УДК 004.056
DOI: 10.36979/1694-500X-2022-22-8-64-81

РАСШИРЯЕМАЯ МОДУЛЬНАЯ ПЛАТФОРМА НА БАЗЕ OPEN SOURCE РЕШЕНИЙ ДЛЯ SECURITY OPERATIONS CENTER

С.В. Корякин, Э.Х. Халмухамедов

Аннотация. Рассматриваются вопросы построения расширяемых модульных платформ для центров мониторинга и реагирования на инциденты информационной безопасности. Предлагается на основе Универсальной среды проектирования автоматизированных систем защищенного исполнения разработать расширяемую модульную платформу для SOC (РМП SOC), которая сможет обеспечивать безопасность автоматизированных систем под управлением алгоритмов машинного обучения нейронной сети, то есть разработке ПО, алгоритмов управления, структуры универсальных автоматизированных систем защищенного исполнения с применением нейросетевых алгоритмов управления системой.

Ключевые слова: облачные технологии; защита информации; защита данных; сетевые сенсоры; межсетевой экран; нейросетевые алгоритмы управления; нейросеть.

SECURITY OPERATIONS CENTER ҮЧҮН OPEN SOURCE ЧЕЧИМДЕРИНИН БАЗАСЫНДА КЕҢЕЙТИЛҮҮЧҮ МОДУЛДУК ПЛАТФОРМА

С.В. Корякин, Э.Х. Халмухамедов

Аннотация. Макалада маалыматтык коопсуздук учурларына мониторинг жүргүзүү жана чара көрүү борборлору үчүн кеңейтилүүчү модулдук аянтчаларды куруу маселелери каралган. Эмгекте коопсуз аткаруунун автоматташтырылган системасын долбоорлоонун универсалдык чөйрөсүнүн негизинде нейрон тармактарын машиналык үйрөнүү алгоритмдеринин көзөмөлүндө автоматташтырылган системалардын коопсуздугун камсыздай ала турган SOC үчүн кеңейтилүүчү модулдук платформаны (РМП SOC) иштеп чыгуу сунушталат, башкача айтканда, программалык камсыздоону, башкаруу алгоритмдерин, системаны башкаруунун нейрондук-тармактык алгоритмдерди колдонуу менен коопсуз аткаруунун универсалдуу автоматташтырылган системаларынын түзүмүн иштеп чыгуу сунушталат.

Түйүндүү сөздөр: булут технологиялары; маалыматтарды коргоо; дайындарды коргоо; тармактык сенсорлор; тармактар аралык экран; нейрон тармагын башкаруу алгоритмдери; нейрон тармагы.

EXPANDABLE MODULAR PLATFORM BASED ON OPEN SOURCE SOLUTIONS FOR SECURITY OPERATIONS CENTER

S.V. Koryakin, E.Kh. Khalmuhamedov

Abstract. The article discusses the issues of building extensible modular platforms for monitoring centers and information security incident response. The paper proposes to develop an extensible modular platform for SOC (RMP SOC), which can ensure the safety of automated systems under the control of neural network machine learning algorithms, that is, in other words, software development, control algorithms, structure universal automated systems of secure execution with the use of neural network algorithms for system control.

Keywords: cloud technologies; information protection; data protection; network sensors; firewall; neuro-network control algorithms; neural network.

Введение. Основной задачей автоматизированных систем защиты информации (АСЗИ) является предотвращение намеренной или ненамеренной передачи (утечки) конфиденциальной информации за пределы информационной системы. Практика показывает, что большая часть ставших известными утечек (порядка 75 %) происходит не по злому умыслу, а из-за ошибок, невнимательности, безалаберности, небрежности работников. Выявлять подобные утечки проще. Остальная часть связана со злым умыслом операторов и пользователей информационных систем. Понятно, что инсайдеры, как правило, стараются преодолеть средства систем защиты информации. Исход этой борьбы зависит от многих факторов и поэтому гарантировать необходимый успех в данном случае весьма проблематично.

Кроме основной задачи, стоящей перед АСЗИ, существует целый ряд вторичных (побочных) задач:

- архивирование пересылаемых сообщений на случай возможных в будущем расследований инцидентов;
- предотвращение передачи вовне не только конфиденциальной, но и другой нежелательной информации (обидных выражений, спама, эротики, излишних объёмов данных и т. п.);
- предотвращение передачи нежелательной информации не только изнутри наружу, но и снаружи внутрь информационной системы;
- предотвращение использования работниками казённых информационных ресурсов в личных целях;
- оптимизация загрузки каналов;
- контроль трафика;
- контроль присутствия работников на рабочем месте.

Многие организации полагают ряд этих задач более приоритетными, чем защита от утечек информации. В связи с этим, соответствующими специалистами был разработан целый ряд программ, ориентированных в основном именно для решения вторичных задач, но способных при этом, в ряде случаев, работать и как средство защиты информации от утечек.

Следует заметить, что от полноценных АСЗИ такие программы отличает отсутствие развитых средств анализа перехваченных данных, который, как правило, производится специалистом по информационной безопасности вручную, что удобно только для небольших организаций, имеющих, например, до десяти контролируемых сотрудников.

Поэтому сегодня, как никогда, актуален вопрос анализа и защиты сетевой инфраструктуры информационной системы и баз данных во всех информационных сферах, которые играют все возрастающую роль в обеспечении безопасности жизнедеятельности общества в целом. Через эту сферу реализуется значительная часть угроз национальной безопасности государства.

Чтобы справиться со стремительно нарастающим потоком информации, вызванным научно-техническим прогрессом, субъекты предпринимательской деятельности, учреждения и организации всех форм собственности вынуждены постоянно пополнять свой арсенал разнообразными техническими средствами и системами, предназначенными для приема, передачи, обработки и хранения информации, и в которых основная защитная функция реализуется соответствующими техническими устройствами (комплексом или системой). Физические процессы, происходящие в таких устройствах при их функционировании, создают в окружающем пространстве побочные электромагнитные, акустические и другие излучения, которые в той или иной степени связаны с обработкой информации и фактически образуют соответствующие технические каналы утечек.

Известно, что защита информации от утечки по техническим каналам достигается проектно-архитектурными решениями, проведением организационных и технических мероприятий, а также выявлением портативных электронных устройств перехвата информации (закладных устройств). Следует отметить, что существует множество проблемных вопросов по защите информации, решение которых зависит от объективных и субъективных факторов, в том числе и дефицита конкретных возможностей ее технической реализации.

Для решения поставленных задач необходим детальный анализ существующих основных подходов к программно-аппаратной реализации обеспечения кибербезопасности автоматизированных систем. На сегодняшний день существуют два таких основных подхода:

- Использование Host-based и Network IDS решений, в том числе SIEM и DLP систем.
- Использование различного уровня сложности межсетевых экранов.

Рассмотрим особенности реализации перечисленных выше подходов на примере продуктов со свободной лицензией и открытым исходным кодом, который при необходимости, можно изменить или адаптировать под существующие нужды и задачи для конкретной автоматизированной системы защищенного исполнения (АИСЗИ).

Программные Host-based и Network IDS решения – это неотъемлемая часть процесса принятия решения о внедрении системы информационной/кибер безопасности. С их помощью можно детектировать различного рода аномалии, как на самих хостах, так и во внутренней информационной сети организации. К таким подозрительным действиям можно отнести: эксплуатацию уязвимостей у запущенных сервисов, повышение привилегий с помощью определенных техник, несанкционированный доступ, активность вредоносного кода и т. д. Это позволяет своевременно реагировать на внезапные угрозы безопасности.

Анализируя работу программных решений более детально, можно сказать, что Host-based Intrusion Detection System работает внутри, на каждом хосте, как отдельный программный продукт. Его основная цель – выявлять подозрительные действия путем проверки каждого события на предмет соответствия принятой модели безопасности, т. е. отслеживать все обращения программных продуктов к определенным ресурсам системы, осуществлять мониторинг изменения содержимого файлов, анализировать системные вызовы, логи всех приложений и т. д. Позволяя, тем самым, не нарушать политику безопасности. Большим преимуществом является то, что он может анализировать действия с большой точностью, определяя и фиксируя в отчетности те процессы и пользователей, которые непосредственно имеют отношение к подозрительным действиям, приводившие к компрометации. Для реализации этого, они используют информационные ресурсы двух видов: отчеты по аудиту операционной системы и системные логи. Одним из наиболее популярных представителей данного типа является: OSSEC, Wazuh, Osquery.

В данной рассматривается вариант построения расширяемой модульной платформы для SOC на примере программного обеспечения (ПО) с открытым кодом Wazuh. Наиболее важными из достоинств выбранного ПО можно отметить следующие:

- изначально это ПО разрабатывалось как форк/ответвление от OSSEC;
- может устанавливаться различными способами docker, puppet, chef, ansible;
- поддерживает мониторинг облачных технологий;
- имеет комплекс установленных требований по стандартам от CIS, PCI DSS и т. д.; покрывает не только часть Host но и Network IDS решением;
- является полностью open source продуктом;
- обладает достаточной простотой в использовании, и поддерживает интеграцию с Elasticsearch и Splunk.

Из недостатков только то, что у него сложная архитектура, где требуется не только развертывание стека ELK, но и серверных компонентов Wazuh.

В пакете технической документации от Wazuh Intrusion Detection System, описаны алгоритмы работы, по данным алгоритмам агенты (ПО, устанавливаемое объектах информационной системы) работают на уровне хоста, совмещая детектирование вторжений и программным злоупотреблением на базе аномальных явлений и сигнатурных технологий. Также агенты могут использоваться для мониторинга активности пользователей и выявления различного рода уязвимостей.

Таким образом, Wazuh агенты могут функционировать во всех популярных операционных системах: Windows, Linux, MacOS, Solaris, AIX и т. д. Используя это ПО можно предотвращать угрозы,

детектировать их, и самое главное, реагировать. Немаловажным фактом является то, что Wazuh агент, собирая данные от всех приложений, передает их на Wazuh Server путем зашифрованного и аутентифицированного канала.

Официальный сайт предоставляет такую архитектуру самого агента (рисунок 1):

Как можно видеть на рисунке 1, в архитектуру Wazuh агента заложено девять модулей: log collector, command execution, file integrity monitoring, security configuration assessment, malware detection, active response, cloud security monitoring, system inventory, containers security monitoring каждый из которых отвечает за свой индивидуальный процесс.

Log collector – это один из множества компонентов, который читает логи и события, собирает важные внутренние сообщения операционной системы и приложений. Для него присутствует поддержка Xpath событий в Windows системах, также Log collector распознает multi-lines формат для Linux. К тому же ПО поддерживает процесс обогащения информации, используя процесс предоставления дополнительной информации в виде metadata, в формате json.

Command execution – этот компонент позволяет периодически использовать авторизованные команды на хосте в целях сбора статистики о состоянии рабочей станции. Осуществлять мониторинг активности пользователей.

File integrity monitoring – данный модуль позволяет осуществлять мониторинг всей файловой системы, давать обратную связь об удалении, изменении и создания объектов. Когда в системе что-то происходит, мы можем увидеть всю картину: кто создал или удалил файл, когда именно это произошло, что за файл подвергся модификации и т. д.

Security configurations assessments – этот компонент позволяет проверить систему на наличие неправильной конфигурации установленных приложений с использованием готовых рекомендаций от Central of Internet Security Control (CIS). Довольно гибкий модуль, так как позволяет писать собственные проверки SCA в целях мониторинга и соблюдения политики безопасности.

System inventory – данный модуль затрагивает инвентаризацию всего, что есть на рабочей станции, начиная от версии операционной системы, сканирования открытых портов, наличия установленных приложений, и заканчивая запущенными процессами.

Malware detection – позволяет обнаруживать присутствие аномалий и руткитов, не используя сигнатурный анализ. Проводя постоянный мониторинг с целью поиска скрытых процессов, файлов и портов.

Active response – позволяет автоматически реагировать на сдетектированную угрозу: блокируя сетевые подключения, останавливая зловредные процессы и удаление вредоносных файлов. Присутствует возможности кастомизации автоматических действий под конкретные нужды.

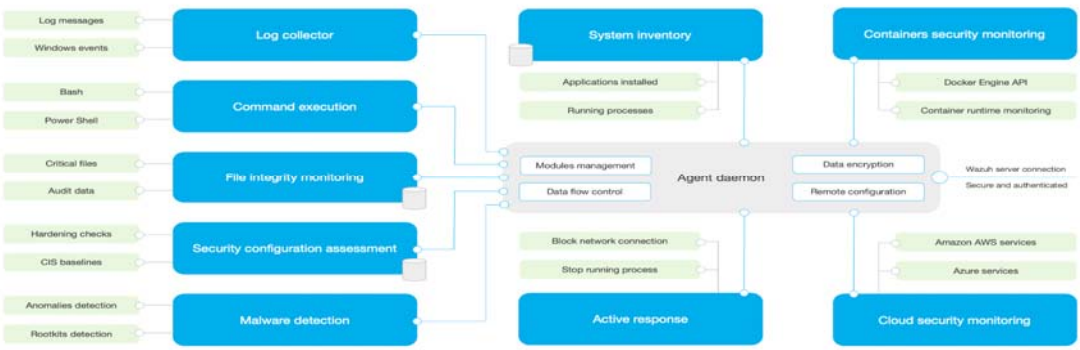

Рисунок 1 – Архитектура Wazuh агента

Containers security monitoring – данный модуль имеет интеграцию с Docker Engine API, что позволяет осуществлять мониторинг и уведомлять о изменениях в докерконтейнерах: сетевых конфигурациях, файловых хранилищах, запущенных в привилегированном режиме контейнерах с выполняемыми командами.

Cloud security monitoring – здесь в системе проведена интеграция с ведущими вендорами облачных технологий: Google, Amazon, Microsoft и т. д. Это позволяет собирать данные о всех событиях, происходящих в облаках сетей.

Wazuh Agent – отправляет всю информацию о рабочей станции, также сообщая свое состояние о конфигурации и статусе работы на Wazuh Server. После подключения возможна удаленная конфигурация, обновление и дальнейшее управление. Связь агента с сервером проходит по защищенному каналу, который поддерживает шифрование, и к тому же может сжимать данные в реальном времени. Кроме того, обладает механизмом управления потоком событий на сервер, дабы избежать переполнения избыточной информации и защиты пропускной способности канала связи. Но для всего этого необходима регистрация агента до первого подключения к серверу. Для этого используется генерируемый ключ, который нужен для аутентификации уже на стороне хоста, также этот ключ будет использоваться для шифрования передаваемых данных.

Необходимо также затронуть сторону сервера Wazuh. Он принимает всю информацию, приходящую от внутренних агентов, для того чтобы проанализировать и непременно уведомить о детектируемых угрозах посредством интеграции WebHooks с Gmail, Slack, Telegram. Одним из достоинств является то, что в нем присутствует использование Threat Intelligence источников для улучшения детектирования угроз. Немаловажным фактом является соответствие требованиям, определенным стандартам: PCI DSS, HIPAA, NIST и обогащение дополнительной информацией об атаках с использованием MITRE ATT&CK. Проводится интеграция с другими продуктами, например, создание Ticketing System с Jira [1].

На официальном сайте разработчика ПО представлена следующая архитектура Wazuh сервера (рисунок 2):

Wazuh сервер работает в Stand-alone режиме, и может быть развернут на виртуальной машине, докер контейнере или где-нибудь в облаке. Но самое главное, что там должна присутствовать операционная система Linux. На схеме можно выделить шесть важных компонентов: Agents registration service, Agents connection service, Analysis engine, Wazuh RESTful API, Wazuh cluster daemon и Filebeat [1].

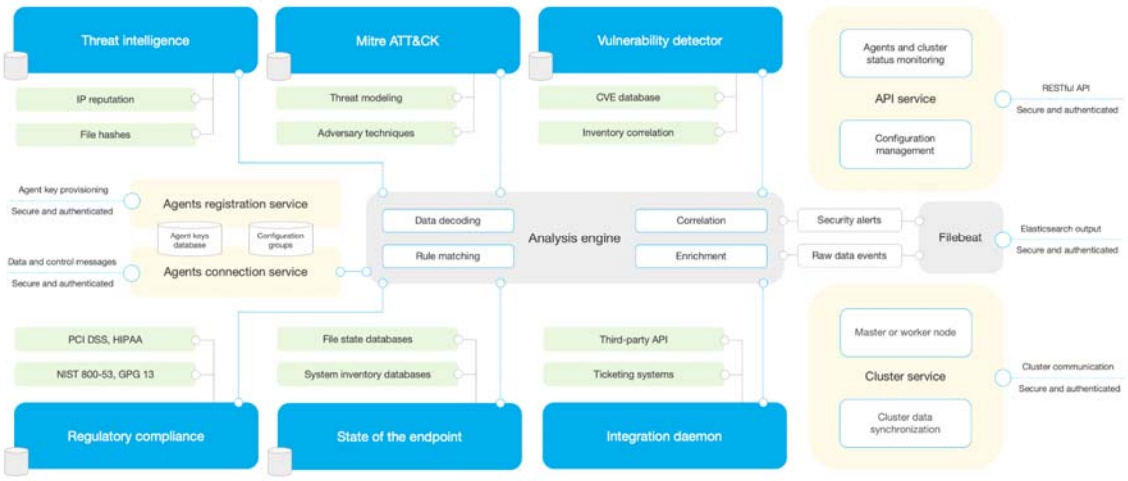

Рисунок 2 – Архитектура Wazuh сервера

Agents registration service – данный компонент отвечает за регистрацию новых агентов путем подготовки и распространения сгенерированных уникальных для каждого агента ключей аутентификации. Этот процесс выполняется как сетевая служба и поддерживает аутентификацию с помощью сертификатов TLS/SSL или путем предоставления фиксированного пароля [2, 3].

Agents connection service – этот модуль принимает данные от агентов. Используя ключ для проверки подлинности каждого агента и шифрования сообщений между агентом и сервером. Кроме того, эта служба используется для управления конфигурацией, позволяя удаленно передавать необходимые настройки агента.

Analysis engine – процесс, который направлен на анализ поступаемой информации. В этой части используются декодеры, которые могут различать типы событий: Windows events, SSH, Web Server Logs и т. д. Это делается для того, чтобы вытащить релевантную информацию. Следующим шагом является проверка поступивших событий по паттернам, если сработало определенное условие, которое было заложено в правилах реагирования, то сразу же будет сгенерировано уведомление и принятие контрмер на основании заложенных автоматизированных действий [3, 4].

Wazuh RESTful API – это интерфейс, который предоставляет возможность управлять различными механизмами Wazuh инфраструктуры. В основном это управление агентами и серверной конфигурацией, мониторинг и проверка статуса на наличие каких-либо ошибок, а также управление декодерами и правилами. Все это реализуется путем взаимодействия с Kibana, где можно построить свой Wazuh dashboard.

Wazuh cluster daemon – данный сервис используется для масштабирования серверов Wazuh, развертывая их в виде кластера. Такая конфигурация, в сочетании с балансировкой сетевой нагрузки, позволяет достичь максимальной доступности и сбалансированной нагрузки.

Filebeat – позволяет поставлять данные и уведомления об угрозах в Elasticsearch, которые были обработаны на стадии Analysis engine. Тем не менее, поставка данных проходит в реальном времени, обеспечивая балансировку при подключении к кластерам Elasticsearch.

Network Intrusion Detection System – разбирая более подробно, можно сказать, что служат они для обнаружения вторжения по сетевому каналу, отслеживая все проходящие сетевые пакеты. Тем самым, анализируя сетевой трафик на предмет известных видов сетевых атак, по данным от сенсоров, расположенных в определенных узлах. Главное отличие от Host-based IDS в том, что это больше направлено на область взаимодействия с сетевыми технологиями. Самыми популярными представителями из этой группы являются: Snort, Suricata и Zeek. Следует заметить, что многие прибегают к варианту интеграции Wazuh с Suricata/Snort для того чтобы можно было внедрить дополнительный участок потока данных для визуализации в готовый dashboard от Wazuh, в Kibana. По архитектуре размещения делится на: до фаервола, за фаерволом, на фаерволе и на каждом критически важном сервере [4].

Все зависит от конкретных целей мониторинга: хотим ли мы видеть все атаки; только те, которые проходят через отдельный фаервол или даже запущенные атаки против отдельных серверов. К примеру, размещая Network IDS/IPS решение до фаервола, откроется возможность видеть, какой же трафик пропускает через себя фаервол, а какой нет. Тем самым понимая, что именно он блокирует, а что пропускает. Контроль сети происходит с полным набором правил для того чтобы предотвращать все атаки, производимые против защищаемой сети (рисунок 3. IDS/IPS до фаервола) [5].

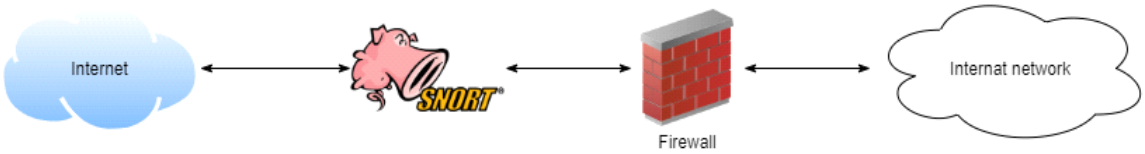

Рисунок 3 – IDS/IPS до фаервола

Следующее решение – поставить программное решение за фаерволом, это позволит увидеть только тот трафик, который проходит через фаервол. Появляется возможность сверять логи этого сенсора и внешнего в целях проверки работоспособности правил на фаерволе (рисунок 4 IDS/IPS за фаерволом).

Данный вид используется тогда, когда ресурсы организации ограничены, но понимание того, что же проходит через фаервол во внутреннюю сеть необходимо. Также как и в других примерах, присутствует возможность запуска программного решения с полным набором правил, но с одним нюансом: нужно направить сенсор на прослушивание внутреннего интерфейса. Это позволяет видеть, какой трафик уже успел пройти через базу правил фаервола во внутреннюю сеть или тот трафик, который генерируется внутри информационной сети организации (рисунок 5 IDS/IPS на фаерволе) [4, 5].

Еще одним вариантом является размещение на каждом одном конкретно важном сервере, который требует пристального мониторинга. Зачастую в программах IDS/IPS реализациях присутствует гибкая система кастомизации правил мониторинга, которые важны для каждого отдельно взятого сервера. К примеру, нет никакой необходимости включать правила, которые написаны для Mail сервера на Web сервере и т. д. (рисунок 6 IDS/IPS на каждом критически важном сервере) [4, 5].

Рано или поздно приходит решение о внедрении IDS решений, и на этом пути возникают множество проблем. Начиная от простых ложных срабатываний, обновления сигнатур, в отказоустойчивости программного решения, следования фиксированному бюджету, интеграции с другими производителями данных систем до превышения пропускной способности канала связи. Учитывая все эти проблемы, необходимо подходить к их решению правильно. В первую очередь необходимо исходить из предоставляемых ресурсов и от целей, которые необходимо реализовать. Для средней информационной сети организации вполне будет достаточно одного Wazuh, ведь он один покрывает сетевые угрозы, так и детектирует угрозы на самих хостах. Если же информационная сеть организация крупная, то следует внедрять уже отдельно Network от Host-based IDS, и ставить его в промежутке сети. Желательно использовать продукты, которые были выделены в Magic Quadrant Gartner: Palo Alto, Pfsense, FortiGate и т. д. А устанавливая две IDS на одном хосту, мы нагружаем весь канал передачи данных, что в свою очередь приводит к отрицательным последствиям.

Таким образом, исходя из анализа существующих программных решений, направленных на обеспечение безопасности автоматизированных систем, можно говорить о том, что программные IDS – это не решение всех проблем безопасности, это всего лишь малая часть того, что может повысить уровень защищенности в организации. Для того чтобы обеспечить защиту от полного перечня проблем безопасности автоматизированных систем, необходимо создать систему, в которой будут объединены все перечисленные выше программные решения, обеспечивающие безопасность автоматизированных систем под управлением нейронной сети, то есть создать Универсальную автоматизированную систему защищенного исполнения с применением нейросетевых алгоритмов управления системой.

Кроме того, важно отметить, что причисленные выше программные средства хорошо зарекомендовали себя на рынке. Open source решения стабильно развиваются своим комьюнити, где каждый разработчик вносит большой вклад в сферу информационной безопасности. Следует отметить, что нет какого-то одного правильного решения при выборе Host-based и Network IDS. Ведь каждая ситуация и инфраструктура по-своему индивидуальна. Но за счет внедрения данных систем, процент защищенности инфраструктуры резко повышается. Главная задача – не навредить бизнесу, правильно подойти к проектированию, заранее выявлять возможные проблемы, пытаться не прерывать данный процесс развития безопасности, оптимизировать расходы в заложенный бюджет и с каждым разом совершенствовать свой защищаемый периметр.

Также очень важно отметить, что «в настоящее время исследовательские группы во всем мире начали использовать алгоритмы машинного обучения и глубокого обучения для применения в системах IDS для повышения эффективности и точности обнаружения сигналов атаки в сети. Для проведения исследований и машинного обучения используются такие наборы данных, как KDD Cup 1999,

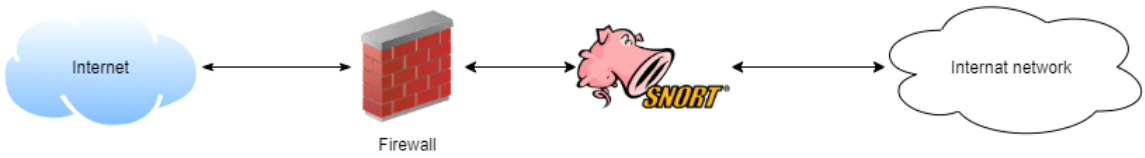

Рисунок 4 – IDS/IPS за фаерволом

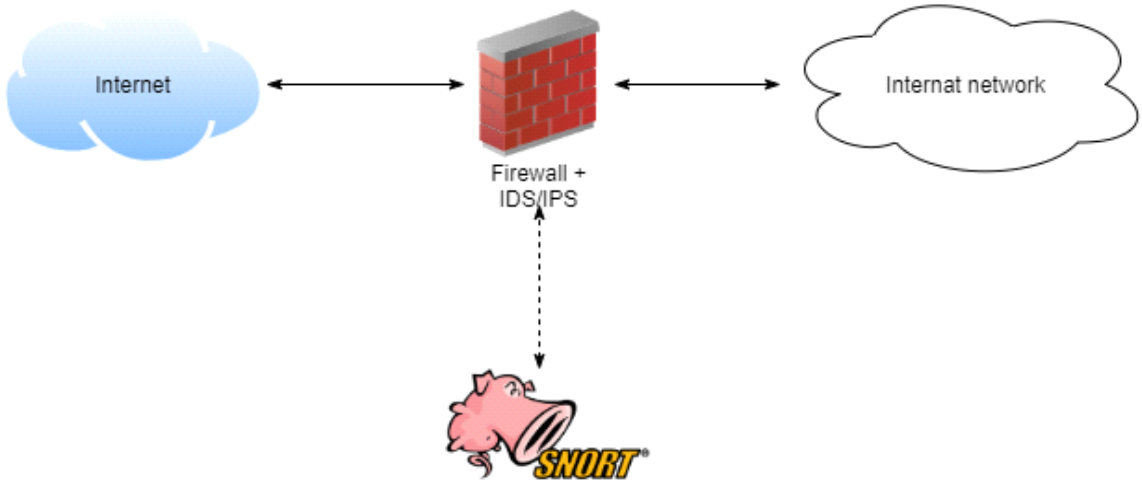

Рисунок 5 – IDS/IPS на фаерволе

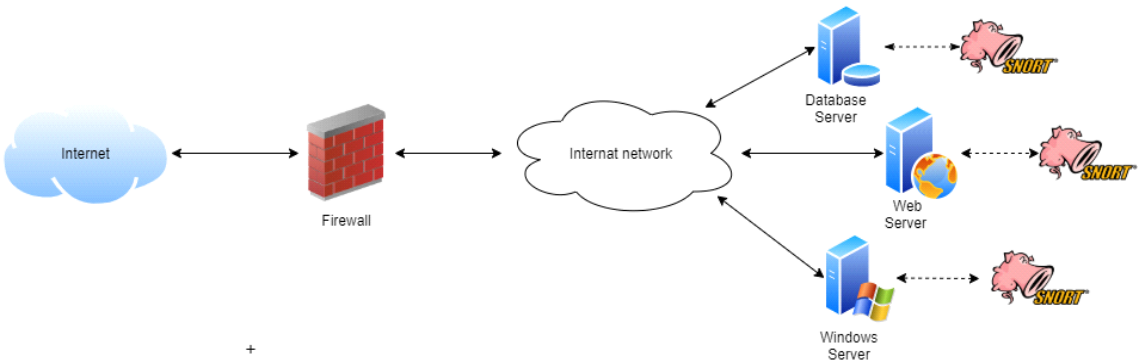

Рисунок 6 – IDS/IPS на каждом критически важном сервере

NSK KDD, CIDDS-001 [5–7]. Соответственно, свойства пакетов будут проанализированы на основе спецификаций наборов данных, а затем переданы в функцию обработки для оценки безопасности или опасности пакетов» [7].

«Другими словами, система мониторинга и безопасности должна изучать и контролировать работу информационной системы в нормальных условиях для записи эксплуатационных параметров, что является основой для обнаружения отклонений в будущем. Кроме того, в настоящее время, наблюдается быстрое развитие Интернета вещей, который облегчает жизнь, при этом улучшается информационная взаимосвязь людей. Тем не менее, это развитие также приводит к увеличению угроз кибербезопасности. Как только киберпространство достигает своего полного состояния, оно становится целью атак хакеров в различных формах. Таким образом, встроенная система IDS/IPS станет довольно пассивной с новыми типами атак. Новые решения необходимы для решения этой проблемы» [7].

В качестве примера можно привести модель Интеллектуальной системы IDS, предложенную группой исследователей на конференции, проводимой Агентством перспективных научных исследований (АПНИ) в 2020 г. Обобщенная модель данной системы представлена на рисунке 7.

Очень важно отметить, что алгоритмы машинного обучения для создания IDS/IPS систем в настоящее время очень перспективны и широко используются, так как при помощи систем, имеющих в своем составе модули машинного обучения, обладающие функциями самообучения и высоко интеллектуального рассуждения, можно получать и проверять по специальным алгоритмам наборы данных в случаях, когда новые атаки основаны на собранных наборах данных о системах защиты.

Таким образом, с учетом сказанного выше, авторами статьи на основе Универсальной среды проектирования АСЗИ (рисунок 8) [8] предлагается вариант расширяемой модульной платформы для SOC (РМП SOC), прототипом которой является известная модель Интеллектуальной системы IDS. Данный вариант расширяемой модульной платформы для SOC (РМП SOC) способен обеспечить безопасность автоматизированных систем под управлением алгоритмов машинного обучения нейронной сети, т. е. фактически является универсальной АСЗИ, управляемой соответствующими нейросетевыми алгоритмами.

Особенностью таких нейросетевых алгоритмов является их функциональная направленность на формирование баз данных, анализ процесса построения архитектуры баз данных на всех уровнях (нижний, средний, верхний), формирование облачных программно-аппаратных систем, способных в режиме реального времени анализировать ситуацию, и на основе проведенного анализа формировать сигнатуры для устранения инцидентов информационной безопасности автоматизированных систем.

Таким образом, для выполнения первого подхода к программно-аппаратной реализации обеспечения кибербезопасности автоматизированных систем – «Использование Host-based и Network IDS решений, в том числе SIEM и DLP систем», необходимо разработать концептуальную модель РМП SOC которая по своей сути будет соответствовать автоматизированной системе защищённого исполнения (АИСЗИ).

Рассмотрим концептуальную модель РМП SOC на примере Use case диаграммы (рисунок 9), в которой в роли актеров, взаимодействующих с системой с помощью так называемых вариантов использования, выступают три участника: персонал, администратор и скрипт.

При этом каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. В процессе проектирования системы было определено, что администратор будет выполнять как реагирование на инцидент, так и устранение уязвимостей. Скрипт, в свою очередь будет выполнять запуск сканера уязвимостей в определенное время. Где впоследствии выдаст отчет администратору по проведенной работе. Задача персонала в данном процессе проста: необходимо выполнять свои прямые обязанности, при этом за действиями персонала будет осуществляться мониторинг с целью выявления аномалий и нарушений установленных политик безопасности системы. На рисунке 9 представлена концептуальная модель в формате Use case.

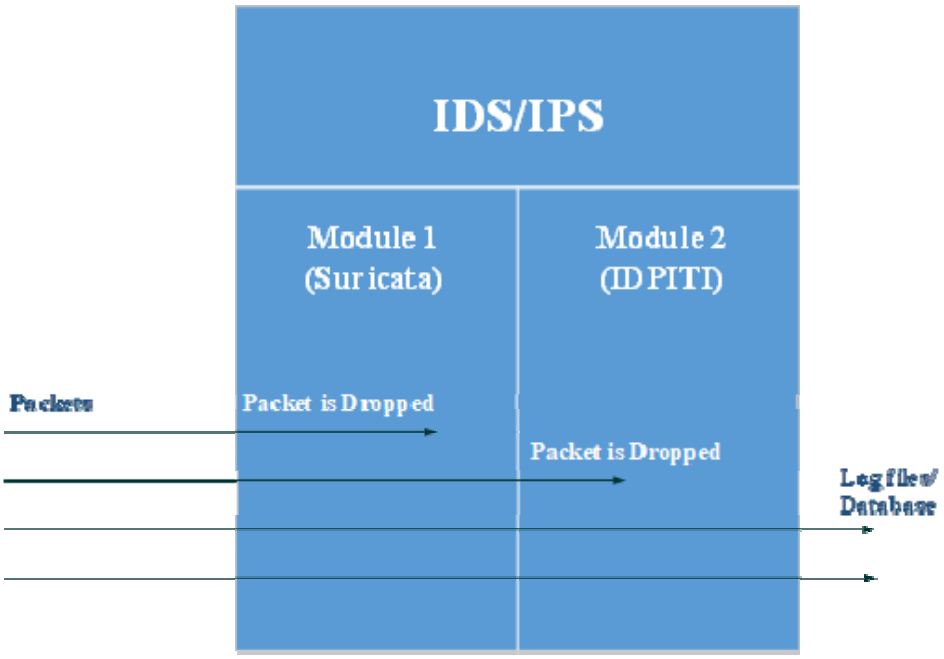

Рисунок 7 – Предлагаемая модель системы IDS/IPS АПНИ

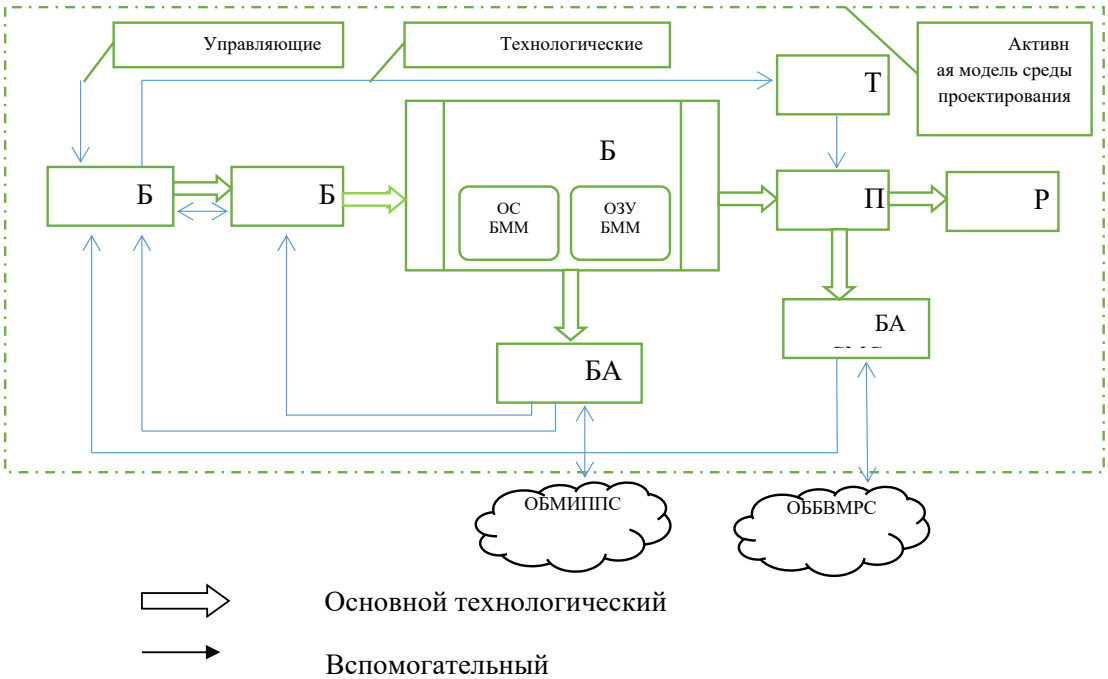

Рисунок 8 – Функциональная структура модели универсальной среды проектирования АСЗИ : БЗРПП – Блок задания режимов процессов проектирования; БУП – Блок управления проектированием; БАСПП – Блок анализа состояний процесса проектирования

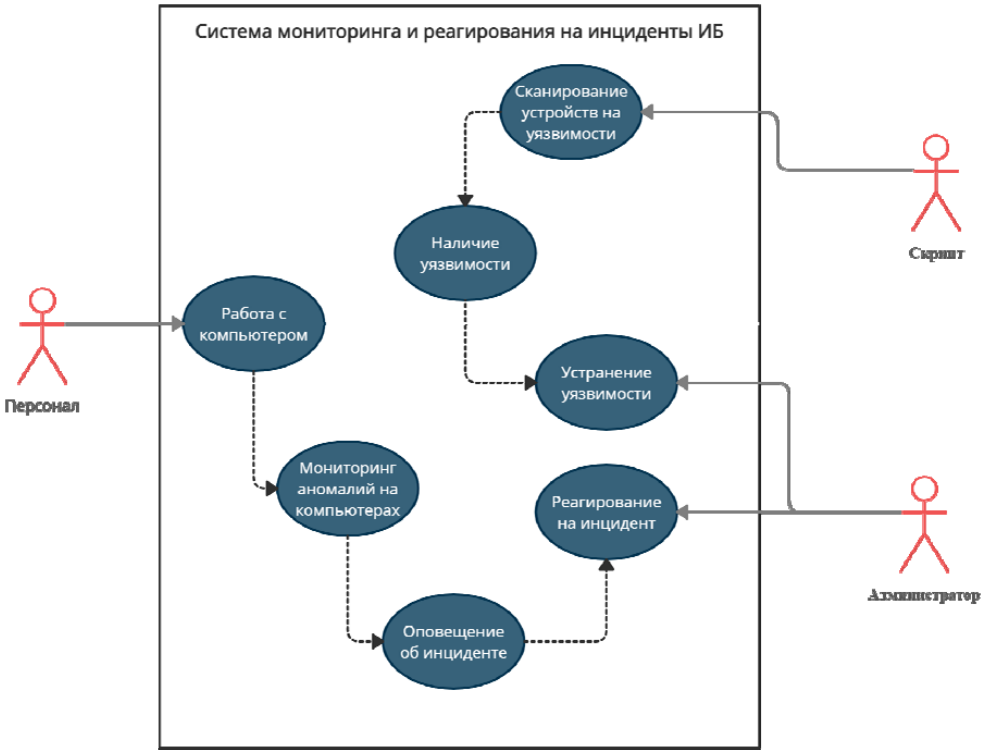

Рисунок 9 – Концептуальная модель (Use case)

Следующим этапом для нашей системы опишем процесс реагирования на инцидент информационной безопасности на примере диаграммы IDEF0.

Для нашей РМП SOC взаимодействие между функциями в IDEF0 представляется в виде дуги, которая отображает поток данных или материалов, поступающих с выхода одной функции на вход другой. В зависимости от того, с какой стороной блока связан поток, его называют соответственно: входным, выходным, управляющим и механизмом.

В контексте процесса реагирования на инцидент информационной безопасности под входным потоком понимается следующее: поставка данных от Data Shippers, обнаружение инцидента и анализ инцидента ИБ. К выходному потоку относятся: расследование инцидента, устранение инцидента, восстановление информационной системы, агрегация новых знаний об угрозе и подробный отчет об инциденте и его устранении. Под механизмом же понимается, как сам администратор системы, так и его программный инструментарий. Конечным этапом является управляющий поток: план реагирования на инцидент, документация к инструментарию, политика безопасности и матрица MITRE (рисунок 10).

Декомпозиция данного процесса включает в себя несколько важных подпроцессов.

Первый из них это фильтрация данных в Logstash и Wazuh Manager. В нашем случае входным потоком в систему является поставка данных от Data Shippers. Впоследствии поступающая информация фильтруется по установленным критериям. После чего, фильтрованная информация передается во второй процесс – хранение данных в Elasticsearch. Входным потоком в систему является обнаружение инцидента в поступающей информации на базе сигнатурных данных.

Процесс аналитики и визуализации данных в Kibana – это третий процесс. На этом этапе происходит анализ инцидента информационной безопасности, с применением различных инструментов: машинного обучения, графов и изучения генерируемых отчетов.

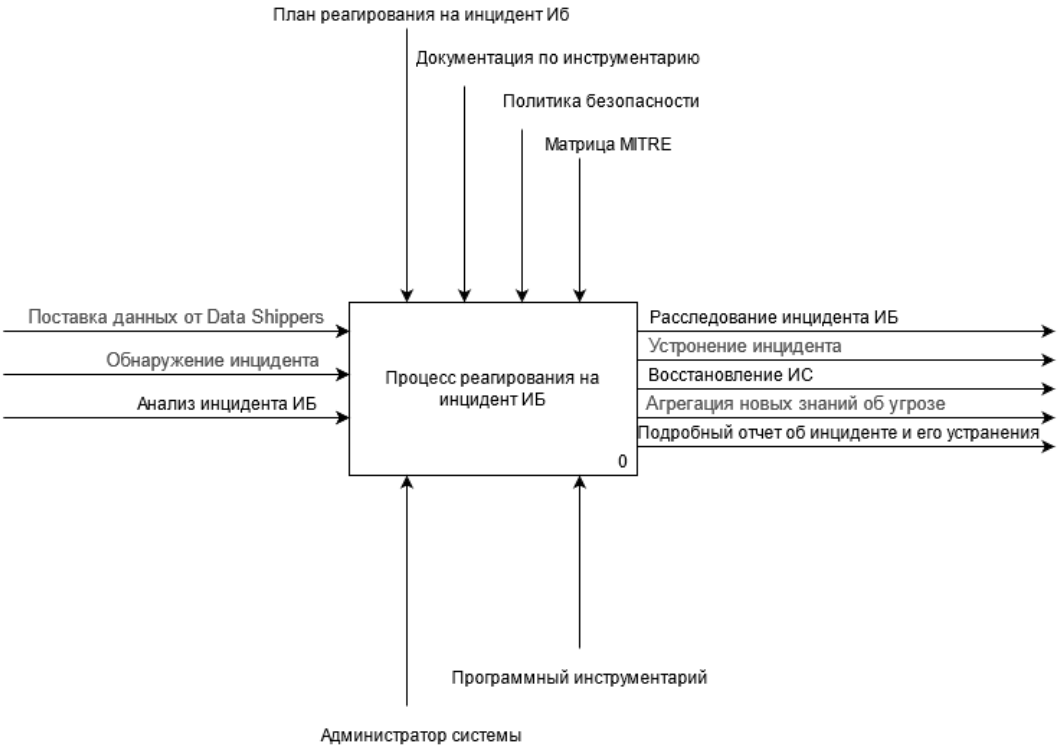

Рисунок 10 – Процесс реагирования на инцидент информационной безопасности (IDEF0)

Самый важный, четвертый процесс – реагирование на инцидент. Опираясь на имеющиеся данные, полученные системой после завершения процесса аналитики и визуализации, пользователи системы проводят расследование инцидента, используя при этом технические, правовые и организационные меры и мероприятия. Одновременно устраняя уязвимости, из-за которых был спровоцирован инцидент, при этом одновременно должна восстановиться работоспособность информационной системы.

Пятый процесс отвечает за составление отчетности. В отчетах описываются все подробности об инциденте и мероприятия, принятые для его устранения. Немаловажным шагом является агрегация новых знаний, полученных в результате изучения атаки на инфраструктуру автоматизированной системы. Подробнее декомпозицию процесса реагирования на инциденты информационной безопасности можно рассмотреть на рисунке 11.

Следующим этапом для нашей системы рассмотрим описание алгоритма мониторинга аномальной активности на примере диаграммы последовательностей. Для нашей системы диаграмма последовательностей – это графическая двухмерная модель взаимодействия объектов, построенная на основе определенного сценария. При проектировании системы был выделен процесс мониторинга аномальных действий на рабочих станциях (персональных компьютерах, ЭВМ). Диаграмма последовательностей для рассматриваемой АИСЗИ представлена рисунке 12.

В роли объектов представленных на диаграмме последовательностей выступает набор программного обеспечения, элементы которого при взаимодействии демонстрируют возможность отслеживания потока событий с вредоносными действиями. Стартом выявления инцидентов ИБ является деятельность персонала, который совершает свою рутинную работу, генерируя лог файлы в журналы событий операционной системы.

Рассмотрим взаимодействие набора программного обеспечения более детально:

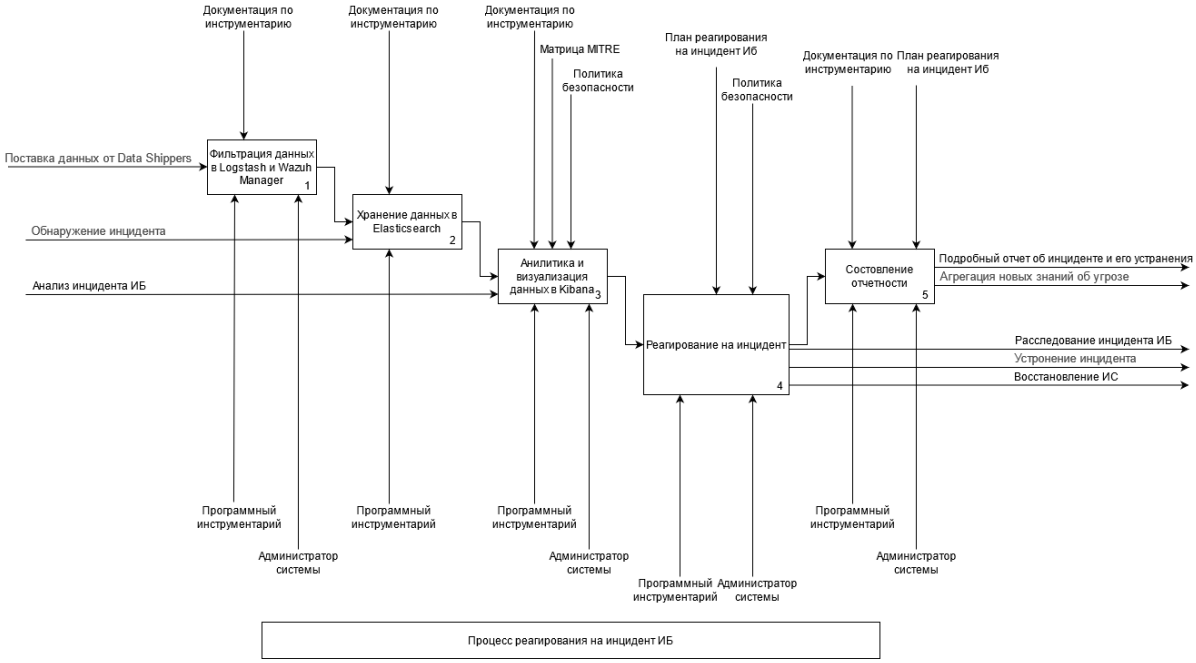

Рисунок 11 – Декомпозиция процесса реагирования на инцидент информационной безопасности (IDEF0)

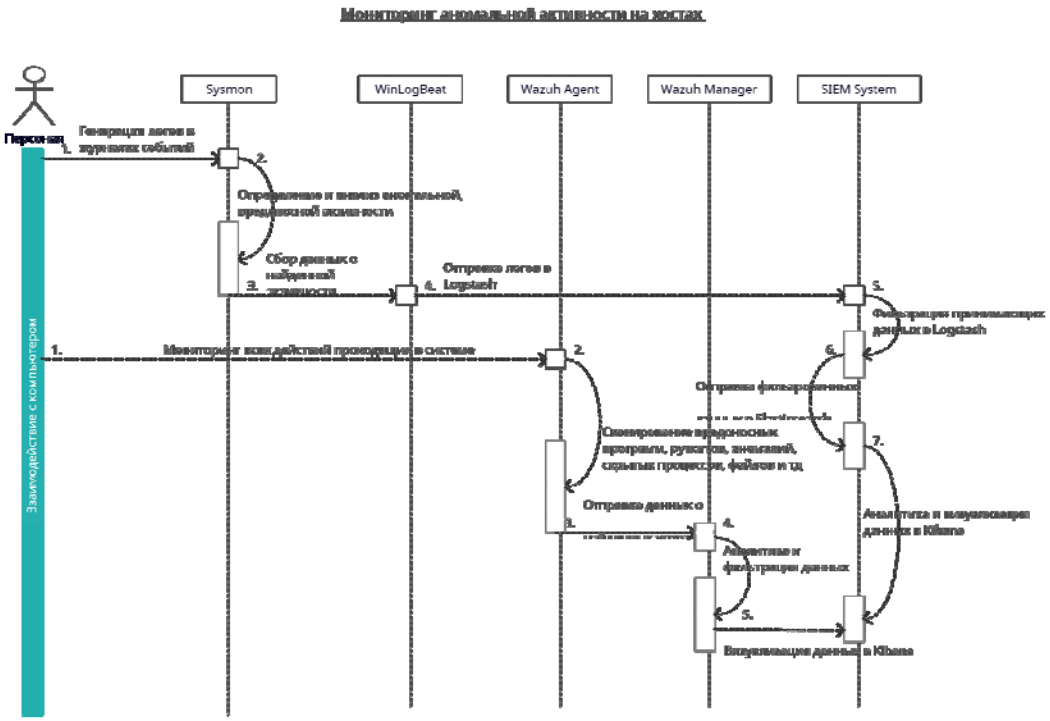

Рисунок 12 – Диаграмма последовательности

Sysmon – это системная служба Windows, установленная в операционной системе отслеживает и регистрирует активности системы в журнале событий. Также предоставляет информацию о создании процессов, сетевых подключениях и изменениях времени создания файлов. На базе собранных событий Sysmon может анализировать посигнатурно каждое событие, сопоставляя их с открытой базой знаний об кибератаках MITRE ATT&CK.

Все найденные аномальные события отправляются на удаленный сервер инструментом Winlogbeat, где установлена SIEM система. За прием и фильтрацию данных в SIEM отвечает Logstash, где изначально были прописаны правила и скоуп (архитектура) нужных данных. После чего logstash отправляет их в базу данных elasticsearch. В свою очередь Kibana, путем запросов на эндпоинты elasticsearch, анализирует и визуализирует полученную информацию.

Рассмотрим описание алгоритма устранения уязвимостей на примере диаграммы состояний.

В нашем случае диаграмма состояний показывает все возможные состояния, в которых может находиться объект, а также процесс смены состояний в результате внешнего влияния. Основными элементами данной диаграммы состояний являются «состояние» и «переход».

В контексте описания алгоритма устранения уязвимостей информационной безопасности системы, стартовой точкой является автоматический запуск сканера уязвимостей. Данный сканер принимает состояние сканирования информационной системы на наличие уязвимостей. По завершении процесса сканирования, сканером выдаются варианты принятия решения, нужно ли устранять найденные уязвимости или нет. В случае ненадобности, сканер просто завершает свою деятельность. В альтернативном случае он переходит в состояние отправки оповещения администратору.

Отправленное сканером оповещение является начальной точкой в процессе устранения уязвимостей для администратора АИСЗИ. Впоследствии процесс принимает состояние поиска решения, если же уязвимость устраняется наличием обновления программного обеспечения, то процесс переходит в состояние тестирования системы. Если же обновление не нарушит другие процессы информационной системы, то оно устанавливается, и данной уязвимости присваивается статус – устранено. Подробнее Алгоритм устранения уязвимостей (диаграмму состояний) можно рассмотреть на рисунке 13.

Далее рассмотрим общее взаимодействие компонентов внутри платформы по мониторингу на примере диаграммы классов. Для нашей системы диаграмма классов имеет два вида: статический и аналитический.

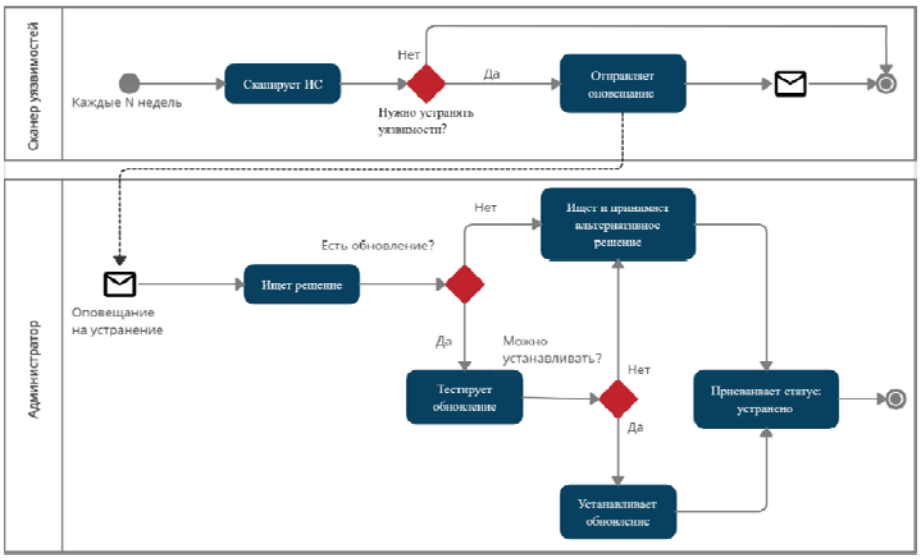

Рисунок 13 – Алгоритм устранения уязвимостей (диаграмма состояния)

Статический вид показывает и рассматривает логические взаимосвязи классов между собой, аналитический вид, в свою очередь, рассматривает общий вид и взаимосвязи классов, входящих в систему.

В описании класса необходимо придерживаться четкой структуры. Верхняя часть отвечает за название класса, оно должно быть уникальным. Средняя часть описывает атрибуты или свойства, может быть не заполнена. Нижний слой отвечает за название операций или услуг, предоставляемыми объектами этого класса. В данном случае предоставляются восемь классов, которые взаимодействуют друг с другом.

Класс Wazuh отвечает за приемку логов от своих же агентов, которые находятся на клиентских хостах, проводя мониторинг угроз в реальном времени. Также проводит операцию отправки принимаемых логов в Elasticsearch.

Класс Elasticsearch принимает логи от поставщиков данных: filebeat, индексируя их в своей базе данных. Данный процесс приводит ко второй операции – хранению проиндексированных данных. В целях аналитики и дальнейшей фильтрации имеет опцию отправки данных на другие эндпоинты.

Класс Logstash имеет три операции: сбор данных от сторонних приложений, которые отправляют данные на его интерфейс; фильтрация принимаемых данных в целях отсеивания ненужной информации; отправка данных дальше по цепочке в Kibana для аналитики.

Класс Filebeat имеет две опции – сбор системных логов и их отправка на эндпоинты Elasticsearch и Logstash.

Класс Nessus занимается тремя главными операциями: сканированием сети на наличие уязвимости; генерацией отчетности по предыдущему шагу; хранением отчетов.

Класс VulnWhisPerer собирает все отчеты от сканера Nessus и позволяет их агрегировать и отправлять на интерфейс Logstash.

Класс Elastalert детектирует угрозы путем сигнатурного анализа в агрегированной информации в Elasticsearch, затем может оповещать на сторонние ресурсы.

Класс Kibana занимается аналитикой и визуализацией всех данных, которые поставляют все перечисленные выше инструменты. Подробнее общее взаимодействие компонентов системы показано на рисунке 14.

Кроме того, нельзя забывать о том, что для любой автоматизированной системы необходимо определение требований к аппаратно-программному обеспечению.

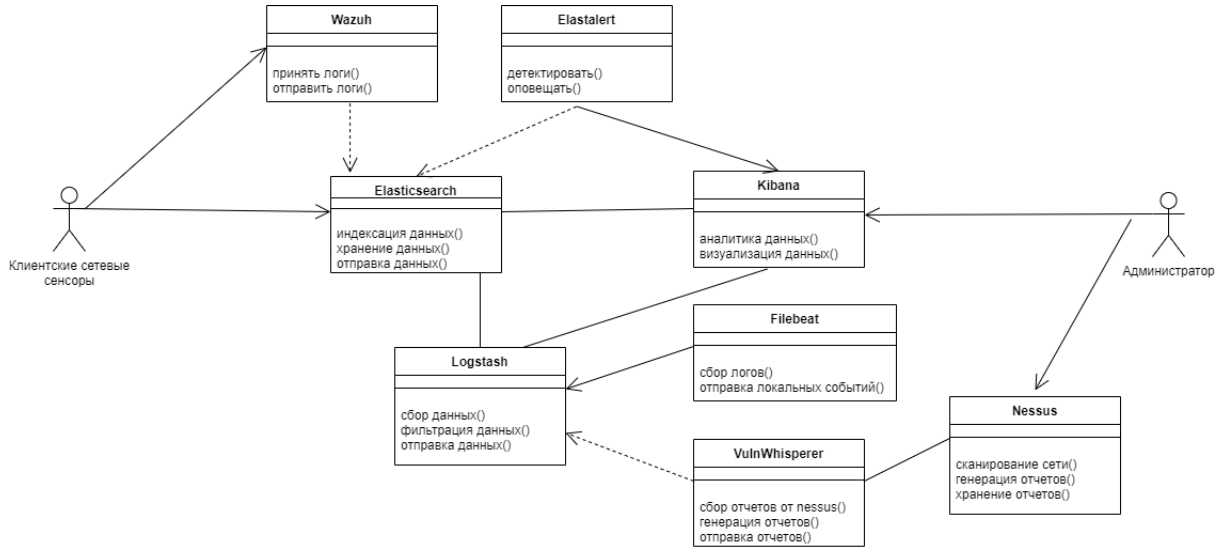

Рисунок 14 – Общее взаимодействие компонентов платформы

Для рассматриваемой системы были определены минимальные требования к программно-аппаратному обеспечению. Для развертывания АИСЗИ необходимо иметь сервер с Unix подобной операционной системой, для системы используется версия не ниже Ubuntu 18.04 LTS. Для того чтобы обеспечить минимальную работоспособность системы, объем оперативной памяти необходимо установить минимум от 8 Гб. Так как серверы разворачиваются в виртуальной среде, то необходимо иметь виртуальный процессор, где минимально должны присутствовать два ядра.

Для нормальной работы системы в сети необходимо перевести в открытый режим работы открытые следующие приведенные порты: 1514(Wazuh), 5044(Logstash), 5601(Kibana), 9200(Elasticsearch), 8060(Praeco), 8834(Nessus) и 22(SSH).

Для клиентских эндпоинтов необходимо иметь операционную систему Windows 10 и Ubuntu 18.04 LTS, где в минимальной комплектации операционной системы у процессора должно быть 2 ядра и 4 Гб оперативной памяти. Также необходимо открыть порты RDP и SSH. Более подробно требования к программно-аппаратному обеспечению системы показаны на рисунке 15.

Поскольку наша система выполняет функции Security Operations Center, для нее разработан процесс взаимодействия подразделений Security Operations Center (SOC).

Согласно процессу взаимодействия подразделений SOC в центре мониторинга и реагирования на инциденты информационной безопасности присутствует ряд подразделений, которые выполняют определенные задачи. Для процесса были определены основные критически важные позиции/направления: Alert Analysts, Incident Responders, Malware Analysts, Forensics, DevSecOps, Threat Intelligence, Threat Hunting, SOC Manager и CISO.

Данные направления делятся на уровни или линии контролируемых точек в инфраструктуре потребителя услуг. Уровней всего три: первый, второй и третий. К первому уровню относится направление Alert Analysts. Это подразделение отвечает за мониторинг инфраструктуры в режиме реального времени. Им приходится работать с большим потоком данных, который необходимо тщательно анализировать, писать правила корреляций для SIEM системы, и заводить тикеты на аномальную активность, проверяя каждое событие на факт наличия инцидента в сети. Впоследствии заведенные тикеты повторно анализируются более компетентными и вышестоящими по уровню аналитиками. При успешном подтверждении о нахождении инцидента, тикет отправляется в систему подразделения Incident Response.

<table>
<tr><th>Azure/Google Clouds</th><th>Requirements</th><th>Operating System</th><th>Open Ports</th><th>Services</th></tr>
<tr><td rowspan="8">Server</td><td rowspan="4">2 vcpus</td><td rowspan="8">Ubuntu 18.04 LTS</td><td>Wazuh => TCP:1514</td><td>ELK Stack</td></tr>
<tr><td>Logstash => TCP:5044</td><td>FileBeat</td></tr>
<tr><td>Kibana => 5601</td><td>Nessus</td></tr>
<tr><td rowspan="4">8 GB memory</td><td>Elasticsearch => 9200</td><td>VulnWhisperer</td></tr>
<tr><td>Praeco => TCP:8080</td><td>Elastalert</td></tr>
<tr><td>Nessus => TCP:8834</td><td>Praeco</td></tr>
<tr><td>SSH => TCP:22</td><td>Wazuh</td></tr>
<tr><td rowspan="8">Client endpoints</td><td rowspan="2">2 vcpus</td><td rowspan="4">Windows 10</td><td rowspan="4">RDP => TCP:3389</td><td>FileBeat</td></tr>
<tr><td rowspan="2">4 GB memory</td><td rowspan="2">Wazuh-agent</td></tr>
<tr><td rowspan="2">2 vcpus</td><td rowspan="4">Ubuntu 18.04 LTS</td><td rowspan="4">SSH => TCP:22</td><td>FileBeat</td></tr>
<tr><td rowspan="2">4 GB memory</td><td rowspan="2">Wazuh-agent</td></tr>
</table>

Рисунок 15 – Требования к программно-аппаратному обеспечению

Подразделение Incident Response находится на втором уровне. Они отвечают за проработку найденного инцидента путем локального расследования и исследования найденных артефактов. При возникновении критической ситуации данные специалисты первыми выезжают на место произошедшего инцидента, пытаясь прервать его и устранить. Отдел Malware Analysts находится в плотном сотрудничестве с данным отделом, позволяя оперативно проводить исследования артефактов, выдавая необходимую информации о вредоносном программном обеспечении – малвари и рекомендации по ее устранению или ослаблению.

На третьем уровне располагаются: Forensics, Threat Hunting и Threat Intelligence. Направление Forensics отвечает за криминалистику, которое сопровождается в целях нахождения пути заражения, идентификаторов группировок, сбора цифровых улик для дальнейшего предоставления их в судебном процессе.

Threat Hunting – довольно новое направление, которое отвечает за непрерывный поиск угроз. Данный процесс построен на техниках, применяемых в Penetration Testing и Red Teaming, но значительно отличается тем, что сам процесс направлен на не эксплуатацию уязвимости дальнейшего ее развития, а лишь на факт ее нахождения без предоставления юридической основы. Все полученные данные на входе процесса собираются в общую базу знаний, где каждый отдел ее обогащает.

Еще один довольно-таки значимый отдел это Threat Intelligence, тут специалисты не имеют каких-либо ограничений в изучении сферы киберугроз. В основном, они занимаются исследовательской деятельностью, где находят и классифицируют новые угрозы безопасности. Изучаются известные атаки любых хакерских группировок, выявляют взаимосвязи, прогнозируют развитие новых тенденций в области безопасности, составляют общедоступные публикации о найденных аномалиях и артефактах и т. д.

Нужно отметить, что отделов бывает огромное количество и необходимо их между собой координировать. Так вот, координацией действий, учетом рисков, бюджета и прочими менеджерскими процессами занимается главный SOC Manager. В свою очередь, он находится под подчинением владельца SOC'а или Chief Information Security Officer (CISO). Подробнее процесс взаимодействия подразделений Security Operations Center можно показан на рисунке 16 [9].

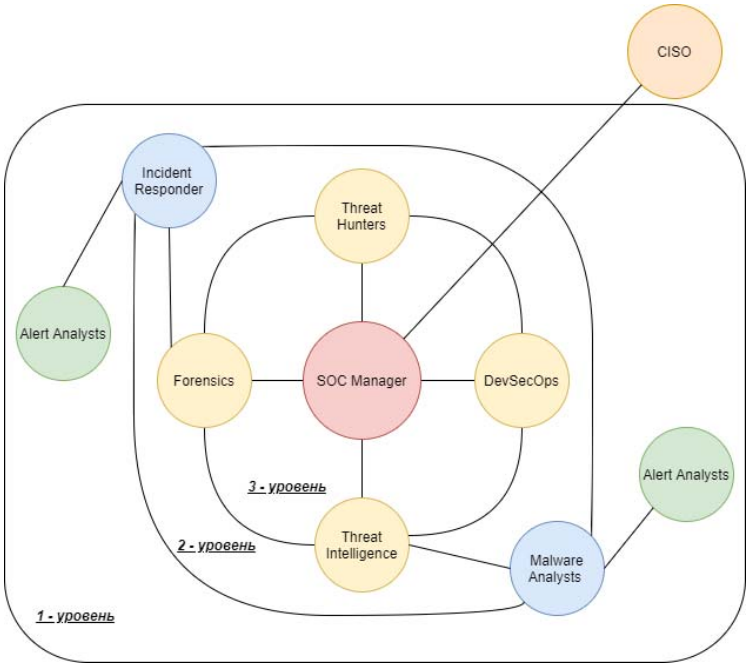

Рисунок 16 – Схема взаимодействия подразделений SOC

Заключение. Обобщая все сказанное выше, можно констатировать, что на сегодняшний день попрежнему актуальна проблема защиты информации автоматизированных систем, несмотря на большое количество разработанных средств ее защиты, программного обеспечения, методологий обнаружения атак.

Введение нового понятия «нейросетевые алгоритмы управления системами информационной безопасности», область применения которого тематически связана с программно-аппаратными средствами обеспечения информационной безопасности, существенно дополняет определение таких понятий, как искусственная нейронная сеть [10] и нейронная сеть [11].

В свою очередь, широкое применение предложенного подхода к формированию нейросетевых алгоритмов управления позволит весьма эффективно использовать имеющееся программно-аппаратные средства для обеспечения максимально-возможной защиты автоматизированных систем от угроз информационной безопасности.

Применение case средств позволило разработать для РМП SOC необходимую Концептуальная модель, описывающую: 1 – процесс реагирования на инцидент информационной безопасности на примере диаграммы IDEF0; 2 – декомпозицию процесса реагирования на инцидент информационной безопасности (IDEF0); 3 – алгоритм мониторинга аномальной активности на примере диаграммы последовательностей; 4 – алгоритм устранения уязвимостей на примере диаграмма состояний системы; 5 – общее взаимодействие компонентов внутри платформы по мониторингу на примере диаграммы классов.

Полученные результаты применения case средств для решения поставленных задач является наглядным примером успешного использования данной технологии, которая может быть рекомендована для решения других подобных задач в сфере информационной безопасности.

Поступила: 28.01.22; рецензирована: 09.02.22; принята: 15.02.22.

Литература

1. Wazuh Components // wazuh.com URL: https://documentation.wazuh.com/ current/getting-started/ components/#components (дата обращения: 05.03.2021)
2. Официальная документация по архитектуре Wazuh Agent. URL: https:// documentation.wazuh.com/4.0/ getting-started/components/wazuh_agent. html#wazuh-agent (дата обращения: 10.03.2021).
3. Официальная документация по архитектуре Wazuh Server. URL: https://documentation.wazuh.com/4.0/ getting-started/components/wazuh_server. html#wazuh-server (дата обращения: 15.03.2021).
4. Материал из Национальной библиотеки им. Н.Э. Баумана. URL: https:// ru.bmstu.wiki/HIDS_(Host-Based_Intrusion_Detection_System) (дата обращения: 17.03.2021).
5. An Introduction to Snort: A Lightweight Intrusion Detection System. URL: https://www.informit.com/articles/article.aspx?p=21778 HYPERLINK «https:// www.informit.com/articles/article.aspx?p=21778&seqNum=9»& (дата обращения: 05.04.2021).
6. *Geier E.* Intro to Next Generation Firewalls / E. Geier. 06 September, 2011.
7. *Ле К.М.* Интегрированная IDS/IPS модель между открытым источником с улучшением машинного обучения / К.М. Ле, Х.А. Фан, А.Ч. Нгуен, Ч.Т. Нгуен // Результаты прикладных и поисковых научных исследований в сфере естествознания и технологий: Сб. науч. тр. по матер. межд. научно-практич. конф. 27 декабря 2019 г. Белгород: ООО Агентство перспективных научных исследований (АПНИ), 2019. С. 81–87. URL: https://apni.ru/article/152-integrirovannaya-idsips-model-mezhdu-otkritim/ (дата обращения: 10.08.2021).
8. *Корякин С.В.* Разработка универсальной среды проектирования автоматизированных систем защищенного исполнения / С.В. Корякин // Проблемы автоматики и управления. Бишкек: ИМА НАН КР 021, No2 (41) (38). С. 40–55.
9. Gregory Jarpey Security Operations Center Guidebook: A Practical Guide for a Successful SOC. 1st edition. Изд. Butterworth-Heinemann, 2017. 206 с.
10. Нейросетевые алгоритмы распознавания объектов. URL: https://nppame.com/napravlenija-razrabotok/ programmnoe-obespechenie/ nejrosetevye-algoritmy-raspoznavaniya/] (дата обращения: 05.06.2021).
11. Определение понятия Нейронная сеть в энциклопедии Wikipedia. URL: https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C (дата обращения: 15.06.2021).