МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт Искусственного Интеллекта

Кафедра «Информационная безопасность» (БК №252)

КУРСОВАЯ РАБОТА

по дисциплине «Предупреждение, выявление и установление причин и условий компьютерных инцидентов»

Тема курсовой работы: «Исследование подходов к созданию межсетевых экранов класса NGFW»

Студент группы ККСО-02-18: *Гуськов М.В.*

Руководитель работы: *Гончаренко В.Е.*

Работа представлена к защите: «____» декабря 2022 года

Оценка: «_____»

Москва 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Межсетевой экран – это вид барьера безопасности, размещенного между различными сетевыми средами, состоящего из специализированного устройства или совокупности нескольких компонентов и технических приемов, через который должен проходить весь трафик из одной сетевой среды в другую и, наоборот, при этом пропускается только авторизованный трафик, соответствующий местной политике безопасности. Межсетевые экраны – одно из основных сетевых устройств, обеспечивающих безопасность компьютерных сетей. МЭ обычно должен гарантировать соответствующую защиту, соизмеримую с оцененными угрозами, за счет стандартного набора правил, безоговорочно запрещающего все для всего трафика между сетями и добавляющего явные правила только для необходимых каналов связи.

С усложнением компьютерных сетей, сетевых протоколов и приложений, работающий с использованием сетей, появилась необходимость в более сложных межсетевых экранах, имеющих наибольший контекст для анализа, просматривающих данные протоколов всех уровней модели OSI, отслеживающих сессии пользователей, распознающие протоколы и приложения, не опираясь лишь на IP-адреса и порты конечных узлов. Такие межсетевые экраны назвали NGFW – new generation firewalls, то есть межсетевые экраны нового поколения.

На данный момент NGFW – далеко не новая технология. Межсетевые экраны уровня приложений (WAF) уже обладают куда большим функционалом в области анализа трафика прикладного уровня. Однако NGFW всё ещё пользуются спросом, так как, например, могут работать с множеством различных протоколов различных протоколов.

В данной работе будут проанализированы технологии, используемые в существующих NGFW, и рассмотрены перспективы их развития.

1. Анализ текущего состояния в области построения технологий и систем межсетевого экранирования класса NGFW

NGFW (new generation firewall — англ. “межсетевой экран нового поколения”) — разновидность межсетевых экранов, работающих на нескольких уровнях модели OSI и ориентированных на фильтрацию трафика приложений.

NGFW способны отслеживать сетевую активность, различая приложения и сессии пользователей в них с сохранением состояния, не опираясь только лишь на значение порта и протокола. NGFW поддерживают различные протоколы, способны находить в трафике сессии приложения конкретные действия пользователей и работать с ними на основе заданных политик. При этом NGFW поддерживают и функции классических МСЭ: например, блокирование соединений по IP и порту. Также NGFW имеют функционал IDS/IPS по обнаружению и предотвращению атак, пассивное сканирование сети для поиска хостов и сервисов в ней и обнаружения уязвимостей.

Таким образом, NGFW — это в некотором роде комбайн, сочетающий в себе множество технологий, что позволяет комплексно подходить к обеспечению безопасности сетевой активности.

Задачи NGFW:

- Различать приложения вне зависимости от порта, протокола, техник обхода обнаружения и SSL-шифрования перед дальнейшими действиями;
- Предоставлять детальную видимость и контроль над приложениями с использованием политик, включая отдельные функции приложений;
- Точно различать пользователей и в дальнейшем использовать информацию о пользователе как атрибут для контроля с помощью политик;

- Предоставлять защиту в реальном времени против широкого ряда угроз, в том числе действующих на уровне приложений;
- Реализовывать возможности классических межсетевых экранов и систем предотвращения вторжения;
- Поддерживать развёртывание дополнительных встроенных механизмов со сравнительно низким влиянием на производительность.

1.1. Исследование технологий передачи данных, корректность которых определяется с использованием межсетевых экранов класса NGFW

В современных реалиях большинство трафика приложений передаётся с помощью протокола HTTP(S). Безусловно, существуют и другие протоколы уровня приложений, однако в данной работе функционирование NGFW будет рассмотрено на примере обработки трафика веб-приложений, использующих протоколы HTTP и HTTPS.

1.1.1. Анализ структуры построения существующих технологий передачи данных, корректность которых определяется с использованием межсетевых экранов класса NGFW

Общение пользователя с веб-приложением происходит по принципам клиент-серверной архитектуры. Клиент посылает запрос к серверу, а сервер даёт на него ответ. Запрос и ответ формируются в соответствии с протоколом HTTP.

При использовании протокола HTTPS общению клиента и сервера предшествует этап TLS Handshake. Он устанавливает защищённое соединение между клиентом и сервером путём выработки общего ключа для шифрования.

Для более комплексного рассмотрения общения с веб-приложением стоит не забыть о том, что при использовании веб-приложений клиенты часто обращаются к серверам по доменному имени. Для разрешения доменного имени в IP-адрес используется протокол DNS. Клиент обращается к DNS-

серверу с запросом на разрешение доменного имени, и сервер в ответ возвращает IP-адрес, соответствующий запрашиваемому домену.

Итак, основные этапы передачи данных при обращении пользователя к веб-приложению:

- Разрешение доменного имени
- TLS Handshake — только для протокола HTTPS
- Общение по HTTP

Таблица 1. Структура обращения клиента к веб-приложению

Этап	**Процесс**	**Процедура**	**Действие**
Разрешение доменного имени	Локальное разрешение доменного имени	Обращение к NSS в Linux	Разрешение имени через файл hosts
			Поиск имени в кэше локального резолвера
		Работа службы DNS Client в Windows	Разрешение имени через файл hosts
			Поиск имени в кэше DNS Client
	Запрос к DNS-серверу	Пересылка DNS-запроса	Формирование DNS-запроса клиентом
			Парсинг DNS-запроса сервером
		Обработка запроса на DNS-сервере	Просмотр кэша сервера
			Просмотр файлов зоны сервера
			Отправка запроса другому DNS-серверу
TLS Handshake	Приветствие	Отправка ClientHello	Формирование сообщения ClientHello
			Отправка сообщения ClientHello серверу
		Отправка ServerHello	Формирование ServerHello
			Отправка ServerHello клиенту
			Отправка CertificateRequest клиенту
			Отправка ServerHelloDone клиенту

Этап	Процесс	Процедура	Действие
	Выработка общего ключа	Отправка клиентской части данных	Проверка приветствия сервера
			Выработка Pre_Master_Secret
			Отправка ClientKeyExchange
			Отправка сертификата
		Вычисление Master_Secret из Pre_Master_Secret	Вычисление Master_Secret на клиенте
			Вычисление Master_Secret на сервере
Общение по HTTP	Запрос от клиента	Формирование запроса от клиента	Проверка на наличие необходимых данных в кэше
			Формирование заголовков запроса
			Формирование тела запроса
		Отправка запроса по сети	Шифрование запроса (при использовании HTTPS)
			Непосредственная отправка запроса
	Ответ от сервера	Получение запроса по сети	Расшифрование запроса (при использовании HTTPS)
			Парсинг запроса
			Обнаружение сессии при помощи cookie
			Аутентификация клиента
		Формирование содержимого ответа	Отработка логики приложения
			Обращение к базам данных
			Работа с файловой системой
			Обращение к другим сетевым ресурсам
		Формирование ответа от сервера	Формирование заголовков ответа
			Формирование тела ответа
		Отправка ответа по сети	Шифрование ответа (при использовании HTTPS)
			Непосредственная отправка ответа

1.1.2. Исследование методологической основы построения технологий передачи данных, корректность которых определяется с использованием межсетевых экранов класса NGFW, и определение сценариев её применения

Рассмотрим подробнее реализацию действий, осуществляемых при передаче данных (столбец 4 таблицы 1).

1.1.2.1. Разрешение имени через файл hosts

В файле /etc/hosts задаются локальные соответствия пар “доменное имя — IP-адрес”.

1.1.2.2. Поиск имени в кэше glibc-резолвера

Резолвер (разрешитель имён) — механизм разрешения имён в Linux. Если разрешение имени удалось, некоторые резолверы (например, system-resolve) сохраняют в кэше полученное значение IP, чтобы при повторном обращении к данному имени не делать повторный запрос к DNS-серверу.

1.1.2.3. Разрешение имени через файл hosts

Аналогично файлу /etc/hosts в Linux, файл %WINDIR%\system32\drivers\etc\hosts в Windows содержит локальные соответствия пар “имя — IP”.

1.1.2.4. Поиск имени в кэше DNS Client

Аналогично резолверам в Linux, в Windows для разрешения имён используется служба DNS Client. Она кэширует полученные записи по схожему принципу, поэтому перед обращением клиента к DNS-серверу проверяются локальные записи в кэше DNS Client.

1.1.2.5. Формирование DNS-запроса клиентом

DNS-сообщение имеет следующую структуру:

Таблица 2. Поля заголовка сообщения DNS

Поле	**Описание**	**Длина в битах**
QR	Обозначает, является ли сообщение запросом (0) или ответом (1).	1
OPCODE	Тип запроса: QUERY (обычный запрос, 0), IQUERY (обратный запрос, 1) или STATUS (запрос состояния сервера, 2)	4
AA	Авторитетный ответ: в ответе устанавливается в 1, если DNS-сервер авторитетный для данной зоны, и в 0 в обратном случае.	1
TC	Усечение: устанавливается в 1, если сообщение было усечено из-за слишком большой длины	1
RD	Требуется рекурсия: в запросе устанавливается в 1, если клиент требует рекурсивный запрос	1
RA	Доступна рекурсия: в ответе устанавливается в 1, если DNS-сервер поддерживает рекурсию.	1
Z	Зарезервировано	3
RCODE	Код ответа: NOERROR (нет ошибки, 0), FORMERR (ошибка формата, 1), SERVFAIL (ошибка на сервере, 2), NXDOMAIN (домен не существует, 3) и так далее	4

Таблица 3. Поля записи DNS

Поле	**Описание**	**Длина в битах**
NAME	Имя узла	Переменная

TYPE	Тип записи (A, AAAA, MX, TXT, …) в числовом виде	2
CLASS	Код класса	2
TTL	Время жизни записи в секундах	4
RDLENGTH	Длина поля RDATA в байтах	2
RDATA	Дополнительные данные записи	Переменная, соответствует RDLENGTH

При отправке запроса клиент вместо полной записи посылает сообщение типа «вопрос», которое имеет структуру записи без полей TTL, RDLENGTH и RDATA.

1.1.2.6. Парсинг DNS-запроса сервером

DNS-сервер разбирает (парсит) полученный от клиента запрос в соответствии с форматом, описанным в 1.1.2.5.

1.1.2.7. Просмотр кэша сервера

Если ранее выполнялся рекурсивный поиск, в кэше сервера могли остаться записи с других серверов. Так как кэш — наиболее быстрое хранилище, оно в первую очередь просматривается на предмет соответствий.

1.1.2.8. Просмотр файлов зоны сервера

У DNS-серверов есть собственные записи, ставящие в соответствие доменное имя и IP-адрес. Хранение, поиск и выдача данных этих записей и является основным функционалом DNS-сервера.

1.1.2.9. Отправка запроса другому DNS-серверу

Если используется режим рекурсивного поиска, то, не найдя у себя необходимой записи зоны, DNS-сервер обращается к другим DNS-серверам для поиска необходимой записи. Получив ответ, в случае успеха изначальный DNS-сервер кэширует его и возвращает клиенту.

1.1.2.10. Формирование сообщения ClientHello

В сообщении ClientHello клиент передаёт серверу следующие параметры:

- версия SSL, поддерживаемая клиентом;
- идентификатор сеанса — значение, по которому впоследствии можно возобновить сеанс;
- случайное число Client_Random;
- список алгоритмов сжатия, шифрования и хеширования информации, поддерживаемых клиентом.

Также вместе с этими данными сервер отправляет клиенту свой сертификат, содержащий в том числе открытый ключ асимметричного шифрования.

1.1.2.11. Отправка сообщения ClientHello серверу

Сообщение отправляется в незашифрованном виде.

1.1.2.12. Формирование ServerHello

В сообщении ServerHello сервер передаёт клиенту следующие параметры:

- версия SSL, поддерживаемая сервером;
- случайное число Server_Random;
- список алгоритмов сжатия, шифрования и хеширования информации, которые будут использоваться при реализации сеанса или соединений.

1.1.2.13. Отправка ServerHello клиенту

Сообщение также передаётся в незашифрованном виде.

1.1.2.14. Отправка CertificateRequest клиенту

Опционально сервер может потребовать аутентификации клиента с помощью сертификата, если того требуют используемые алгоритмы.

1.1.2.15. Отправка ServerHelloDone клиенту

Это сообщение служит для того, чтобы обозначить окончание передачи сообщений ServerHello.

1.1.2.16. Проверка приветствия сервера

Чтобы продолжить алгоритм, необходимо, чтобы клиент поддерживал алгоритмы, предложенные сервером. Если это условие не выполняется, выполнение протокола прерывается.

1.1.2.17. Выработка Pre_Master_Secret

Это значение генерируется с использованием версии клиента и случайного значения.

1.1.2.18. Отправка ClientKeyExchange

В этом сообщении отправляется значение Pre_Master_Secret, зашифрованное на открытом ключе сервера.

1.1.2.19. Отправка сертификата

Если сервер ранее потребовал от клиента сертификат, клиент отправляет свой сертификат.

1.1.2.20. Вычисление Master_Secret на клиенте

Для вычисления значения Master_Secret на вход нелинейной функции подаются значения Client_Random, Server_Random и Pre_Master_Secret. В результате обмена приветствиями оба значения Random есть на обеих сторонах, а значение Pre_Master_Key передаётся от клиента к серверу в зашифрованном виде сообщением ClientKeyExchange.

1.1.2.21. Вычисление Master_Secret на сервере

Значение вычисляется абсолютно так же, как на клиенте, что описано в 1.1.2.20.

1.1.2.22. Проверка на наличие необходимых данных в кэше

Если клиент уже делал аналогичный запрос к тому же серверу ранее, и в ответе была обозначена политика сохранения ответа в кэше, то при новом запросе, если кэш ещё актуален, клиент может не обращаться к серверу, а получить данные из кэша.

1.1.2.23. Формирование заголовков запроса

HTTP имеет расширяемую структуру заголовков: каждое веб-приложение может использовать свои собственные заголовки для своих целей.

В спецификациях HTTP уже заданы стандартные поля заголовков, обеспечивающие как выполнение базовых функций HTTP (адресат запроса, метод запроса), так и аутентификацию, функции безопасности, использование прокси и т.д.

1.1.2.24. Формирование тела запроса

В зависимости от конкретного веб-приложения и конкретной вызываемой его функции, тело запроса может иметь совершенно разный формат. Это может быть как обычный текст, так и JSON или другой язык разметки. Данные чаще всего представляют в текстовой форме (т.е. сериализуют). Тело запроса также может быть пустым, когда вся специфика запроса описана в заголовках.

1.1.2.25. Шифрование запроса (при использовании HTTPS)

Если используется HTTPS, запрос шифруется с помощью алгоритма шифрования, установленного в ходе TLS Handshake, с использованием Master_Secret в качестве ключа симметричного шифрования.

1.1.2.26. Непосредственная отправка запроса

Как и любой другой запрос уровня приложений, HTTP-запрос инкапсулируется с использованием заголовков более нижних по модели OSI уровней: например, TCP, IP, Ethernet.

Здесь стоит заметить, что за конкретным сервисом (или всем веб-сервером) чаще всего зафиксированы один или несколько портов TCP/UDP. В свою очередь IP- и MAC-адреса должны идентифицировать конкретную машину и её физический интерфейс. Однако так происходит не всегда. На эти данные можно опираться при анализе трафика, но лишь частично.

1.1.2.27. Расшифрование запроса (при использовании HTTPS)

Как и при шифровании запроса, описанном в 1.1.2.25, используются алгоритм и ключ, оговоренные в ходе TLS Handshake.

1.1.2.28. Парсинг запроса

Сервер должен разобрать запрос клиента на заголовки и тело запроса. Дальнейшее поведение сервера при обработке тела запроса диктуется не в последнюю очередь различными заголовками запроса.

1.1.2.29. Обнаружение сессии при помощи cookie

HTTP — протокол без сохранения состояния, однако позволяет серверу отслеживать сессии с помощью установки значений cookie на клиентах. Клиенты могут указывать cookie в своих запросах, чтобы этот запрос был ассоциирован с предыдущими в рамках сессии.

1.1.2.30. Аутентификация клиента

HTTP предлагает заголовки для использования механизмов по аутентификации клиента. В результате аутентификации сервер может проверить, авторизован ли клиент, отправивший запрос, для требуемого действия. Сервер может отклонить запрос, если клиент не авторизован.

1.1.2.31. Отработка логики приложения

Веб-приложение может представлять собой сложный программный комплекс. Для выработки ответа на вопрос сервер должен выполнить некоторую программу. В зависимости от специфики выполняемых в рамках такой программы операций, площадь атаки может раскрываться очень широко.

1.1.2.32. Обращение к базам данных

Веб-приложения часто работают с базами данных. Данные из HTTP-запроса могут использоваться для составления запроса к базе данных (БД), что может приводить к инъекциям в запросы к БД. В результате клиент, используемый злоумышленником, может получить информацию, на получение которой у него нет прав, или получить доступ к управлению БД.

1.1.2.33. Работа с файловой системой

Веб-приложения могут хранить свои данные как в базах данных, так и в виде файлов в файловой системе. Программа, работающая с файловой системой, может иметь ряд уязвимостей, например, Path Traversal: клиент

может задать путь файловой системы, по которому программа читает данные, и получить данные из файлов, доступ к которым он не должен иметь.

1.1.2.34. Обращение к другим сетевым ресурсам

Для получения необходимых данных сервер может обращаться к другим серверам, отправляя в свою очередь новые запросы к другим сервисам. Такие запросы также могут стать инструментом или объектом атаки.

1.1.2.35. Формирование заголовков ответа

Когда сервер выработал ответ на уровне логики приложения, он приступает к формированию HTTP-ответа. Заголовки выставляются в соответствии с логикой передачи ответа. Например, может быть выставлена политика кэширования данного ответа. В заголовке ответа указывается код статуса запроса: был ли запрос выполнен успешно, с какой-либо ошибкой или требование к клиенту выполнить некоторое дальнейшее действие.

1.1.2.36. Формирование тела ответа

Данные ответа, как и в теле запроса (1.1.2.24), должны быть сериализованы для передачи посредством HTTP. Также возможен пустой ответ, когда содержимое ответа исчерпывается заголовками.

1.1.2.37. Шифрование ответа (при использовании HTTPS)

Как и для шифрования запроса, описанного в 1.1.2.25, используются алгоритм и ключ, оговоренные в ходе TLS Handshake.

1.1.2.38. Непосредственная отправка ответа

Как и для отправки запроса (1.1.2.26), происходит инкапсуляция трафика протокола HTTP в трафик протоколов нижних уровней. Связанные с этим особенности аналогичны.

Укажем в таблице методологическую основу для описанных действий. В большинстве своём это документы RFC.

Таблица 4. Методологическая основа действий при обращении клиента к веб-приложению

Этап	**Процесс**	**Процедура**	**Действие**	**М.О.**
Разрешение доменного имени	Локальное разрешение доменного имени	Обращение к NSS в Linux	Разрешение имени через файл hosts	hosts(5)
			Поиск имени в кэше локального резолвера	system-resolved
		Работа службы DNS Client в Windows	Разрешение имени через файл hosts	hosts
			Поиск имени в кэше DNS Client	Microsoft-Windows-DNS-Client
	Запрос к DNS-серверу	Пересылка DNS-запроса	Формирование DNS-запроса клиентом	RFC 1035
			Парсинг DNS-запроса сервером	RFC 1035
		Обработка запроса на DNS-сервере	Просмотр кэша сервера	RFC 1034
			Просмотр файлов зоны сервера	RFC 1034
			Отправка запроса другому DNS-серверу	RFC 1034
TLS Handshake	Приветствие	Отправка ClientHello	Формирование сообщения ClientHello	RFC 2246 RFC 4346 RFC 8446
			Отправка сообщения ClientHello серверу	
		Отправка ServerHello	Формирование ServerHello	
			Отправка ServerHello клиенту	
			Отправка CertificateRequest клиенту	
			Отправка ServerHelloDone клиенту	
		Отправка клиентской	Проверка приветствия сервера	

Этап	Процесс	Процедура	Действие	М.О.
	Выработка общего ключа	части данных	Выработка Pre_Master_Secret	
			Отправка ClientKeyExchange	
			Отправка сертификата	
		Вычисление Master_Secret из Pre_Master_Secret	Вычисление Master_Secret на клиенте	
			Вычисление Master_Secret на сервере	
Общение по HTTP	Запрос от клиента	Формирование запроса от клиента	Проверка на наличие необходимых данных в кэше	RFC 2616
			Формирование заголовков запроса	
			Формирование тела запроса	
		Отправка запроса по сети	Шифрование запроса (при использовании HTTPS)	RFC 2246 RFC 4346 RFC 8446
			Непосредственная отправка запроса	RFC 1122
	Ответ от сервера	Получение запроса по сети	Расшифрование запроса (при использовании HTTPS)	RFC 2246 RFC 4346 RFC 8446
			Парсинг запроса	RFC 2616
			Обнаружение сессии при помощи cookie	RFC 2109
			Аутентификация клиента	RFC 2617
		Формирование содержимого ответа	Отработка логики приложения	Выполнение логики приложения
			Обращение к базам данных	SQL, NoSQL
			Работа с файловой системой	Ext4, NTFS, ZFS, …

Этап	Процесс	Процедура	Действие	М.О.
			Обращение к другим сетевым ресурсам	Различные сетевые протоколы
		Формирование ответа от сервера	Формирование заголовков ответа	RFC 2616
			Формирование тела ответа	
		Отправка ответа по сети	Шифрование ответа (при использовании HTTPS)	RFC 2246 RFC 4346 RFC 8446
			Непосредственная отправка ответа	RFC 1122

1.1.3. Построение структурно-функциональной схемы систем и средств. Исследование существующих систем, комплексов и средств, реализующих технологии передачи данных, корректность которых определяется с использованием межсетевых экранов класса NGFW

Построим структурно-функциональную схему обращения клиента к серверу. Алгоритм довольно линейный, поэтому в нём почти нет ветвлений.

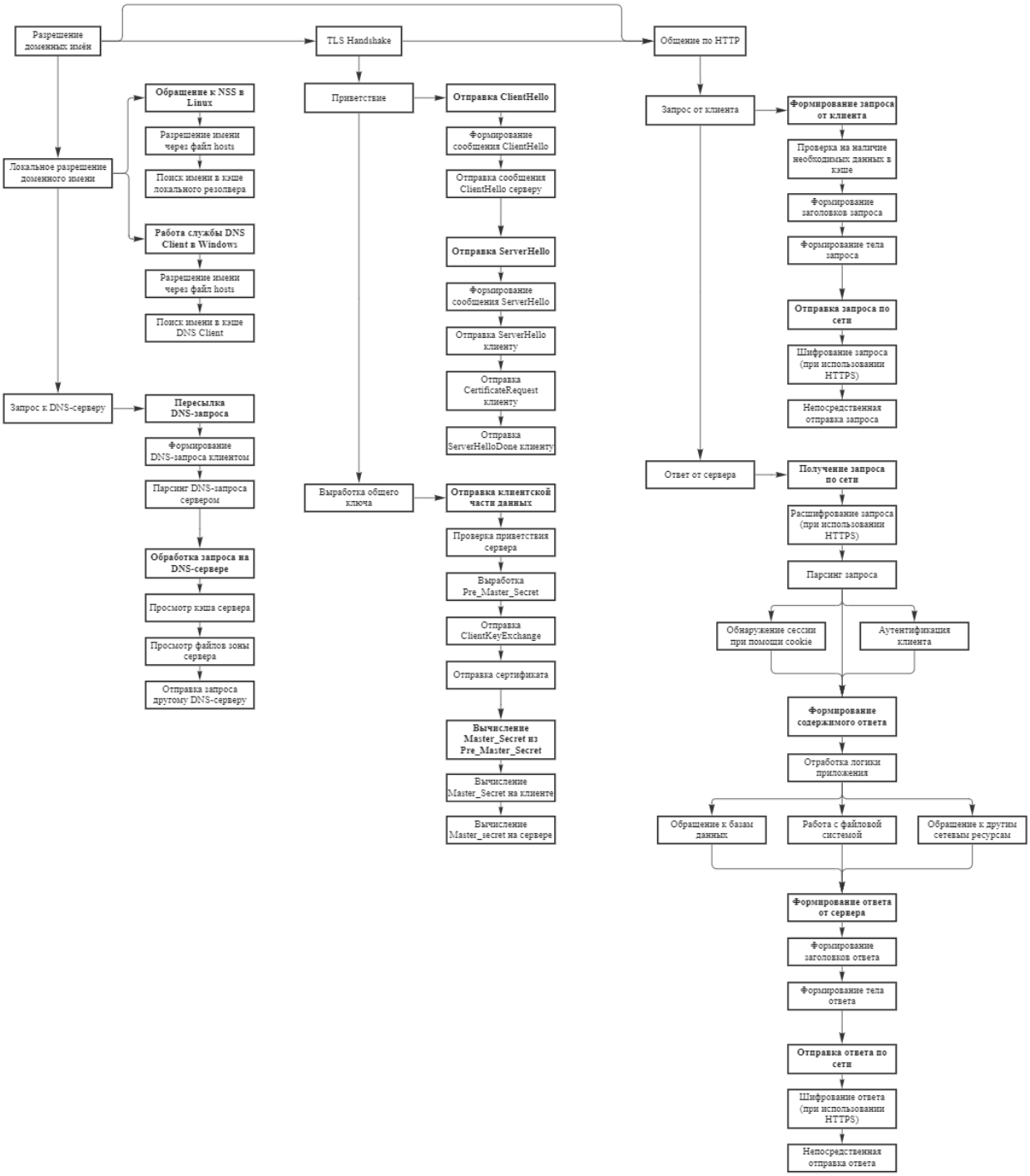

Рисунок 1 - Структурно-функциональная схема обращения клиента к веб-приложению

Также прилагается схема выполнения протоколов TLS Handshake и HTTP с установлением соединения по TCP.

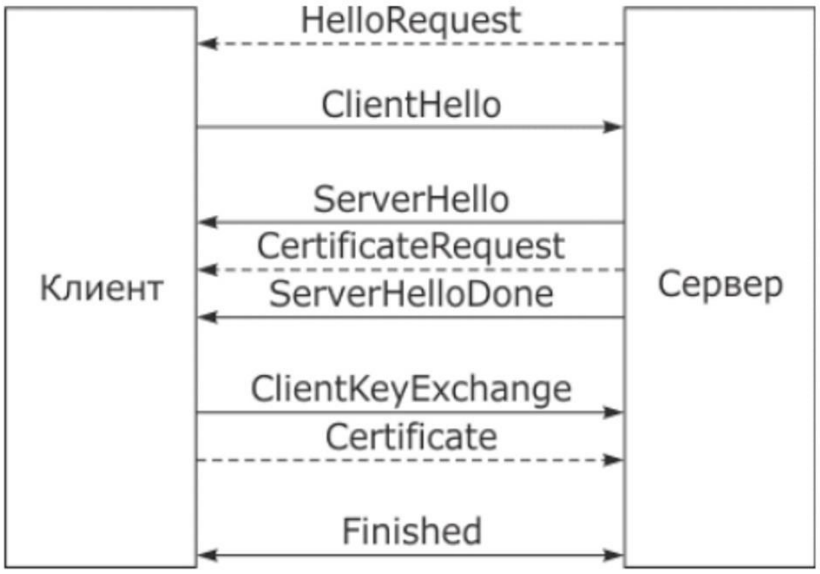

Рисунок 2 - Схема выполнения протокола TLS Handshake

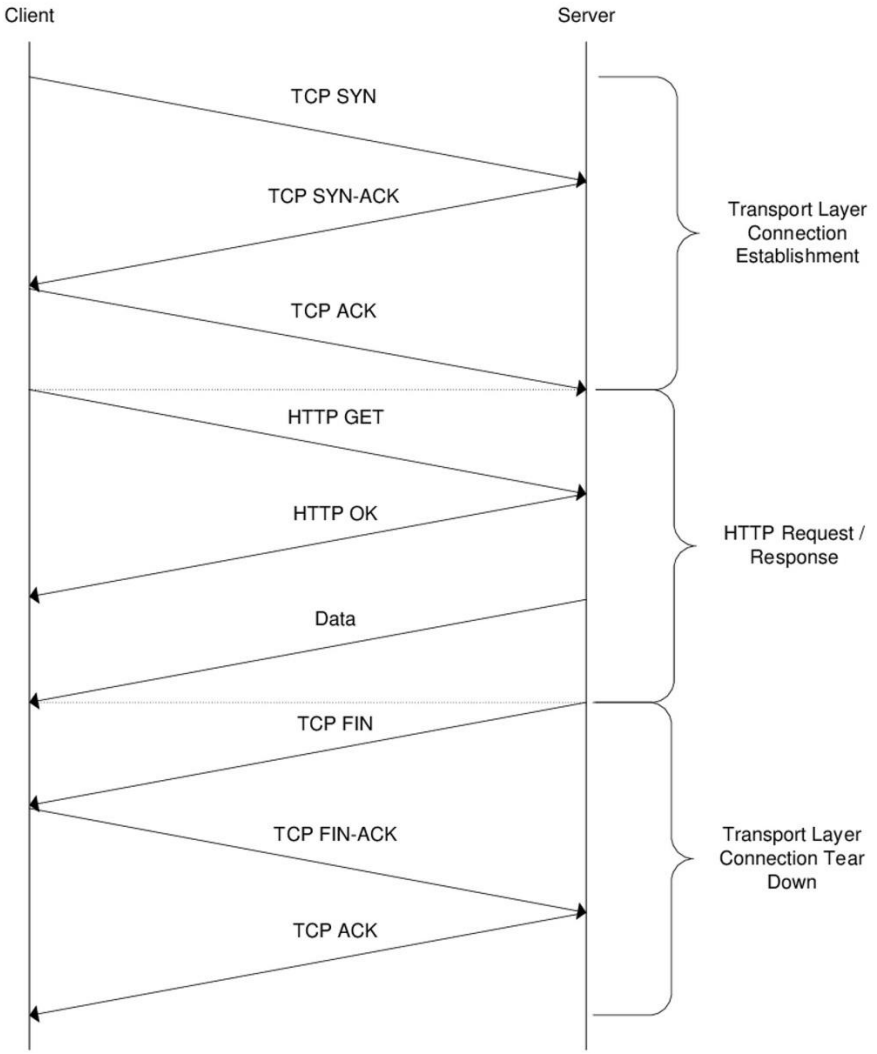

Рисунок 3 - Схема выполнения запроса HTTP с установлением соединения по TCP

1.2. Анализ состава задач в области построения технологий создания межсетевых экранов класса NGFW. Анализ предмета исследований – исследование существующих прикладных технологий создания межсетевых экранов класса NGFW.

Прикладные технологии — технологии межсетевого экранирования, реализуемые системами класса NGFW.

1.2.1. Анализ структуры построения существующих прикладных технологий создания межсетевых экранов класса NGFW и их составляющих (этапов, процессов, процедур, действий и т.п.).

На основе рассмотренной выше структуры технологий передачи данных выделим основные уязвимости такого процесса работы с данными и, соответственно, основной функционал NGFW по предотвращению атак на общение клиента и веб-приложения.

Основные задачи NGFW рассмотрим как этапы его функционирования.

Таблица 5. Структура функционирования NGFW

Этап	**Процесс**	**Процедура**	**Действие**
Идентификация приложений	Получение скрытого трафика приложения	Расшифрование зашифрованного трафика (использование SSL/TLS)	Обнаружение зашифрованного трафика
			Расшифрование зашифрованного трафика
		Декодирование закодированного трафика	Деобфускация трафика
			Обнаружение туннелирования
	Распознавание приложения	Распознавание по сигнатурам	Составление базы сигнатур
			Применение сигнатур к трафику
		Эвристическое распознавание	Составление базы эвристик
			Применение эвристик к трафику
Идентификация	Пассивная поддержка	Получение данных из AD	Запрос к доменному контроллеру AD

Этап	Процесс	Процедура	Действие
пользователей	базы данных пользователей		Отслеживание трафика AD
		Работа с трафиком пользователей	Поиск аутентификационных данных в трафике
			Запросы к устройствам пользователей
	Распознавание пользователей	Распознавание устройств	Распознавание на основе базовых идентификаторов
			Фингерпринтинг
		Распознавание пользователь-ских данных	Отслеживание сессий
			Отслеживание аутентификационных данных
Идентифи-кация содержимого	Предотвраще-ние угроз	Обнаружение угроз	Потоковое антивирусное сканирование
			Сигнатурное и эвристическое сканирование
		Работа с обнаруженны-ми угрозами	Отправка трафика в песочницу
			Действия с трафиком в соответствии с политиками
			Обновление базы данных сигнатур
	Фильтрация трафика	Фильтрация URL	Запрет трафика на основе прав пользователей
			Запрет трафика на основе базы запрещённых URL
		Фильтрация файлов и данных	Проверка данных на наличие чувствительной информации
			Проверка типов файлов
Управле-ние с помощью политик	Составление политик	Анализ предметной области	Использование данных пассивного сканирования
			Использование политик ИБ организации
		Формирова-ние политик	Составление правил для сочетаний пользователей, приложений и данных
			Указание необходимости дополнительных действий
			Разрешить трафик

Этап	Процесс	Процедура	Действие
	Применение политик	Базовые действия с трафиком	Запретить трафик
		Расширенные действия с трафиком	Отправить на сканирование угроз
			Перенаправить через прокси
			Проанализировать на наличие определённых данных

1.2.2. Исследование теоретических аспектов построения межсетевых экранов класса NGFW, определение сценариев применения теоретической основы построения существующих прикладных технологий.

1.2.2.1. Обнаружение зашифрованного трафика

NGFW должен обнаруживать шифрованный трафик и отличать его от простых бинарных данных, передаваемых по сети. Это может осуществляться за счёт анализа заголовков или при помощи анализа структуры данных.

1.2.2.2. Расшифрование зашифрованного трафика

Для расшифрования TLS-трафика в NGFW применяется подмена сертификата. NGFW проксирует весь трафик HTTPS. Он общается с клиентом, представляясь сервером и используя собственный сертификат TLS. Для этого необходимо, чтобы сертификат NGFW был доверенным на клиентах сети, контролируемой NGFW. С сервером же NGFW общается, представляясь клиентом. Таким образом, сам NGFW получает возможность расшифровывать трафик HTTPS и анализировать его.

1.2.2.3. Деобфускация трафика

Настоящий трафик может быть скрыт в структуре трафика уровня приложения. Например, исполняемый код может быть разбит на несколько частей и закодирован. При выполнении кода эти части могут быть декодированы, собраны обратно и выполнены. Таким способом часто

пользуются злоумышленники для того, чтобы избежать обнаружения вредоносного кода.

NGFW должен быть способен обнаруживать обфусцированный трафик и деобфусцировать его для дальнейших проверок.

1.2.2.4. Обнаружение туннелирования

Протокол прикладного уровня может быть использован в качестве туннеля для общения по другому протоколу. Примером тому может быть протокол SSH или технологии VPN. NGFW должен быть способен обнаруживать туннелирование и анализировать трафик внутри туннеля.

1.2.2.5. Составление базы сигнатур

Базы сигнатур могут быть получены как долговременным анализом сетевой активности, так и от поставщиков таких баз, занимающихся составлением таких баз в промышленных масштабах.

1.2.2.6. Применение сигнатур к трафику

Трафик должен быстро и эффективно сравниваться с рядом сигнатур, а результаты передаваться в системы мониторинга и применяться для решения по дальнейшим действиям с трафиком с использованием политик.

1.2.2.7. Составление базы эвристик

Аналогично сигнатурам (1.2.2.5), эвристики можно составлять как на собственном опыте, так и из централизованных источников.

1.2.2.8. Применение эвристик к трафику

Эвристики отличаются от сигнатур более абстрактным форматом, анализом поведения, поэтому необходимо сохранять состояние сессий трафика и применять эвристический анализ с использованием прежде полученного трафика, то есть использовать контекст.

1.2.2.9. Запрос к доменному контроллеру AD

Контроллер Active Directory в инфраструктуре Windows содержит данные о всех пользователях сети, их устройствах и правах. NGFW может запрашивать данные о пользователях у доменного контроллера.

1.2.2.10. Отслеживание трафика AD

Помимо прямых запросов к AD, NGFW может просматривать трафик протокола LDAP, использующегося для общения в AD, и находить в нём актуальную информацию о пользователях.

1.2.2.11. Поиск аутентификационных данных в трафике

NGFW может просматривать трафик на предмет операций аутентификации или регистрации в сервисах. Таким образом, он может отслеживать информацию о пользователях пассивно. Отслеживая такой трафик, NGFW должен сопоставлять пользователя с используемыми им устройствами, IP-адресами и приложениями.

1.2.2.12. Запросы к устройствам пользователей

Кроме пассивного отслеживания трафика, NGFW может напрямую обратиться к устройствам для выяснения, какой пользователь их использует.

1.2.2.13. Распознавание на основе базовых идентификаторов

Под базовыми идентификаторами понимаются MAC-адрес, IP-адрес, имя хоста и другие более-менее постоянные характеристики

1.2.2.14. Фингерпринтинг

Фингерпринтинг — это техника распознавания устройств без установки на них меток по набору метрик. Метрики могут со временем меняться или быть неуникальными между устройствами, но по их набору можно однозначно идентифицировать устройство.

1.2.2.15. Отслеживание сессий

NGFW может отлеживать состояния пользовательских сессий даже в протоколах без сохранения состояния, чтобы наблюдать за контекстом их поведения.

1.2.2.16. Отслеживание аутентификационных данных

NGFW находит в трафике операции аутентификации и, сравнивая используемые данными с данными о пользователе, определяют, какой пользователь пользуется данным устройством.

1.2.2.17. Потоковое антивирусное сканирование

NGFW должен начинать сканирование трафика на наличие вредоносного программного обеспечения ещё до полной загрузки сообщения, при поступлении первых его байтов, чтобы увеличить производительность.

1.2.2.18. Сигнатурное и эвристическое сканирование

Сканирование должно производиться на основе базы сигнатур и эвристик вредоносного ПО.

1.2.2.19. Отправка трафика в песочницу

При необходимости NGFW может отправить подозрительный трафик в особую программную среду, “песочницу”, где подозрительная программа выполняется в изолированных условиях, а её активность логгируется и анализируется на наличие вредоносных действий.

1.2.2.20. Действия с трафиком в соответствии с политиками

После произведения анализа трафика, имея представление о том, к каким пользователю и приложению относится данный трафик и какие данные в нём содержатся, NGFW может применить к потоку данных определённые действия, заданные соответствующей политикой NGFW.

1.2.2.21. Обновление базы данных сигнатур

База данных сигнатур должна поддерживаться в актуальном состоянии. Если в трафике обнаружена угроза, её сигнатура должна быть занесена в базу сигнатур.

1.2.2.22. Запрет трафика на основе прав пользователей

NGFW может запрещать доступ к некоторым URL для заданных пользователей.

1.2.2.23. Запрет трафика на основе базы запрещённых URL

NGFW может блокировать доступ к некоторым группам URL, заданных статически или по некоторому правилу.

1.2.2.24. Проверка данных на наличие чувствительной информации

NGFW может анализировать данные на наличие нежелательных к распространению данных (например, данные банковских карт).

1.2.2.25. Проверка типов файлов

В соответствии с политиками может быть запрещена передача некоторых типов файлов (например, исполняемых файлов). При этом анализ типа файла может проводиться не только по расширению имени файла, но и по структуре его содержимого.

1.2.2.26. Использование данных пассивного сканирования

Политика может составляться на основе пассивного сканирования сети. Например, если с определённого хоста рассылается большое количество вредоносного трафика, может быть принято решение об ограничении его сетевой активности.

1.2.2.27. Использование политик ИБ организации

В основном политики задаются вручную в соответствии с представлениями администраторов безопасности о том, какие пользователи должны иметь доступ к различным приложениям и какая активность в данных приложениях им разрешена.

1.2.2.28. Составление правил для сочетаний пользователей, приложений и данных

На основе пользователя, с которым ассоциирован трафик, приложения, к которому пользователь обращается, и данных, содержащихся в трафике, NGFW должен принять решение о дальнейших действиях: разрешить или запретить такой трафик.

1.2.2.29. Указание необходимости дополнительных действий

На основе тех же данных возможно принятие решение о дополнительных действиях: например, направить трафик в песочницу или отправить через прокси.

1.2.2.30. Разрешить трафик

Если трафик легитимный, NGFW отправляет трафик его адресату.

1.2.2.31. Запретить трафик

Если трафик нелегитимный, он не отправляется дальше, и предпринимаются соответствующие действия.

1.2.2.32. Отправить на сканирование угроз

Для некоторых типов трафика сканирование на наличие угроз может быть отключено политикой (например, высокочувствительные к скорости передачи данных приложения). Однако существуют и приложения, трафик которых с высокой вероятностью может содержать вредоносный код, поэтому в политике может быть указана необходимость сканирования трафика на наличие угроз.

1.2.2.33. Перенаправить через прокси

Перенаправление через прокси может иметь разные цели: например, дополнительное сканирование трафика по различным признакам.

1.2.2.34. Проанализировать на наличие определённых данных

Некоторый трафик часто содержит нежелательные данные определённой структуры. NGFW может сканировать трафик на наличие таких данных при соответствующем требовании политики.

Описанные теоретические основы позволяют понять, как функционируют существующие NGFW, и на основе этих данных сформулировать предложения по улучшению новых решений в этой области. Также такое описание может служить наглядным пособием при ознакомлении с технологиями NGFW.

1.2.3. Построение структурно-функциональной схемы и информационно-алгоритмической модели систем и средств. Исследование существующих систем, комплексов и средств, реализующих прикладные технологии создания межсетевых экранов класса NGFW.

Построим структурно-функциональную схему выполнения задач NGFW. При обработке трафика возникает множество частных случаев, поэтому в схеме достаточно много ветвлений.

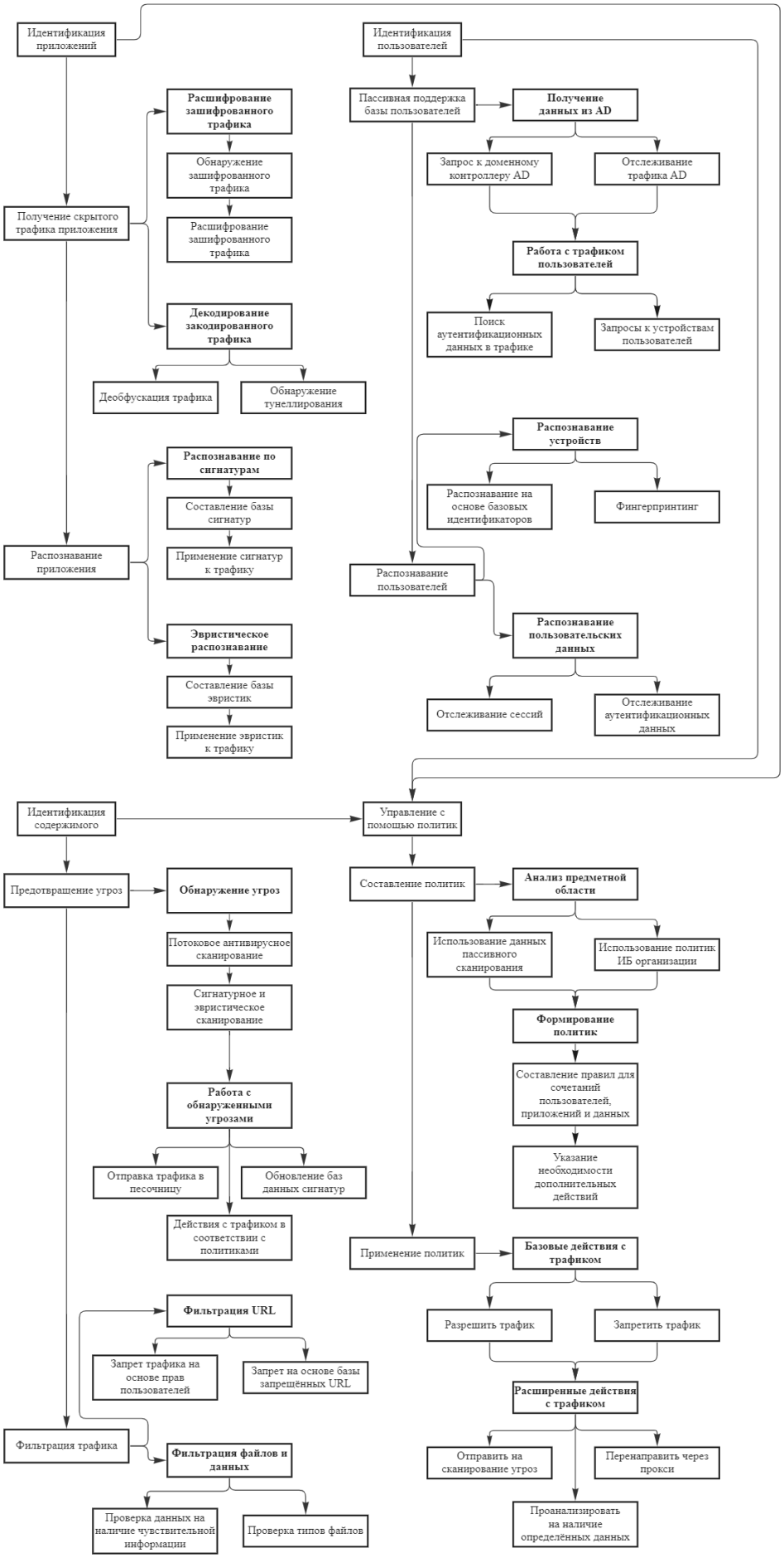

Рисунок 4 - Структурно-функциональная схема выполнения задач NFGW

2. Анализ ограничений существующих технологий создания межсетевых экранов класса NGFW. Формирование требований к современной технологии создания межсетевых экранов класса NGFW

Существующие на данный момент NGFW – уже довольно мощное решение. Оно способно комплексно анализировать трафик разных уровней и протоколов, различая приложения, не опираясь на IP-адрес и порт. Однако в плане гибкости и самообучения NGFW уступают современным реализациям WAF.

2.1. Потенциально перспективные функции NGFW

Уникальные для WAF особенности, которых нет в современных NGFW:

- Автоматическое обучение и поведенческий анализ
- Защита пользователей
- Сканирование уязвимостей
- Виртуальный патчинг
- Обнаружение корреляций и цепочек атак

Рассмотрим каждое из них.

2.1. Автоматическое обучение и поведенческий анализ

Для атак на веб-приложения злоумышленники активно используют уязвимости нулевого дня (0-day), что делает бесполезными сигнатурные методы анализа. Вместо этого нужно анализировать сетевой трафик и системные журналы для создания модели нормального функционирования приложения, и на основе этой модели выявлять аномальное поведение системы. WAF в силу своей архитектуры может разобрать весь сеанс связи пользователя, и потому способен на более углубленный поведенческий анализ, чем NGFW. В частности, это позволяет выявлять атаки с использованием автоматических средств (сканирование, подбор паролей, DDoS, фрод, вовлечение в ботнеты).

2.2. Защита пользователей

Существует класс атак (например, CSRF), направленных на клиента веб-приложения. Поскольку трафик атаки не проходит через защитный периметр, на первый взгляд защитить пользователя оказывается невозможно. Однако такие атаки изначально возможны только за счёт уязвимостей в передаваемом программном коде. Если бы NGFW отслеживали наличие подобного рода уязвимостей в передаваемом трафике, они могли бы защитить пользователей от таких атак.

2.3. Сканирование уязвимостей

На межсетевые экраны возлагается не только задача защиты веб-приложений, но и задача мониторинга атак. При этом грамотный мониторинг основан на понимании слабостей защищаемого ПО, что позволяет отсеять неактуальные попытки атак и выделить только те, которые касаются реальных уязвимостей, имеющихся в системе.

Лучшие образцы WAF имеют в своем распоряжении интегрированные сканеры уязвимостей, работающие в режиме чёрного ящика, или динамического анализа (DAST). Такой сканер может использоваться в режиме реального времени для быстрой проверки тех уязвимостей, которые «прощупывают» злоумышленники.

2.4. Виртуальный патчинг

Даже известные уязвимости невозможно устранить сразу: исправление кода требует средств и времени, а зачастую и остановки важных бизнес-процессов; иногда в случае использования стороннего ПО исправление невозможно вообще. Для парирования таких «частных» угроз в системах IDS/IPS, а по наследству в UTM/NGFW, применяются пользовательские сигнатуры. Но проблема в том, что написание такой сигнатуры требует от пользователя глубокого понимания механизма атаки. В противном случае пользовательская сигнатура может не только «пропустить» угрозу, но и породить большое количество ложных срабатываний.

В наиболее современных WAF используется автоматизированный подход к виртуальному патчингу. Для этого используется анализатор исходных кодов приложения (SAST, IAST), который не просто показывает в отчёте строки уязвимого кода, но тут же генерирует эксплойт, то есть вызов с конкретными значениями для эксплуатации обнаруженной уязвимости. Эти эксплойты передаются в WAF для автоматического создания виртуальных патчей, которые обеспечивают немедленное «закрытие бреши» ещё до исправления кода.

2.5. Обнаружение корреляций и цепочек атак

Традиционный межсетевой экран дает тысячи срабатываний на подозрительные события, в которых необходимо разбираться вручную, чтобы выявить реальную угрозу. Как отмечает Gartner, вендоры систем IPS вообще предпочитают отключить большинство сигнатур веб-приложений, чтобы снизить риск возникновения таких проблем.

Современный WAF может группировать сходные срабатывания и выявлять цепочку развития атаки — от разведки до кражи важных данных или установки закладок. В результате вместо списка из тысяч подозрительных событий ИБ-специалисты получают несколько десятков действительно важных сообщений.

Помимо описанных технологий перспективным направлением развития любой области знаний сегодня популярно считать использование машинного обучения и нейросетей. Нейросетевой анализ трафика был бы и вправду эффективным для автоматического распознавания уязвимостей нулевого дня и разработки новых сигнатур, эвристик и политик. Однако такая функция очень требовательна к аппаратному обеспечению NGFW и существенно снижает производительность обработки трафика.

2.2. Требования к современной технологии создания NGFW

Основываясь на описанных ограничениях существующих решений, перечислим основные требования к современным NGFW:

- Более глубинный анализ трафика, автоматизация на более абстрактном уровне. NGFW должен анализировать весь контекст, находить различные сессии сетевой активности, относящиеся к одной логической сессии в приложении и самообучаться на основе логов.
- Анализ журналированного трафика для поиска контекста к анализируемому в реальном времени трафику и пропущенных угроз. Для этого также требуется аппаратное обеспечение, позволяющее быстро записывать и долговременно хранить большое количество трафика.
- Отслеживание актуальных уязвимостей в устройствах сети и автоматическое блокирование лишь трафика, потенциально опасного для действительных уязвимостей.
- Реализация нейросетевого анализа трафика. Использование соответствующего аппаратного обеспечения. Для снижения негативного воздействия на производительность анализа трафика нейросетевой анализ должен производиться на отдельном сопроцессоре не в реальном времени, а по кэшированным данным.

3. Разработка методических рекомендаций по построению современной технологии создания межсетевых экранов класса NGFW

3.1. Выбор структуры построения современной технологии создания межсетевых экранов класса NGFW

Существующие NGFW уже обладают модульной структурой и сочетают в себе множество разнородных элементов, что говорит об их способности без особых сложностей включать в себя новые элементы. Таким образом, можно использовать архитектуру существующих NGFW и инкрементально улучшать их аппаратную и программную базу.

3.2. Определение методических, алгоритмических и технологических решений в области построения этапов, процессов, процедур и т.п. современной технологии создания межсетевых экранов класса NGFW

При построении современной технологии создания NGFW сохраняется основной функционал этих межсетевых экранов – а значит, структура этапов, процессов и процедур сохранится. Появятся надстройки в виде сохранения трафика в устройство хранения информации и анализа сохранённых данных. Опишем данный процесс в виде нового этапа вместе с содержащимися в нём процессами, процедурами и действиями.

Таблица 6. Описание структуры надстройки глубинного анализа контекстного трафика

Этап	**Процесс**	**Процедура**	**Действие**
Глубинный анализ контекстного трафика	Сохранение трафика	Разметка трафика	Первичный анализ трафика по заданным параметрам
			Запись выявленных метаданных
		Сохранение трафика в хранилище	Сохранение трафика в быстрый кэш
			Сохранение трафика в долговременное хранилище
	Анализ сохранённого трафика	Классический анализ	Поиск связанных сессий приложений
			Обнаружение пропущенных угроз
			Поиск долговременного контекста для текущего трафика
		Нейросетевой анализ	Формирование новых сигнатур и эвристик
			Обнаружение новых уязвимостей
			Предложение новых политик
			Обнаружение связей между сущностями и закономерностей

Опишем подробнее каждое из действий.

3.2.1. Первичный анализ трафика по заданным параметрам

Администраторы сети или создатели NGFW указывают набор параметров трафика, которые NGFW должен получать или вычислять для дальнейшей разметки трафика.

3.2.2. Запись выявленных метаданных

Трафик ассоциируется с некоторой разметкой, содержащей метаданные о нём, упрощающие поиск необходимого трафика при дальнейшем анализе. Такие метаданные служат аналогом индекса в базе данных и могут быть предметом запроса к базе хранимого трафика.

3.2.3. Сохранение трафика в быстрый кэш

Недавно полученный трафик записывается в память, имеющую малое время доступа, для оперативного поиска кратковременного контекста текущего трафика. Быстрая память дорогая, её объём сильно ограничен – поэтому здесь хранятся лишь некоторая часть недавно полученного трафика.

3.2.4. Сохранение трафика в долговременное хранилище

Весь трафик сохраняется на диск, позволяющий сохранять большое количество информации на длительное время. Такие диски имеют низкую скорость, поэтому данные с них редко используются для анализа трафика реального времени. Однако сохраняемые на длительное время данные могут многократно использоваться для поиска долговременного контекста трафика, поиска закономерностей и других активностей, требующих анализа трафика за большой промежуток времени.

3.2.5. Поиск связанных сессий приложений

Несколько сессий приложения, разнесённых во времени, могут быть не ассоциированы классическим NGFW с одним пользователем в силу отсутствия возможности сравнить старый трафик с новым. Если в трафике отсутствуют явные идентификаторы пользователя, ассоциация не будет найдена. Длительное хранение трафика позволяет напрямую сравнивать трафик и находить схожие между сессиями признаки, которые не были бы иначе использованы как идентификаторы пользователя.

3.2.6. Обнаружение пропущенных угроз

При обновлении баз сигнатур или эвристик становится возможным отследить, была ли в прошлом допущена угроза, описываемая новыми сигнатурами и эвристиками. Кроме того, если в сети обнаружена заражённая машина, можно точно отследить, каким образом это произошло.

3.2.7. Поиск долговременного контекста для текущего трафика

Увеличение просмотрового окна для контекста трафика позволяет находить более сложные цепочки действий и обнаруживать растянутые во времени атаки.

3.2.8. Формирование новых сигнатур и эвристик

Нейросеть, обученная на массиве вредоносного трафика и соответствующих ему сигнатур и эвристик может обнаруживать схожие признаки в новом трафике и обнаруживать в трафике эксплоиты на уязвимости нулевого дня, не имея точно соответствующих им сигнатур.

3.2.9. Обнаружение новых уязвимостей

Аналогично 3.2.8, нейросеть может обнаруживать уязвимости, анализируя проходящий через NGFW трафик.

3.2.10. Предложение новых политик

На основе существующих политик и проходящего через NGFW трафика (в частности выявляемых в трафике угроз), нейросеть может предлагать администраторам рекомендации по формированию новых политик.

3.2.11. Обнаружение связей между сущностями и закономерностей

На основе множества параметров нейросеть может находить сложные связи между единицами трафика, пользователями, приложениями, протоколами и т.д. Это может помочь в анализе трафика на других этапах и задаёт новый качественный уровень мониторинга.

3.3. Определение порядка использования методических рекомендаций по построению технологии и системы создания межсетевых экранов класса NGFW

Предложенные улучшения не являются безусловно и единственно верными и носят характер рекомендации, предложения к рассмотрению и усовершенствованию.

Предложенный материал также может служить в учебных целях при первичном ознакомлении с описываемой тематикой.

4. Выбор архитектуры построения межсетевых экранов класса NGFW

4.1. Построение структурно-функциональной схемы

Структура современного NGFW, как уже говорилось, сохраняется. Новая надстройка присоединяется к общей структуре как параллельный процесс, не встроенный в общий алгоритм в кратковременной перспективе. Обнаруженные зависимости, уязвимости и сигнатуры добавляются в базу данных и прочитываются механизмами, обрабатывающими текущий трафик, в свою очередь. Реализация без прерываний последовательности обработки трафика в основном потоке позволяет реализовать предложенную надстройку как отдельный параллельно работающий модуль – а значит, его можно проектировать отдельно от основной системы.

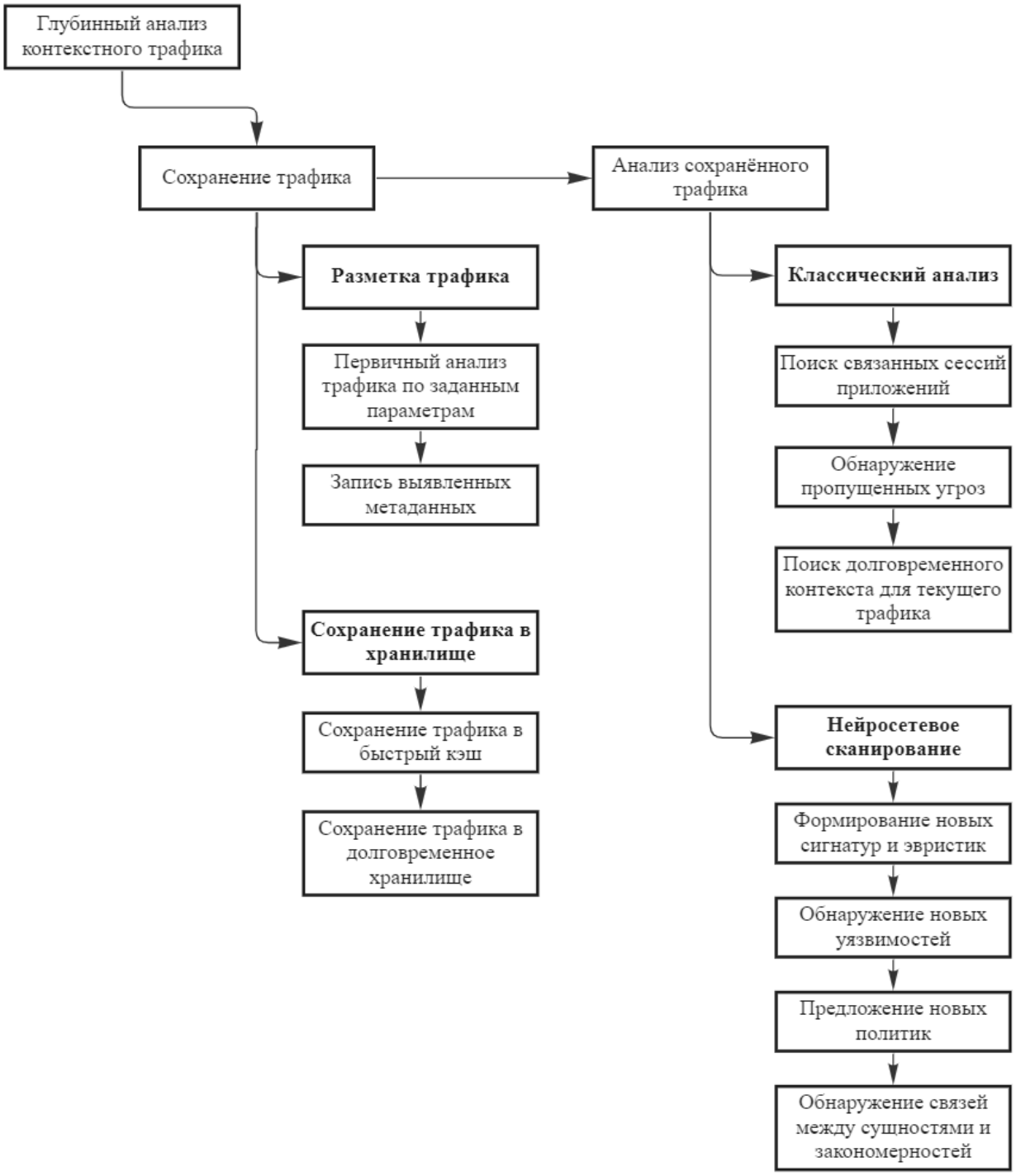

Рисунок 5 - Структура надстройки глубинного анализа контекстного трафика

4.2. Формирование информационно-алгоритмической модели

Основные этапы разработки современных NGFW:

4.2.1. Анализ возможностей конкретных улучшений по требованиям, описанным в 2.2.

Необходимо проведение научно-исследовательской работы в областях, описанных в списке требований. Результатом научно-исследовательской работы является оценка перспективности предложенных улучшений, предложения по изменению требований для оптимизации итоговых результатов и технические требования к изготовлению образцов новых NGFW.

4.2.2. Построение опытных образцов NGFW, соответствующих описанным требованиям.

Необходимо проведение опытно-конструкторской работы по разработке новых NGFW, отвечающих заданным требованиям. Продукт также должен отвечать требованиям 4.4.

4.2.3. Тестирование полученных опытных образцов в реальных условиях, выявление недостатков реализации.

Изготовленные образцы должны быть протестированы путём установки в реальную сеть с зеркалированием на него трафика. Для этого может применяться технология, аналогичная SPAN. Необходимо проверить продукт на корректность обнаружения инцидентов безопасности и на адекватность выдаваемых предложений.

4.2.4. Устранение выявленных недостатков.

По результатам тестирования необходимо подготовить набор данных о некорректной работе NGFW, после чего на их основе переработать программно-аппаратную базу продукта для устранения недостатков. После этого продукты с исправленными недостатками отправляются на шаг 3 для повторного тестирования. После итерации, в которой выявленные недостатки будут незначительными, продукт можно выпускать в массовое производство.

4.3. Выбор программно-аппаратной платформы

Основа программно-аппаратной платформы остаётся той же, что и в существующих решениях. Новое решение разрабатывается как модуль к существующей платформе, а существующая платформа должна быть переработана для поддержки подключения разрабатываемого модуля.

Аппаратная часть модуля существенно отличается от классической наличием сопроцессора для работы нейросети и увеличенного количества кэш-памяти (детали реализации могут варьироваться в зависимости от требований от SRAM-памяти до SSD-накопителей) и долговременной памяти (варьируется от SSD- до HDD-накопителей).

Программная платформа должна поддерживать работу с нейросетями. Предполагается использование существующих решений и минимальные трудозатраты на их приспособление к специфике задачи.

5. Определение перспективных направлений исследований в данной предметной области

Потенциально перспективным направлением исследований являются конкретные возможности применения нейронных сетей для их применения в NGFW. Проекты последних лет показывают, что нейросети пригодны для решения совершенно разных задач, поставленных нестандартно, и решают их порой даже не хуже людей. Можно исследовать возможности по автоматизации некоторых обязанностей сотрудников SOC с помощью нейросетей.

Также необходимы исследования по возможностям вертикального и горизонтального масштабирования NGFW для увеличения пропускной способности или возможностей по анализу трафика при сохранении пропускной способности. Конкретно, исследований требует кластеризация NGFW и их глубокая интеграция с другими продуктами.

ЗАКЛЮЧЕНИЕ

В данной работе были рассмотрены основные принципы функционирования межсетевых экранов типа NGFW и лежащие в их основе технологии передачи данных при общении пользователей с веб-приложениями. Стоит отметить, что в реальности ряд приложений, работа с которыми происходит с использованием компьютерных сетей, не ограничен веб-приложениями. Существуют и другие приложения, коммуникация с которыми производится посредством отличных от HTTP(S) протоколов. NGFW способны обрабатывать трафик и таких приложений, и принципы его работы с трафиком таких приложений не сильно отличаются от случая с HTTP(S).

В работе предложено направление развития современных NGFW: долговременное сохранение трафика и его параллельный анализ для расширения и углубления контекста анализа трафика реального времени. В экспериментальном формате для этого могут использоваться технологии нейронных сетей.

СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ Р ИСО/МЭК 27033-4-2021 Информационные технологии (ИТ). Методы и средства обеспечения безопасности. Безопасность сетей. Часть 4. Обеспечение безопасности межсетевого взаимодействия с использованием шлюзов безопасности.
2. Miller L. Next-Generation Firewalls For Dummies®, Palo Alto Networks Limited Edition – John Wiley & Sons, Inc., 2019, 82c.
3. RFC 1034, 1035, 1122, 2109, 2246, 2616, 2617, 4346, 8446
4. Чем защищают сайты, или Зачем нужен WAF? – Режим доступа: https://habr.com/ru/company/pt/blog/269165/