеть становится уязвимой. Также возникает риск того, что новая атака и ее сигнатура неизвестны, что делает систему бесполезной. Сигнатурные IDS способны отслеживать шаблоны или состояния. Шаблоны – это те сигнатуры, которые хранятся в постоянно обновляемой базе. Состояния – это любые действия внутри системы. Начальное состояние системы – нормальная работа, отсутствие атаки. После успешной атаки система переходит в скомпрометированное состояние, то есть заражение прошло успешно. Каждое действие (например, установка соединения по протоколу, не соответствующему политике безопасности компании, активизация ПО и т. д.) способно изменить состояние. Поэтому сигнатурные IDS отслеживают не

действия, а состояние системы. В то время как NIDS чаще отслеживают шаблоны, HIDS – в основном состояния.

IDS, основанные на аномалиях. Данная разновидность IDS по принципу работы в чем-то схожа с отслеживанием состояний, только имеет больший охват. IDS, основанные на аномалиях, используют машинное обучение. Для правильной работы таких систем обнаружения угроз необходим пробный период обучения. Система анализирует работу сети в текущий момент, сравнивает с аналогичным периодом и выявляет аномалии. Аномалии делятся на три категории:

– статистические;

– аномалии протоколов;

– аномалии трафика.

Статистические аномалии выявляются, когда система IDS составляет профиль штатной активности (объем входящего / исходящего трафика, запускаемые приложения и т. д.) и сравнивает его с текущим профилем.

Для выявления аномалий протоколов IDS-система анализирует коммуникационные протоколы, их связи с пользователями, приложениями и составляет профили. Например, веб-сервер должен работать на порте 80 для HTTP и 443 для HTTPS. Если для передачи информации по HTTP или HTTPS будет использоваться другой порт, IDS пришлет уведомление.

Также IDS способны выявлять аномалии, любую небезопасную или даже угрожающую активность в сетевом трафике. Одним из примеров применения таких систем является предотвращение DoS-атаки. Обычно, при попытках проведения подобного рода атак, они предотвращаются на уровне сетевого экрана. В случае более комплексных и координированных атак трафик может рассылаться с разных адресов, что усложняет детектирования подобных вторжений. Технологии IDS позволяют анализировать сетевой трафик и заблаговременно предотвращают подобные атаки.

IDS делятся на два основных вида по месту установки:

– Network Intrusion Detection System (NIDS) – работает на уровне сети;

– Host-based Intrusion Detection System (HIDS) – на уровне отдельно взятого хоста.

Сетевые системы мониторинга вторжений имеют свои преимущества, поскольку они следят за всей сетью, не требуя дополнительных решений. Однако подобная защита имеет свои недостатки. Ресурсы процессора и ОЗУ используются для отслеживания всего сетевого трафика. Чем больше этот трафик, тем больше ресурсов потребуется, что может привести к перегрузке системы и заметным задержкам в передаче данных. К тому же большой объем информации может обеспечить возможность проведения кибератаки на уязвимую сеть.

Хостовые системы мониторинга вторжений (HIDS) занимаются поиском вторжений на уровне узла. Они устанавливаются на отдельный хост внутри сети и защищают только его. HIDS анализируют входящий и исходящий трафик только для этого устройства. Работа системы HIDS основана на создании снимков файлов: он делает снимок текущей версии и сравнивает его с предыдущей версией, чтобы обнаружить возможные угрозы. Лучше всего устанавливать HIDS на критически важные машины в сети, которые редко меняют конфигурацию [5; 8].

Suricata – программное обеспечение для реализации IDS (система обнаружения вторжений), IPS (система предотвращения вторжений) и NSM (мониторинг сетевой безопасности). Suricata поддерживает формат правил Snort, но при этом имеет ряд отличий, таких как более продвинутая система IPS, работу в многопоточном режиме, возможность использования GPU в режиме IDS, многозадачность, высокая производительность.

Snort – это свободная система предотвращения вторжений (IPS) и обнаружения вторжений (IDS) с открытым исходным кодом, которая способна регистрировать пакеты и анализировать трафик в IP-сетях в режиме реального времени. Она может выполнять протоколирование, анализ и поиск содержимого, а также широко применяться для активного блокирования или пассивного обнаружения целого ряда атак и сканирований, таких как атаки на буфер переполнения, скрытое сканирование портов, атаки на веб-приложения, SMB-зондирование и определение операционной системы [9].

Выбор тестировочного трафика для оценки NIDS был произведен с учетом различных соображений. Так, при замере он может быть

как подлинным, так и воссозданным. Для полноты теста считается корректным производить замер с использованием подлинного системного трафик, в условиях настоящей нагрузки. Однако из-за большого числа проводимых тестов, использующих постоянный системный трафик, результат мог сильно разнится из-за непостоянной природы сетевой нагрузки. В качестве решения было предложено использовать поток трафика, который был записан в формат PCAP (перехват пакета).

Материалы и методы

Испытания и тесты были проведены на виртуальном полигоне, что сыграло важную роль в обеспечении избыточности и точности экспериментов в ходе данного исследования. Использование виртуализации позволило создать несколько изолированных и независимых сред для каждой из тестируемых конфигураций [7]. Для реализации виртуальных машин в основном использовалась платформа VirtualBox, которая была выбрана за ее стабильность и гибкость при работе с различными операционными системами. В качестве основной рабочей среды для виртуальных машин была выбрана операционная система Ubuntu 24.04 LTS, что обеспечивало надежную и совместимую платформу для развертывания всех необходимых инструментов и приложений. Каждая виртуальная машина была выделена с учетом определенных ресурсов для выполнения тестов: четыре потока процессора на базе шестиядерного AMD 7500f с тактовой частотой 3,7 ГГц, а также 4 ГБ оперативной памяти DDR5 с частотой 6000 МГц, что позволяло эффективно обрабатывать нагрузку в условиях испытаний. Для хранения данных каждая виртуальная машина получила максимальный объем диска в 35 ГБ. Системный трафик был воспроизведен для каждой из виртуальных сред индивидуально, что позволяло создавать независимые тестовые сценарии и исключать взаимное влияние различных процессов и сервисов. В качестве средств обнаружения вторжений были использованы две популярные системы – Snort и Suricata, каждая из которых была настроена с применением стандартных правил сообщества, что позволило проверить их производительность и эффективность в реальных условиях. Для каждой из систем использовались следующие версии: Suricata v7.0.8 и Snort v2.9.20, что также отражает актуальные и распространенные конфигурации.

Маршрутизатором в тестируемой сети служил MikroTIk RouterBoard RB951G-2HnD, который обеспечивал правильную маршрутизацию трафика между различными компонентами сети. Операционная система, на которой происходила настройка и развертывание систем обнаружения вторжений, была также развернута в виртуальной среде, что позволило обеспечить высокую гибкость и адаптируемость в ходе тестирования. Схема созданной сети и подробное описание топологии были представлены на рисунке 1, который иллюстрирует структуру сети и взаимодействие различных ее элементов.

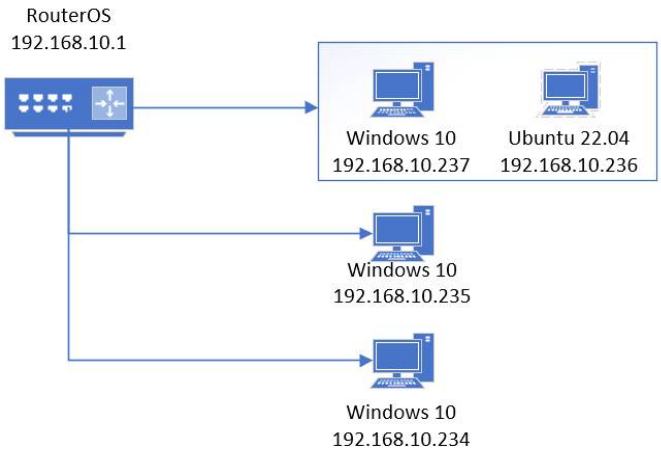


Рис. 1. Топология испытываемой сети

На всех машинах перед проведением тестирования, была произведена чистая установка системы, настройки сетевого экрана были оставлены по умолчанию. Настройка пересылки пакетов к IDS производилась посредством инструментов, поставляемых в базовой версии ОС RouterOS. Установка Suricata производилась из репозитория. После чего была произведена настройка Suricata с учетом используемых сетевых интерфейсов было. Произведено обновление Suricata, а также выбор библиотеки, которая будет предоставлять правила, используемый для анализа трафик.

Перед установкой Snort, была произведена подготовка системы. Для этого, помимо обновления, потребовалось добавление библиотеки LibDAQ. Установка Snort производилась посредством компиляции исходного кода. В дальнейшем тестировании использовались общедоступный набор правил Snort, предоставляемый на сайте проекта. Для тестирования использовался сервис testmynids.org, запуск производился в интерактивном режиме. При проверке скорости обработки трафика и детектирования вторжения настроенными системами, в сети генерировалась искусственная нагрузка [1–2], значения которой показаны в таблице 1 и отражены на рисунках 3 и 4. Это позволяло, помимо всего прочего, изучить поведение систем детектирования в условиях высокой нагрузки на сеть. Трафик генерировался с помощью утилиты iPerf3, в передаче участвовали машины с сетевыми адресами 192.168.10.235 и 192.168.10.234. Пример записей, составленных системами обнаружения вторжения предоставлен в Приложении А. В таблице 2 предоставлена общая информация о скорости обнаружения вторжения, а также общая точность работы систем обнаружения. На рисунке 2 предоставлен маршрут генерируемого трафика, а также направление атаки.

Результаты

Поток трафика атаки был направлен на обе сетевые IDS с плавающими настройками процессора. Способность сетевых IDS просматривать пакеты в условиях нагрузки, наряду с точностью срабатывания являлись приоритетными показателями для оценки работы систем детектирования. Всякий раз, когда проводилась проверка, вводные и выходные данные работы сетевых IDS документировались. Предоставленные сведения об особенностях работы сетевых IDS показали количество произведенных предупреждений, какое количество пакетов было перехвачено, и какая доля системных оповещений была произведена корректно [4].

В этой первой группе экспериментов мы оцениваем производительность двух NIDS с точки зрения использования ЦП и памяти, то есть скорости обработки и ресурсов памяти,

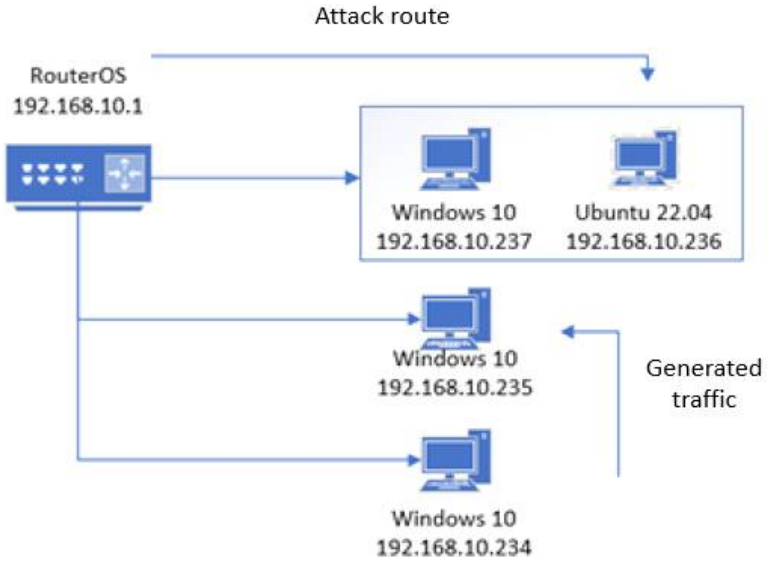


Рис. 2. Схема прохождения релевантного трафика

которые используются процессом NIDS во время анализа полученных пакетов и генерации оповещений [6]. Чтобы определить показатели потребления ЦП и памяти, мы записали показатель потребления ЦП, взяв наибольшее значение, которое появляется на интерфейсе HTOP и T-SUI во время прохождения теста. Каждый тест повторяется несколько раз, чтобы получить надежные результаты.

Потеря пакетов и потеря оповещений: в этой группе экспериментов мы оцениваем производительность двух систем с точки зрения потери пакетов, то есть процент потерянных и не проанализированных NIDS пакетов. Эта проблема обычно возникает, когда скорость и объем трафика увеличиваются. В этом случае NIDS не может выполнить проверку всех пакетов, что увеличивает частоту ложных отрицательных результатов и позволяет потенциально вредоносным пакетам проникать во внутреннюю сеть без обнаружения. Поэтому NIDS должна быть достаточно эффективной для обработки сетевого трафика при увеличении скорости и объема трафика и, следовательно, минимизировать частоту потери пакетов. В таблице 2 указана успешность детектирования различных типов атак.

При проверке скорости обработки трафика и детектирования вторжения настроенными системами, в сети генерировалась искусственная нагрузка. Это позволяло, помимо всего прочего, изучить поведение систем

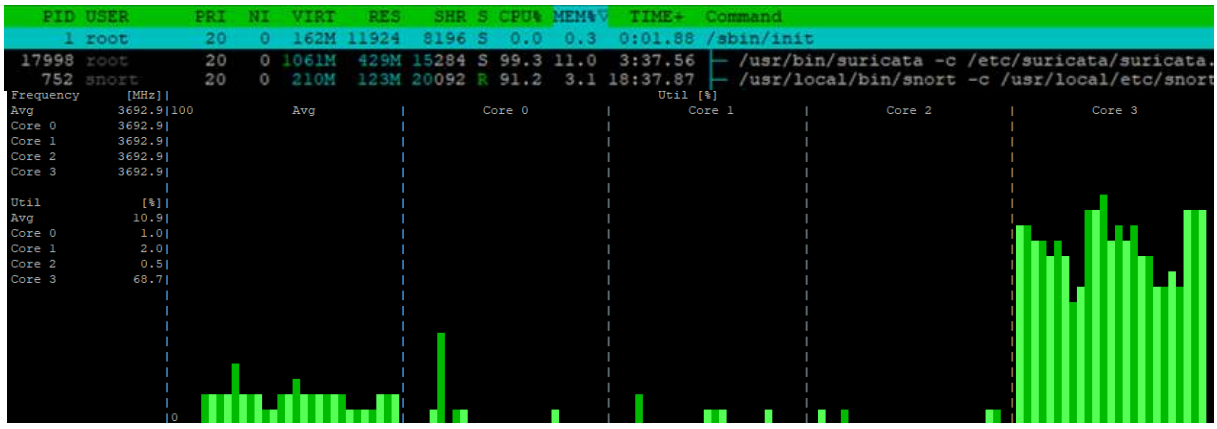


Рис. 3. Задействование системных ресурсов Snort под нагрузкой

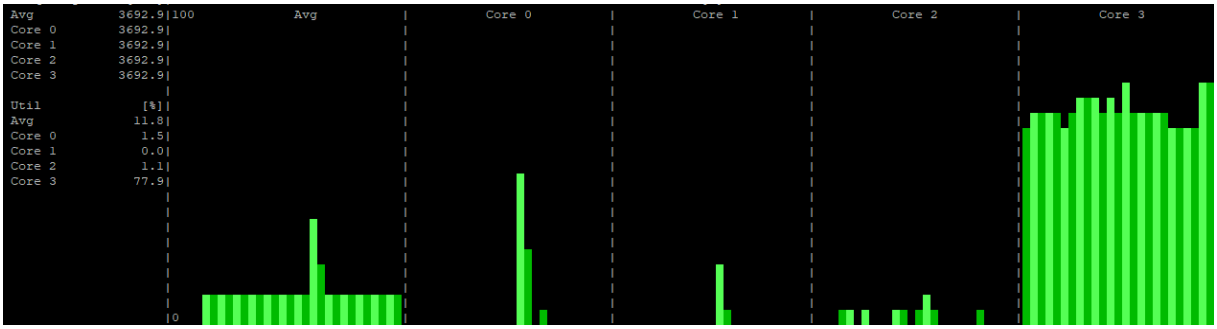


Рис. 4. Задействование системных ресурсов Suricata под нагрузкой

Таблица 1

Средние значения задействования системных ресурсов системами обнаружения

Наименование ресурса	Suricata	Snort
Средняя нагрузка на ЦП	75 %	55 %
Оперативная память	429Mb	123Mb
Объем занимаемой виртуальной памяти	1061Mb	210Mb

Таблица 2

Распознавание различных типов атак системами IDS

Вид детектируемой атаки	Suricata	Snort
Linux UID	Детектирование и определение	Детектирование и определение
HTTP Basic Authentication	Детектирование и определение	Атака не зафиксирована
HTTP Malware User-Agent	Детектирование и определение	Атака не зафиксирована
Bad Certificate Authorities	Детектирование и определение	Атака не зафиксирована
Tor .onion DNS response and known IPs connection	Детектирование и определение	Атака не зафиксирована
EXE or DLL download over HTTP	Детектирование и определение	Атака не зафиксирована
PDF download with Embedded File	Детектирование и определение	Атака не зафиксирована
Simulate SSH Outbound Scan	Детектирование и определение	Атака не зафиксирована
Miscellaneous domains	Детектирование и определение	Атака не зафиксирована
Anonymous filesharing website	Детектирование и определение	Атака не зафиксирована

Таблица 3

Средние значения потери пакетов и скорости обнаружения

Наименование IDS	Suricata	Snort
Среднее время реакции	350 мс	300 мс
Количество зафиксированных атак	1000	921

детектирования в условиях высокой нагрузки на сеть. Трафик генерировался с помощью утилиты iPerf3, в передаче участвовали машины с сетевыми адресами 192.168.10.235 и 192.168.10.234. В таблице 3 предоставлена общая информация о скорости обнаружения вторжения, а также общая точность работы систем обнаружения.

Из результатов проведенного сравнения видно, что стандартные правила Snort во многом уступают тем, которыми располагает Suricata. В условиях настоящей нагрузки, использование Snort потребует от администраторов сети больших усилий по настройке и написанию правил, необходимых для надежного детектирования вторжений и нежелательной активности. Однако среднее время детектирования Snort было на 50 мс быстрее. Это опережение сохранялось и при использовании системами детектирования одинаковых правил что может сигнализировать меньшем потреблении ресурсов системой под нагрузкой.

Обсуждение

Вопрос точности является одним из наиболее важных аспектов при оценке сетевых систем обнаружения вторжений (IDS), и это измерение имеет ключевое значение при сравнении различных их различных реализаций. В рамках этого анализа рассматривались такие параметры, как возможность проведения атак на систему, обработка ложных положительных и ложных отрицательных срабатываний, ограничение доступных ресурсов и способность эффективно работать при высоких нагрузках, связанных с большим объемом передаваемых данных. Как было продемонстрировано, система Suricata показала более высокую точность, чем Snort, что позволило ей более эффективно выявлять угрозы в большинстве случаев. Однако стоит отметить, что для достижения таких результатов Suricata предъявляет более высокие требования к ресурсам системы. Это, с одной стороны, приводит к тому, что она достигает своего эксплуатационного предела быстрее, чем Snort, что проявляется в более высоком уровне потери пакетов при интенсивной нагрузке. С другой стороны, Snort в меньшей степени зависит от объемов обрабатываемого трафика и предоставленных ресурсов, что позволяет этой системе иметь более низкий коэффициент сброса пакетов в условиях работы с большими данными. Несколько тестов показали, что Suricata производит детектирование медленнее, чем Snort, особенно в условиях ограниченных ресурсов. Однако важно отметить, что выявляют Suricata как более эффектив-

ную в ряде специфических сценариев, особенно когда необходимо обрабатывать большие потоки данных. Это позволяет утверждать, что Suricata является более универсальным решением, способным адаптироваться к различным условиям эксплуатации [3].

Тем не менее, если сеть сталкивается с ситуацией, когда объем передаваемых данных значительно превышает возможности, предоставляемые одной машиной, возможен одновременный запуск нескольких экземпляров Snort на разных вычислительных блоках. Это решение может обеспечить аналогичную адаптивность, как у Suricata, потребует меньшие вычислительные мощности, однако при этом потребует дополнительной настройки для обеспечения работы нескольких инстансов Snort в рамках развернутых систем обнаружения вторжений.

Заключение

Распространенность сетевого оборудования в наше время объясняется существенным увеличением числа устройств, функционал которых в значительной степени зависит от постоянного подключения к глобальной сети. Этот тренд приводит к тому, что устройства все активнее интегрируются друг с другом, что способствует росту общей скорости передачи данных. В результате потребность в быстром обмене информацией между пользователями значительно возрастает. Возможность мгновенно получать и передавать сообщения становится все более важной, что, в свою очередь, усиливает требования к доступным решениям. Эти решения должны не только эффективно передавать генерируемый трафик, но и обеспечивать его качественную обработку с целью выявления потенциальных угроз.

В ходе тестирования была отмечена более низкая задержка детектирования у системы обнаружения вторжений Snort. В то же время Suricata демонстрирует более высокую точность детектирования ввиду более полного набора правил, поставляемых сообществом. Однако более высокие требования к обработке данных, которые предъявлялись Suricata, приводили к тому, что эта система достигала своих эксплуатационных пределов быстрее, чем Snort. Это становилось очевидным из-за увеличивающегося количества потерянных пакетов при повышенных нагрузках. Относительная зависимость между потерянными пакетами и ложными отрицаниями была проиллюстрирована в тестах для обоих программных решений. Сравнение показало, что Suricata на многоконцентрационных установках имеет меньший объем потери пакетов, чем Snort, что подтверждает ее более высокую эффективность в условиях многозадачности. Однако стоит отметить, что Suricata потребляет больше вычислительных ресурсов, что может привести к замедлению ее работы в сравнении с Snort. Это позволяет утверждать, что Suricata обладает большей универсальностью, хотя ценой этого является значительное увеличение требований к вычислительному оборудованию. В условиях, когда объем передаваемых данных превышает возможности Snort по обработке, оптимальным решением может стать запуск нескольких параллельных экземпляров Snort на разных вычислительных центрах. Такой подход может обеспечить адаптивность, схожую с Suricata, но с меньшими требованиями к каждому отдельному вычислительному центру. Однако этот метод также потребует дополнительных расходов на настройку и интеграцию отдельных приложений, обеспечивающих синхронную работу нескольких центров, что может увеличить общие затраты на развертывание и эксплуатацию системы.

REFERENCES

1. Anugrah F., Ikhwan S., Jafaruddin A.G. Implementasi Intrusion Prevention System (IPS) Menggunakan Suricata Untuk Serangan SQL Injection. *Techné: Jurnal Ilmiah Elektroteknika*, 2022, vol. 21 (2), pp. 199-210. DOI: 10.31358/techne.v21i2.320

2. Bendovschi A. Cyber-Attacks – Trends, Patterns and Security Countermeasures. *Procedia Economics and Finance*, 2015, vol. 28, pp. 24-31. DOI: 10.1016/S2212-5671(15)01077-1

3. Boukebous A., Fettache M., Bendiab G., Shiaeles S. A Comparative Analysis of Snort 3 and Suricata. *IEEE IAS Global Conference on Emerging Technologies*, 2023, pp. 1-6. DOI: 10.1109/GlobConET56651.2023.10150141

4. Gupta A., Sharma L. Performance Analysis and Comparison of Snort on Various Platforms. *International Journal of Computer Information Systems and Industrial Management Applications*, 2020, vol. 10, pp. 23-32.

5. Humayun M., Niazi M., Jhanjhi N. et al. Cyber Security Threats and Vulnerabilities: A Systematic Mapping Study. *Arabian Journal for Science and Engineering*, 2020, vol. 45, pp. 3171-3189. DOI: 10.1007/s13369-019-04319-2

6. Raharjo D., Salman M. Analyzing Suricata Alert Detection Performance Issues Based on Active Indicator of Compromise Rules. *Jurnal Teknik Informatika (Jutif)*, vol. 4 (3), pp. 601-610. DOI: 10.52436/1.jutif.2023.4.3.1013

7. Tsetse A., Tweneboah-Koduah S., Rawal B., Zheng Z., Prattipati M. A Comparative Study of System Virtualization Performance. *2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)*. Los Angeles, CA, USA, 2019, pp. 372-378. DOI: 10.1109/IRI.2019.00064

8. Ugboaja S., Osuo-Genseleke M., Chigozie-Okwum C. Cyber Attacks: A Literature Survey. *2nd International Conference on Education and Development*, 2019, pp. 295-302.

9. White J., Fitsimmons T., Matthews J. Quantitative Analysis of Intrusion Detection Systems: Snort and Suricata. *Proceedings of SPIE – The International Society for Optical Engineering*, 2013, vol. 8757. DOI: 10.1117/12.2015616

10. Yuchong L., Qinghui L. A Comprehensive Review Study of Cyber-Attacks and Cyber Security. *Emerging Trends and Recent Developments, Energy Reports*, 2021, vol. 7, pp. 8176-8186. DOI: 10.1016/j.egyr.2021.08.126

COMPARATIVE ANALYSIS OF SNORT AND SURICATA INTRUSION DETECTION SYSTEMS

Evgeny S. Semenov

Candidate of Sciences (Engineering), Associate Professor,
Head of the Department of Telecommunication Systems,
Volgograd State University
semenov.evgeniy@volsu.ru
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

Nikita R. Dmitrienko

Postgraduate Student, Department of Telecommunication Systems,
Volgograd State University
ITTa-231_682251@volsu.ru
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

Ivan F. Gladkov

Student, Department of Telecommunication Systems,
Volgograd State University
ITSm-231_227876@volsu.ru
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

Abstract. Different approaches to the organization of open-source intrusion detection systems are considered. In this context, network intrusion detection systems become an important element of network security, providing reliable protection against potential threats to the integrity, confidentiality, and availability of data in the network. Many network intrusion detection systems have been proposed in the literature, but Snort and Suricata are the best known and most common among open-source solutions. The paper compares the performance of these two network intrusion detection systems. A quantitative study was performed in a virtualized network environment to evaluate the performance of each system in terms of accuracy, memory and central processing unit usage, and packet processing speed, and packet loss rate. The study

found that Snort and Suricata perform well but have some limitations that need to be addressed. During testing, Snort's intrusion detection system exhibited lower detection latency. At the same time, Suricata demonstrates higher detection accuracy due to a more comprehensive set of community-supplied rules. However, the higher processing requirements of Suricata caused this system to reach its operational limits faster than Snort.

Key words: IPS, NIDS, HIDS, Snort, Suricata.