УДК 004.087
DOI: 10.36979/1694-500X-2023-23-4-67-78

ОБЛАЧНАЯ СИСТЕМА МЕЖСЕТЕВЫХ ЭКРАНОВ

С.В. Корякин, Ю.С. Корякина

Аннотация. Рассматривается прототип облачной системы межсетевых экранов/брандмауэров (ОСМЭ), которая своим функционалом должна обеспечивать защиту автоматизированных систем от киберугроз. В работе предложен подход к формированию конфигурации функциональных возможностей межсетевых экранов/брандмауэров, объединённых аппаратным облаком под управлением нейронной сети. Предложена архитектура ОСМЭ, функционирующая по принципу аппаратно-ориентированных сетей. Предложен алгоритм работы и архитектура аппаратного межсетевого экрана/брандмауэра, функционирующего по принципам работы аппаратно-ориентируемой сети. Дано определение аппаратной облачной сети. Дано определение аппаратного облака.

Ключевые слова: нейронная сеть; программно-аппаратное ядро; технологии проектирования; облачные библиотеки; системы и среды передачи информации; угрозы информационной безопасности автоматизированных систем; нейросетевые алгоритмы управления.

ТАРМАКТАР АРАЛЫК ЭКРАНДАРДЫН БУЛУТ СИСТЕМАСЫ

С.В. Корякин, Ю.С. Корякина

Аннотация. Макалада тармактар аралык экрандардын/брандмауэрлердин булут системасынын прототиби (ББТ) каралат, ал өзүнүн функционалдуулугу менен автоматташтырылган системаларды киберкоркунучтардан коргоону камсыз кылууга тийиш. Бул эмгекте нейрон тармагы башкарган аппараттык булут менен бириктирилген тармактар аралык экрандардын/брандмауэрлердин функционалдык мүмкүнчүлүктөрүнүн конфигурациясын калыптандыруу ыкмасы сунушталат. Аппараттык-багытталган тармактар принцибинде иштеген тармактар аралык экрандардын/брандмауэрлердин булут системасынын архитектурасы сунушталган. Аппараттык багыттагы тармактын иштөө принциптери боюнча иштеген аппараттык тармактар аралык экрандардын/брандмауэрлердин иштөө алгоритми жана архитектурасы сунушталды. Аппараттык булут тармагынын аныктамасы берилген. Аппараттык булуттун аныктамасы берилген.

Түйүндүү сөздөр: нейрон тармагы; программалык-аппараттык өзөк; дизайн технологиялары; булут китепканалары; маалымат берүү системалары жана чөйрөлөр; автоматташтырылган системалардын маалыматтык коопсуздугуна коркунучтар; нейро-тармактык башкаруу алгоритмдери.

CLOUD FIREWALL SYSTEM

S.V. Koryakin, Yu.S. Koryakina

Abstract. This article is about a prototype cloud-based firewall system(CBFS), which have to protect automated systems from cyber threats. This article is about a proposed approach to configuring the functionality of firewalls, united by a hardware cloud controlled by a neural network. Also, the architecture of cloud-based firewall systems is proposed, functioning according to the principles of hardware-oriented networks. An algorithm of work and the architecture of a hardware firewall are proposed. It functions according to the principles of device-oriented networking. The definition of a hardware cloud network is given. The definition of a hardware cloud is given.

Keywords: neural network; hardware and software core; design technologies; cloud libraries; information transmission systems and environments; threats to information security of automated systems; neuro-network control algorithms.

На сегодняшний день известно, что мошенничество с данными и кибератаки являются четвертым и пятым глобальными рисками, с которыми сталкивается каждая автоматизированная система или организация. По своей значимости эти риски приравниваются к стихийным бедствиям. Также достоверным фактом является то, что атаки хакеров во всём мире происходят каждые 11 секунд [1].

С ростом кибератак возникает потребность в обеспечении безопасности данных. Так как киберпреступники с каждым днем ищут уязвимости, анализируют и создают способы и инструменты для обхождения контуров защиты автоматизированных систем (АС), специалисты по информационной безопасности должны совершенствовать подсистемы защиты и быть всегда на шаг впереди, поэтому выбранная тематика на сегодняшний день является актуальной.

Рассмотрим один из методов, с помощью которого будет создан новый, более функционально эффективный метод борьбы с киберпреступностью.

Межсетевые экраны (МСЭ) – это аппаратные и программные меры для предотвращения негативных воздействий извне, то есть они представляют собой локальное (однокомпонентное) или функционально-распределенное средство (комплекс), реализующее контроль за информацией, поступающей в автоматизированную систему и выходящей из нее, и обеспечивающее защиту автоматизированной системы посредством фильтрации информации, то есть ее анализа по совокупности критериев и принятия решения о ее распространении в автоматизированной системе. В тексте будут использованы понятия межсетевой экран (МСЭ), брандмауэр, firewall, как эквивалентные [2].

Файрволл работает как фильтр: из всего потока трафика просеивается только разрешенный. Это первая линия защитных укреплений между внутренними сетями и внешними, такими как интернет. Технология применяется уже на протяжении 25 лет.

Необходимость в межсетевых экранах возникла, когда стало понятно, что принцип полной связности сетей больше не работает. С распространением ПК и интернета возникла необходимость отделять внутренние сети от небезопасных внешних, чтобы уберечься от злоумышленников и защитить компьютер от взлома [2].

Для защиты могут быть применены аппаратный межсетевой экран – это может быть отдельное устройство или часть маршрутизатора, и его альтернативный вариант – программный межсетевой экран, примером может служить файрволл, встроенный в Windows (рисунок 1).

«Существует два основных способа создания наборов правил межсетевого экрана: «включающий» и «исключающий»» [3]. Исключающий межсетевой экран позволяет прохождение всего трафика, за исключением трафика, соответствующего набору правил. Включающий межсетевой экран действует прямо противоположным образом, он пропускает только трафик, соответствующий правилам, и блокирует все остальное.

Включающий межсетевой экран обеспечивает гораздо большую степень контроля исходящего трафика. «Поэтому он является лучшим выбором для систем, предоставляющих сервисы в сети

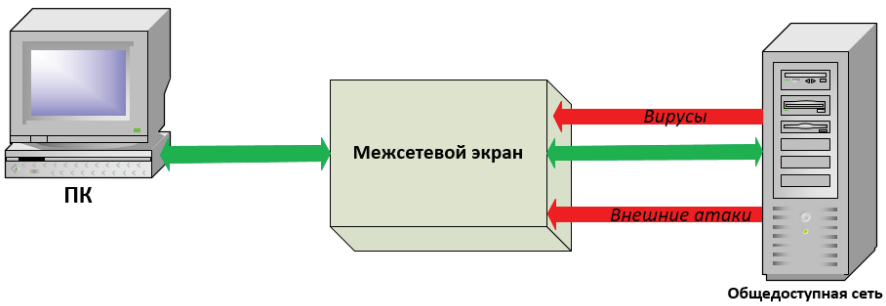


Рисунок 1 – Схема работы межсетевого экрана

интернет. Он также контролирует тип трафика, порождаемого вне и направляющегося в вашу приватную сеть. Трафик, не попавший в правила, блокируется, а в файл протокола вносятся соответствующие записи. Включающие брандмауэры обычно более безопасны, чем исключающие, поскольку они существенно уменьшают риск пропуска межсетевым экраном нежелательного трафика» [4].

Фильтрация трафика происходит на основе заранее установленных правил безопасности. Для этого создается специальная таблица, куда заносится описание допустимых и недопустимым к передаче данных.

Файрволлы могут запрещать или разрешать доступ, основываясь на разных параметрах: IP-адресах, доменных именах, протоколах и номерах портов, а также комбинировать их [5].

«Эффективность МСЭ обуславливается тем, что:

- все соединения проходят через межсетевой экран (в противном случае, если есть альтернативный сетевой маршрут, эффективность сильно снижается);
- межсетевой экран пропускает только санкционированный трафик;
- брандмауэр должен противостоять атакам против самого себя.

У данного решения есть ряд преимуществ:

- Межсетевой экран ограничивает доступ к определенным службам (например, общий доступ к веб-узлу может быть разрешен, а к telnet – запрещен).
- Межсетевой экран – средство аудита, может заносить в журнал информацию о любом проходящем трафике.
- Межсетевые экраны обладают возможностями по оповещению о конкретных событиях» [6].

Для реализации идеи будет использован открытый код, что поможет сократить объем работы и уменьшить время, затраченное на разработку. Проверка продукта будет осуществляться с помощью Фреймворка, рассмотрим некоторые из них.

Cobalt Strike – это отличный фреймворк для эксплуатации и постэксплуатации. В качестве пейлоада используется Beacon, у которого есть возможности обфускации и фриза для обхода антивирусов, поддерживает миграцию в процессы, подходит в качестве сервера C2 – особенно удобно ориентироваться при большом скоупе. Из коробки имеет генератор полезных нагрузок в один клик, а также различные методы доставки, что экономит немало времени. Cobalt Strike генерирует собственные исполняемые файлы и библиотеки DLL с помощью Artifact Kit. Они, в свою очередь, отправляют полезную нагрузку, что помогает обойти некоторые антивирусы. Однако у данного решения есть существенная проблема – он недоступен для рядовых пользователей. Cobalt Strike – коммерческий продукт, и разработчики серьезно относятся к его распространению. Есть пробный период на 21 день, но в таком режиме мы стакиваемся с существенными ограничениями [6].

Пробная версия «не загружает и не использует гибкие профили C2. Это функция, которая позволяет пользователям изменять сетевые индикаторы в полезной нагрузке Beacon. Каждый HTTP-запрос GET пробной версии включает заголовок X-Malware со строкой EICAR в качестве содержимого. Аналогично модули для атак на Java включают файл EICAR внутри пакетов .jar. Также из пробной версии удален основной энкодер полезной нагрузки Cobalt Strike. Все эти ограничения сделаны для того, чтобы пробную версию нельзя было использовать в злонамеренных целях» [7] (рисунок 2).

Metasploit является великолепным инструментом в системах проверки. Он может использоваться при пентестинге для создания отчетов совместно с другими системами автоматического обнаружения уязвимостей. С помощью этого фреймворка можно установить, являются ли уязвимости опасными и можно ли ими воспользоваться для проникновения в систему [8]. Metasploit может использоваться для проверки новых эксплоитов на локальном сервере, специально предназначенном для этих целей. С помощью этого фреймворка вы легко можете проверить эффективность продукта [8] (рисунок 3).

Существующие решения имеют ряд недостатков. «Основной недостаток проявляется в самой архитектуре межсетевого экранирования. Традиционные МСЭ до сих пор используют старые правила политики безопасности, основанные на IP-адресах и TCP/UDP портах, в то время как современные

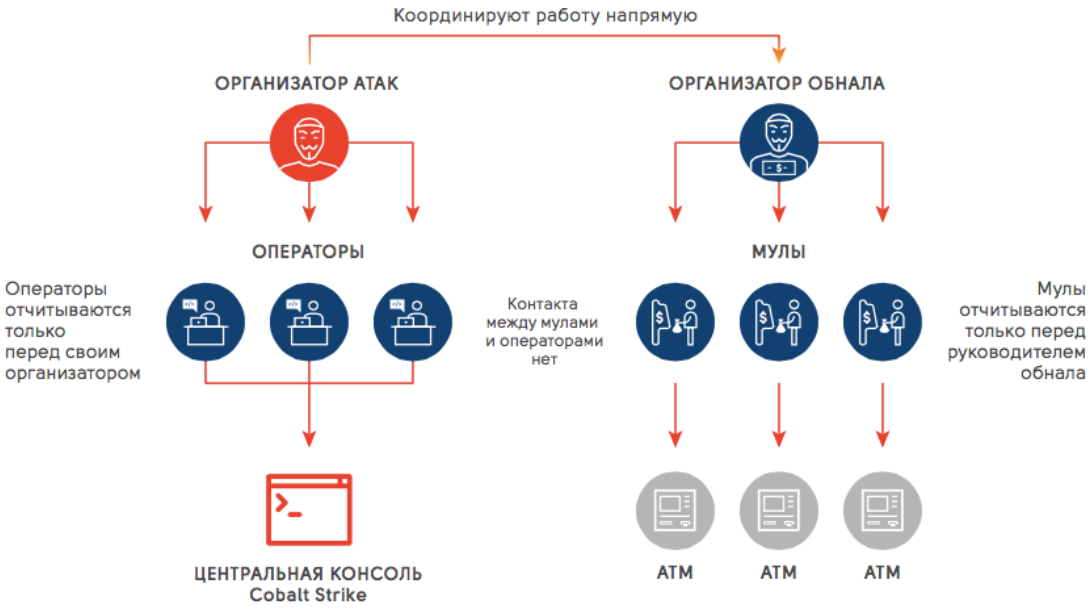


Рисунок 2 – Устройство группы Cobalt Strike

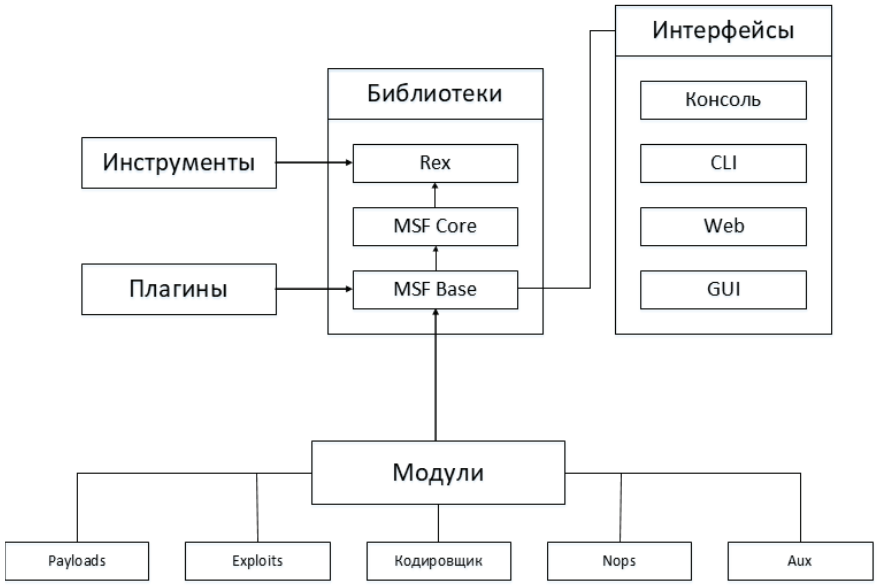


Рисунок 3 – Схема архитектуры Metasploit

приложения могут использовать как нестандартные порты, так и динамически использовать целый пул портов TCP/UDP, легко обходя при этом политики безопасности традиционного межсетевого экрана» [9].

Ещё одним важным недостатком является применение дорогостоящих, так называемых «помощников» МСЭ, к которым можно отнести отдельно стоящие серверы URL-фильтрации, IPS, антивирусную защиту и т. д. Такой подход приводит к несогласованности политик безопасности и не позволяет решить проблему мониторинга и управления трафиком приложений, при этом возникает проблема в неточной или неполной классификации трафика, сложных процедурах управления и дополнительных задержках, вызванных множеством процессов сканирования [10].

Если в рамках МСЭ реализовано сразу много сервисов безопасности, например, сам по себе МСЭ, потом система предотвращения вторжений, сетевой антивирус и т. п., то одновременное включение всех этих функций может существенно снизить реальную производительность такого устройства. В настоящее время в мире наблюдается постоянная модернизация существующих сетевых приложений, поэтому компания-разработчик МСЭ должна иметь возможность быстрого и гибкого управления профилями защиты.

Третьим недостатком является то, что при реализации традиционной архитектуры подсистем защиты информационных систем, в состав которых входят несколько МСЭ, объединённых в сеть управление сервисами безопасности, системами обнаружения вторжений, сетевыми антивирусами и сетями межсетевых экранов, осуществляется ядром подсистемы защиты, т. е. в процессе функционирования подсистемы защиты вся нагрузка ложится на центральный процессор, который управляет ядром подсистемы защиты. При этом значительно снижается производительность подсистемы защиты в целом, что приводит к снижению быстродействия подсистемы защиты, а значит и к увеличению времени обнаружения несанкционированного доступа в информационную систему, что напрямую приводит к уменьшению коэффициента защищенности информационной системы.

Одним из подходов к решению этих проблем является разработка новой концепции построения системы межсетевых экранов (ОСМЭ), основанной на облачных и нейросетевых технологиях и технологиях ПЛИС программируемых логических интегральных схем, которые применительно к рассматриваемым проблемам представляют собой аппаратные облачные сети (АОС).

Следует отметить, что в данном случае под АОС понимается сеть передачи данных, в которой определение функционала устройства сети происходит по неизвестному алгоритму в пределах локального аппаратного облака. При этом алгоритмы определения функционала устройств сети и архитектура локального аппаратного облака для каждого частного случая определяются и формируются ядром АОС, управляемым нейронной сетью.

В свою очередь, под аппаратным облаком подразумевается модель, обеспечивающая доступ по требованию к общей сети программно-аппаратных устройств для использования алгоритмов определения функциональных возможностей и распределения вычислительных ресурсов. При этом архитектура, электротехнические составляющие, IT инфраструктура находятся в «туманной завесе», т. е. в «облаке» и скрыты от объектов глобальной сети Internet. При этом взаимодействие внутри аппаратного облака между устройствами осуществляется по шифрованным каналам передачи информации – облачным кольцевым информационным тоннелям.

С учетом возможностей нейросетевых технологий алгоритм создания элементов облачной системы межсетевых экранов (ОСМЭ) можно представить в следующем виде:

1. С помощью фреймворка, хранящего в своей базе данных шаблоны сигнатур различных межсетевых экранов, создается модель межсетевого экрана (МСЭ), удовлетворяющего всем критериям сетевой безопасности для указанной информационной системы – модель МСЭ ИС.

2. После сигнатура модели МСЭ при помощи САПР преобразуется в пригодную для использования в ПЛИС платах сигнатуру модели МСЭ ИС, после чего ПЛИС программируется и становится аппаратным МСЭ ИС с адаптированным/ расширенным функционалом.

3. Далее происходит объединение группы аппаратных МСЭ ИС, географически удаленных друг от друга, в систему аппаратных облачных МСЭ ИС, управляемых ядром облачной системы аппаратных межсетевых экранов (ОСМЭ). Система из таких устройств будет соединена облачным сервисом под управлением нейронной сети и, по сути, станет подсистемой защиты для ИС.

Для подсистем защиты ИС, управляемых нейронной сетью, можно выделить следующие характерные задачи:

1. Анализ трафика с последующим предсказанием возможного вторжения. При решении данной задачи используется главное преимущество нейросети, которое состоит в том, что она может самостоятельно обучаться, не опираясь на заложенные в нее данные.

2. Выявление аномалий нормального поведения сети, при котором любое отклонение будет рассматриваться, как потенциальная угроза сетевой безопасности или атака.

В обобщенном виде логическая блок-схема работы нейронной сети представлена на рисунке 4 [11].

Рассмотрим алгоритм работы нейронной сети.

1. После обработки входящего потока данных от аппаратных облачных МСЭ ИС (сенсоров) на вход нейронной сети поступает уведомление/сигнал о нестандартных действиях в работе, другими словами, на вход нейронной сети поступает уведомление о появлении аномального трафика данных (АТД).

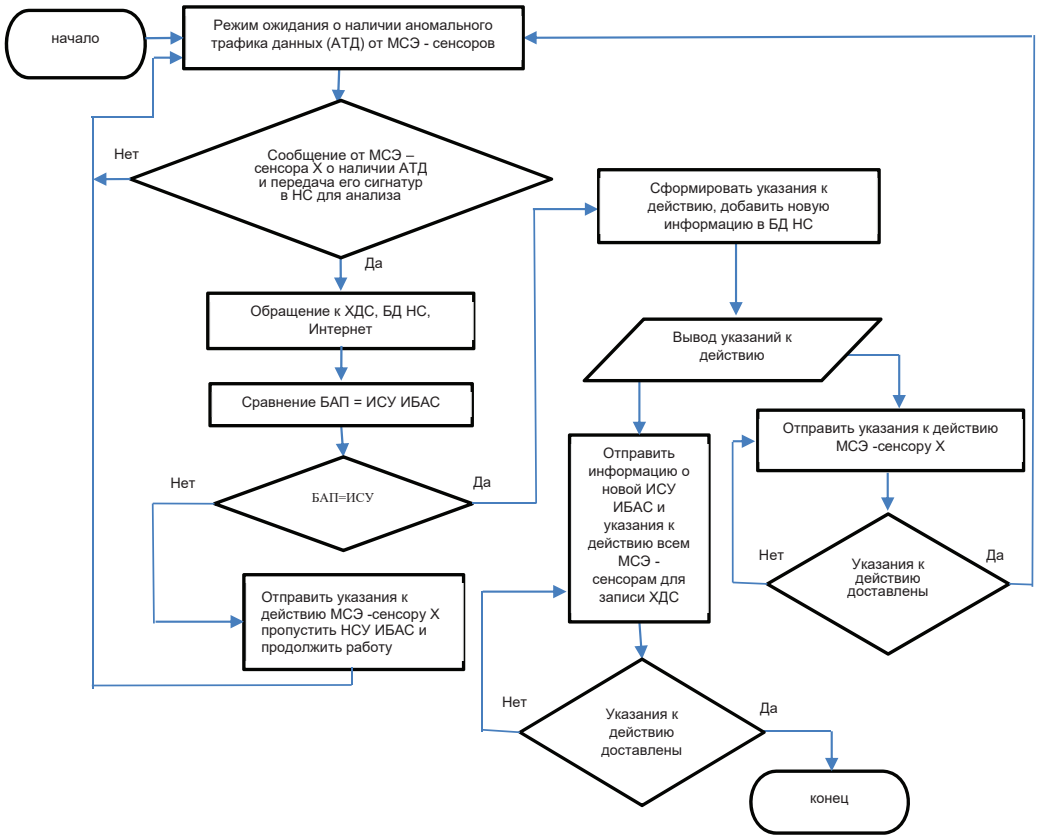


Рисунок 4 – Логическая блок схема алгоритма функционирования нейронной сети:
АТД – блок аномальных пакетов трафика данных – нестандартное действие извне;
ОХДС – облачное хранилище данных сенсора; ИСУ ИБАС – идентифицированные сигнатуры угроз;
НСУ ИБАС – неидентифицированные сигнатуры угроз ИБАС

2. Нейронная сеть, применив обученные алгоритмы, проводит анализ аномального трафика.

3. Далее выделяется блок аномальных пакетов и классифицируется, как потенциальная сигнатура угрозы (СУ) информационной безопасности автоматизированной системы (ИБАС).

4. После классификации сигнатур угроз нейронная сеть обращается к базе данных, хранилищу данных сенсоров (хранилище данных, расположенное внутри сенсоров), ресурсам глобальной сети интернет, где содержится информация о ранее известных СУ ИБАС, и путем сравнения определяет является ли действие угрозой.

5. Если в базах данных находятся схожие по сигнатурам угрозы, нейронная сеть к соответствующему типу угроз ИБАС и отправляет сенсору, отправившему запрос указание к действию по алгоритму.

6. Далее происходит добавление информации в базу данных о выявленном подтипе ранее известной угрозы ИБАС.

7. Следующим шагом нейронная сеть запускает процесс информирования сенсоров, входящих в облако.

8. Если нейронная сеть при обращении к базам данных не находит сигнатур, подтверждающих принадлежность блока аномальных пакетов данных к угрозам ИБАС, и не является угрозой, происходит информирование сенсора, направившего запрос о том, что действие не является угрозой.

Следует отметить, что большинство видов кибератак на существующие ИС легко идентифицировать, т. к. алгоритмы работы этих атак и сигнатуры ПО, связанного с ними, известны заранее и хранятся в соответствующей БД сигнатур угроз ОСМЭ. Тем не менее, всегда существует потенциальная угроза целенаправленных «атак-новелл» на уязвимости в ИС, которая может нанести реальный урон ИС до момента обнаружения и обезвреживания аппаратным МСЭ-сенсором [12].

Логическая блок-схема работы МСЭ-сенсора представлена на рисунке 5.

Рассмотрим алгоритм работы МСЭ-сенсора.

1. При поступлении нестандартного трафика данных (НТД), содержащего в своем составе блок аномальных пакетов (АТД) на вход сенсора, запускается стандартный алгоритм проверки межсетевого экрана (рисунок 5), по которому происходит сравнение с сигнатурами ранее известных угроз – идентифицированными сигнатурами угроз (ИСУ) ИБАС, хранящимися в хранилище данных сенсора (ХДС), и, в случае обнаружения принадлежности блока аномальных пакетов к ранее известной угрозе, по стандартному алгоритму запускаются указания к действию заблокировать АТД и продолжить работу.

2. Если при сравнении АТД, поступившего на вход устройства с сигнатурами ранее известных угроз – идентифицированными сигнатурами угроз (ИСУ) ИБАС, хранящимися в хранилище данных сенсора, не будут идентифицированы критерии относящие АТД к какой-либо ранее известной угрозе, хранящейся в хранилище данных сенсора, АТД направляется для обработки в нейро-сетевой модуль и классифицируется, как неизвестное, нестандартное действие извне – неидентифицированная сигнатура угроз (НСУ) ИБАС.

3. Нейро-сетевой модуль по алгоритму направляет на вход нейронной сети уведомление/сигнал о нестандартных действиях в работе, другими словами, на вход нейронной сети поступает уведомление о появлении аномального трафика данных НСУ ИБАС.

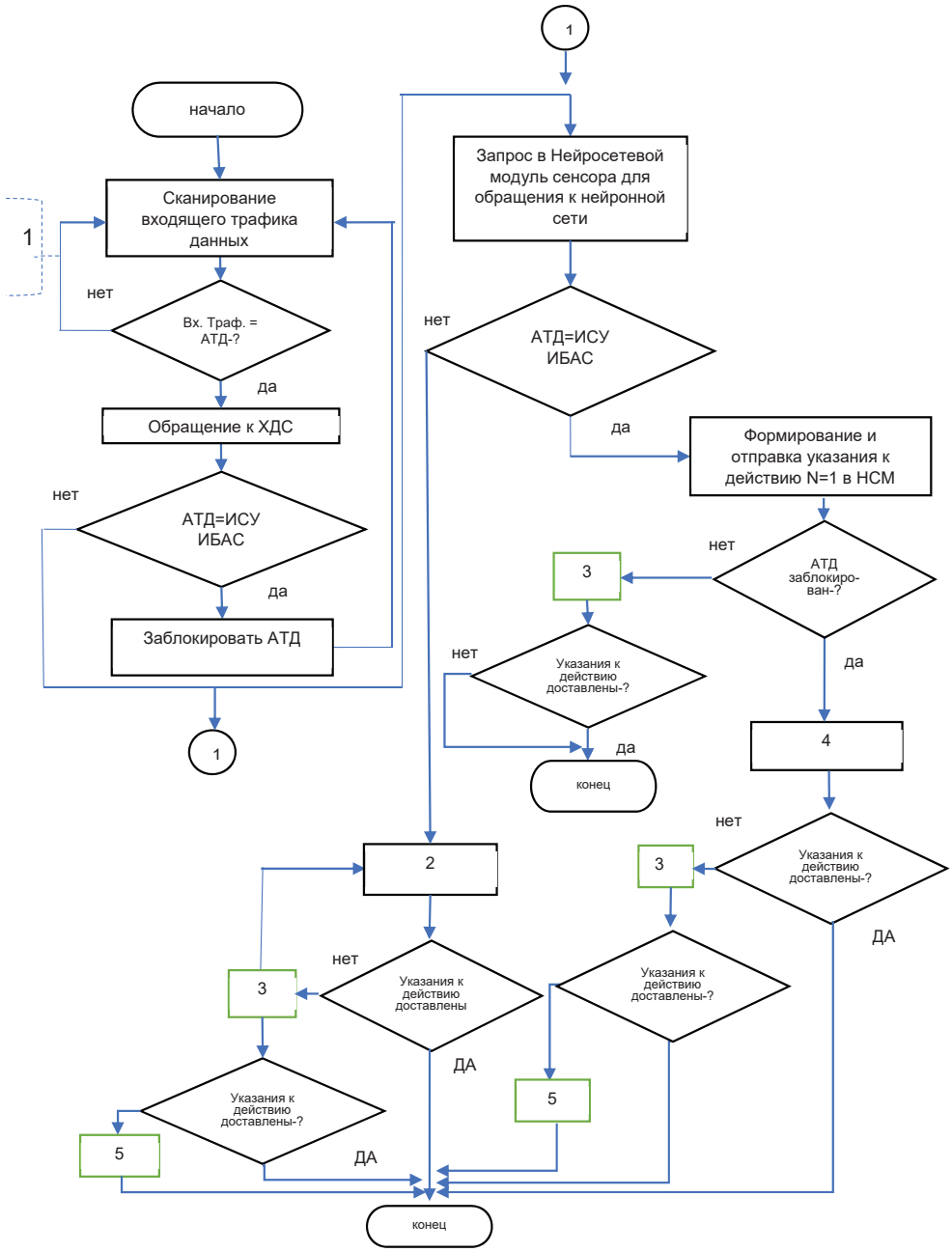
4. Далее запускается алгоритм работы нейронной сети.

5. Если нейронная сеть идентифицирует НСУ ИБАС как угрозу ИБАС, то нейро-сетевому модулю от нейронной сети приходит уведомление о подтверждении угрозы ИБАС, и направляется команда заблокировать НСУ ИБАС и продолжить работу.

6. Если нейронная сеть не идентифицирует НСУ ИБАС как угрозу ИБАС, то нейро-сетевому модулю от нейронной сети приходит уведомление и команда пропустить НСУ ИБАС и продолжить работу.

Блок-схема архитектуры аппаратного МСЭ ИС с расширенным функционалом приведена на рисунке 6. Данная блок-схема отличается от существующего прототипа (Metasploit) наличием нейро-сетевого модуля, который существенно расширяет функционал аппаратного МСЭ ИС.

Логическая схема работы нейро сетевого модуля приведена на рисунке 7.



Где: АТД – блок аномальных пакетов трафика данных-не стандартное действие из вне.
ХДС – хранилище данных сенсора.
ИСУ ИБАС – идентифицированные сигнатуры угроз
НСУ ИБАС – неидентифицированные сигнатуры угроз ИБАС
1– пропустить ИБАС и продолжить работу
2 – Отправить указания к действию N=1- «пропустить НСУ ИБАС и продолжить работу» в НМС,
3 – Повторить отправку указаний к действию
4 – Отправить информацию о новой ИСУ ИБАС и указания к действию N=1 – «Заблокировать АТД», всем МСЭ - сенсорам для записи ХДС
5 – отправить отчет в БД инцидентов ИБ и завершить процесс

Рисунок 5 – Логическая схема работы сенсоров

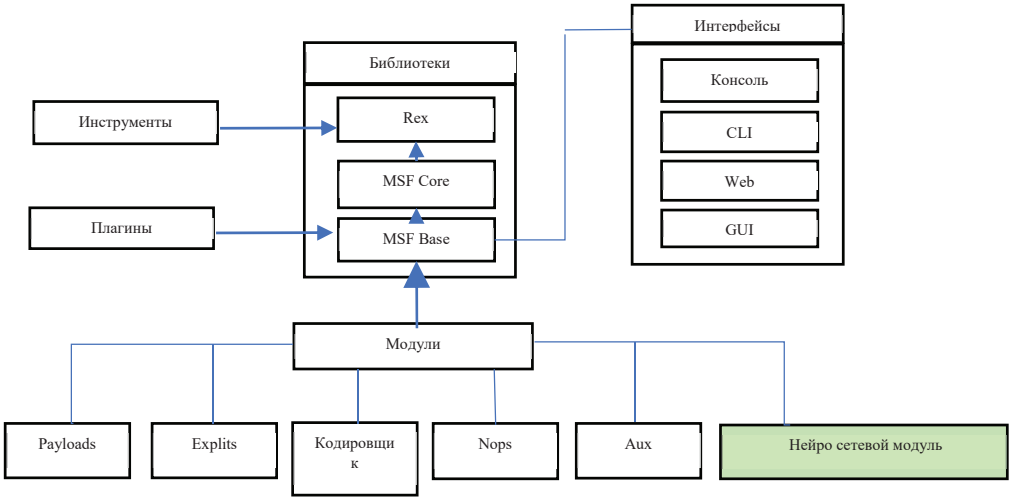
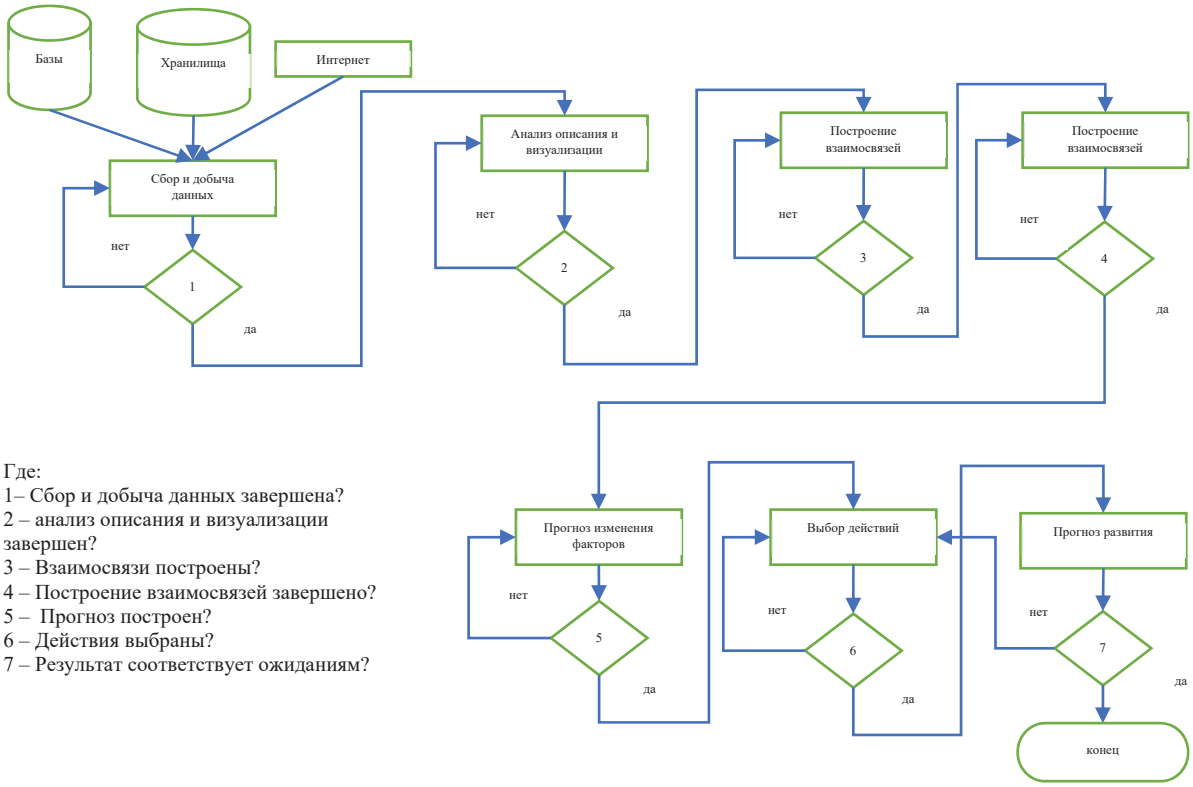


Рисунок 6 – Схема архитектуры аппаратного МСЭ ИС



Где:
1– Сбор и добыча данных завершена?
2 – анализ описания и визуализации завершен?
3 – Взаимосвязи построены?
4 – Построение взаимосвязей завершено?
5 – Прогноз построен?
6 – Действия выбраны?
7 – Результат соответствует ожиданиям?

Рисунок 7 – Логическая схема работы нейросетевого модуля

Исходя из изложенного выше можно констатировать, что применение искусственных нейронных сетей для детектирования вторжений сегодня является инновационным способом при создании систем защиты информации от НСД. Это обусловлено тем, что нейросети обладают гибкостью, что дает им способность обучаться в режиме реального времени, что повышает вероятность верного срабатывания при детектировании атак [12, 13].

Использование ПЛИС в качестве аппаратной компоненты для создания конфигурируемых облачных аппаратных МСЭ при реализации предлагаемого подхода, позволяет решить вопросы, связанные со снижением производительности подсистемы защиты в целом, снижением быстродействия подсистемы защиты, уменьшением времени обнаружения несанкционированного доступа в информационную систему, а также повышением коэффициента защищенности информационной системы.

Другими словами, можно говорить о том, что предлагаемые на базе ПЛИС конфигурируемые МСЭ, имеющие в своем составе нейросетевой модуль, обладают функцией самоорганизации, что позволяет им автономно осуществлять конфигурацию функций МСЭ, автономно реагировать на потенциальную угрозу и принимать решение о ликвидации НСД.

При этом, по сравнению с традиционными подходами построения сетей МСЭ, где обработка производится процессором ядра подсистемы защиты, использование предлагаемых на базе ПЛИС конфигурируемых МСЭ значительно уменьшает время обработки входящей информации и обнаружения НСД в ИС за счет высокого быстродействия ПЛИС, а обращение к ядру подсистемы защиты будет происходить только в трех случаях:

1. Получение обновлений о наличии новых сигнатур угроз и конфигураций функций МСЭ.
2. Отправки отчетов не идентифицированных сигнатур угроз.
3. Отправки не идентифицированных сигнатур угроз.

На рисунке 8 представлена структурная блок-схема облачной системы межсетевых экранов, где ядром системы выступает обученная нейронная сеть, управляющая ОСМЭ; sensor 1÷N − аппаратные модули защиты – МСЭ ИС, объединённые между собой облачным кольцевым информационным тоннелем системы брандмауэров/ МСЭ в режиме верификации.

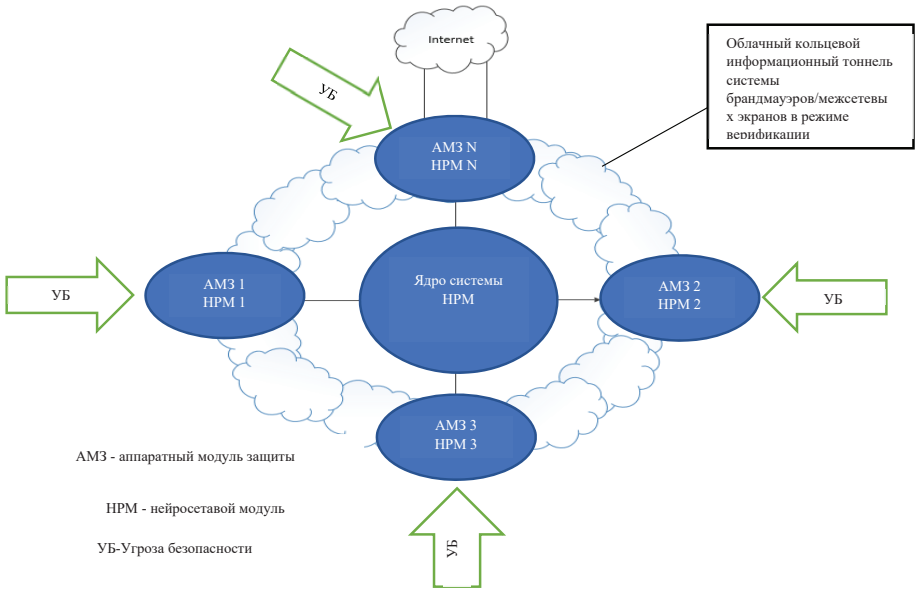


Рисунок 8 – Структурная блок-схема облачной системы межсетевых экранов

Показано, что задача формирования конфигурации правил защиты АМЗ-МСЭ, являющихся объектами ОСМЭ, заключается в подборе АМЗ-МСЭ, совокупность которых смогла бы обеспечить за ограниченное время не только максимальную вероятность обнаружения УБ, но и минимальную вероятность «ложной тревоги». Для ее решения предложен адаптированный способ организации защитных механизмов в облачных подсистемах защиты ИС, реализация которого позволит учесть индивидуальные особенности и характеристики АМЗ-МСЭ. Для реализации способа предложены алгоритмы обнаружения УБ, устраняющие недостатки традиционных логических схем. Одним из практических результатов внедрения предложенного способа в подсистемах защиты ИС будет являться сокращение времени обнаружения УБ.

«Традиционно для более эффективного анализа трафика, в сетях, как правило, используется алгоритм поиска профиля трафика. В рамках исследования разработанный подход дополняется методами интеллектуального анализа данных, которые позволяют снизить набор характеристик, что значительно ускоряет обработку информации. Это важно при анализе больших потоков данных, которые происходят в сетях виртуальных центров обработки данных из-за многочисленных пересечений каналов связи» [14].

В представленном способе организации АМЗ – МСЭ в аппаратном облаке правила межсетевого экрана/брандмауэра разрабатываются НСМ отдельно для уровня модели OSI, отвечающие за коммутацию, транспортировку и сетевое взаимодействие. «Все правила состоят из двух частей – заголовков, для идентификации пакетов, и действий. Заголовки для правил уровня L2 – порт пакета, переключатель источника, MAC-адреса отправителя и получателя пакета, тип протокола и т. д. Заголовки правил уровня L3-L4 – IP-адреса и порты источника и адресата пакета, тип инкапсулированного протокола и т. д. Возможные действия – удаление или разрешение пакета. При обработке правила объединяются в цепочки – упорядоченный список правил с идентификатором. Вся цепочка правил проверяется в порядке приоритета до тех пор, пока не будет правила, подходящего для проанализированного пакета. В этом случае выполняется действие, указанное в этом правиле, и последующее выполнение цепочки завершается. Чтобы передать правило в другую цепочку, можно указать соответствующее действие с его идентификатором» [14].

Таким образом, при обнаружении аномалий в сетевом трафике, приходящем на порты АМЗ-МСЭ ИС, происходит поиск сигнатур, хранящихся в хранилищах данных АМЗ-МСЭ, в случае, когда в хранилищах данных АМЗ-МСЭ сигнатуры со схожими признаками отсутствуют, нейро-сетевой модуль запускает алгоритм проверки сигнатур, хранящихся в базах данных ядра сети ОСМЭ, при отсутствии поиск в базах данных из сети Internet. Далее нейросеть проводит анализ, идентифицирует поверяемую сигнатуру, относит ее к классу угроз, добавляет в БД, после чего дает команду АМЗ-МСЭ ИС и дублирует ее на все АМЗ-МСЭ, подключенные к ОСМЭ, тем самым происходит аппаратное определение сети (АОС) [8–10].

Еще одним преимуществом предложенного подхода с использованием нейронной сети Кохонена, является способность идентифицировать новые кластеры. Обученная нейронная сеть распознает кластеры в данных обучения и назначает все данные тому или иному кластеру. Если сеть затем встречает набор данных, который отличается от любого из известных образцов, он будет независимо определять новый кластер элементов. Эта функция очень актуальна, поскольку она позволяет защитить архитектуру ОСМЭ без фактического изменения алгоритмов обнаружения атак [11, 15, 16].

Экспериментальные исследования показали, что разработанная модель прототипа ОСМЭ, являющаяся компонентом информационной системы как подсистема сетевой защиты/безопасности, позволяет за счет использования ПЛИС создавать распределенные аппаратные облачные сети, географически удалённые друг от друга, значительно сократить время отклика между межсетевым экраном и ядром информационной системы, при этом при проведении кибератак значительно сократить время отклика приложений, что, в свою очередь, значительно повышает устойчивость системы при совершении НСД.

Другими словами, предложенная модель прототипа ОСМЭ позволяет расширить функционал МСЭ за счет внешнего ПЛИС модуля, тем самым значительно повышая коэффициент защищенности ИС от угроз информационной безопасности.

Поступила: 26.12.22; рецензирована: 11.01.23; принята: 13.01.23.

Литература

1. *Урусова Т.Н.* Тенденции развития угроз информационной безопасности и меры по снижению рисков взлома. Будущее науки-2020 / Т.Н. Урусова // Сб. научных статей 8-й Межд. молод. науч. конф.: в 5 т. 21–22 апреля. – Курск, 2020 – С. 429–433.
2. *Коккоз М.М.* Методы борьбы с угрозами информационной безопасности государства / М.М. Коккоз, А.У. Альжанова, А.М. Аубакиров, Д.К. Жарилхасинова // Молодой ученый. 2017. № 8(142). Караганда. С. 237–240.
3. Свободная энциклопедия Википедия. Межсетевой экран. URL: https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B6%D1%81%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%8D%D0%BA%D1%80%D0%B0%D0%BD (дата обращения: 20.05.2022).
4. *Руденко О.Г.* Искусственные нейронные сети / О.Г. Руденко, Е.В. Бодянский. – Харьков, 2005.
5. Varonis Systems, Информационная безопасность, Тестирование IT-систем, Metasploit. Руководство для начинающих. URL: https://habr.com/ru/company/varonis/blog/528578/. (дата обращения 16.04.2022).
6. Проблемы распределенной обработки данных; классификация сетей по различным признакам. Сравнительная характеристика сетей различных типов. URL: https://studfile.net/preview/4431318/page:3/. (дата обращения: 01.01.2023).
7. Фреймворки для постэксплуатации. URL: https://xakep.ru/2019/10/18/post-exploitation-frameworks/ (дата обращения: 01.01.2023).
8. Нейронные сети в кибербезопасности. URL: https://habr.com/ru/post/587694/ (дата обращения: 16.05.2022).
9. Нейронные сети Кохонена. URL: https://neuronus.com/theory/nn/955-nejronnye-seti-kokhonena.html. (дата обращения: 10.05.2022).
10. Информационная безопасность. Журнал. 2013. № 1. URL: http://biblioclub.ru/index.php?page=book_red&id=210607 (дата обращения: 05.01.2023).
11. *Корякин С.В.* Разработка программно-аппаратного ядра универсальной среды проектирования автоматизированных систем защищенного исполнения / С.В. Корякин // Проблемы автоматики и управления. Бишкек: ИАИТ, 2020. № 1 (38). С. 60–69.
12. *Ногин В.Д.* Принятие решений в многокритериальной среде: количественный подход / В.Д. Ногин. М.: Физматлит, 2005. 176 с.
13. *Советов Б.Я.* Интеллектуальные системы и технологии : учебник для студ. учреждений высш. проф. образования / Б.Я. Советов. М.: «Академия», 2013. 320 с.
14. *Болодурина И.П.* Исследование модели нейронной сети для обеспечения безопасности и качества обслуживания мультиоблачной платформы / И.П. Болодурина, Д.И. Парфенов // Информационные и математические технологии в науке и управлении. 2018. № 3 (11). С. 18–26. URL: https://www.elibrary.ru/item.asp?id=36322156 (дата обращения: 05.01.2023).
15. *Курзыкина А.В.* Проблемы внедрения автоматизированной информационной системы / А.В. Курзыкина // Молодой ученый. 2017. № 4. С. 124–167. URL: https://moluch.ru/archive/138/38806/ (дата обращения: 16.09.2020).
16. Matthias Ehrgott. Multicriteria Optimization. Springer, 2nd edition, 2005.