УДК 004.056.5

Максимов А.Е.
студент (бакалавр)
4 курс, департамент информационной безопасности
Дальневосточный федеральный университет
г. Владивосток, Россия
Баранов Д.Б.
студент (бакалавр)
4 курс, департамент информационной безопасности
Дальневосточный федеральный университет
г. Владивосток, Россия
Коваленко М.А.
студент (бакалавр)
4 курс, департамент информационной безопасности
Дальневосточный федеральный университет
г. Владивосток, Россия
Явтуховский Е.Ю.
ассистент преподавателя в департаменте информационной безопасности
Дальневосточный федеральный университет
г. Владивосток, Россия

МЕТОДЫ И СРЕДСТВА ЗАЩИТЫ ВЕБ-ПРИЛОЖЕНИЙ ОТ АТАК НА ПРИКЛАДНОМ УРОВНЕ

В статье описываются методы защиты веб-приложений от сетевых угроз с использованием Web Application Firewall (WAF). В рамках работы были рассмотрены основные векторы угроз, отражённые в OWASP Top 10, и возможные подходы к построению защиты с помощью WAF-решений. На примере двух поколений отечественного продукта данного класса от Positive Technologies раскрываются принципы реализации, интеграции с

инфраструктурой предприятия и повышения устойчивости к атакам, также были рассмотрены различные архитектурные и функциональные особенности, учитываемые при выборе решения.

Ключевые слова: информационная безопасность, веб-приложения, межсетевой экран уровня приложений, угрозы прикладного уровня, инъекции и атаки.

METHODS AND TOOLS FOR PROTECTING WEB APPLICATIONS FROM APPLICATION-LEVEL ATTACKS

MAKSIMOV A.E.
Bachelor student
4th year, Department of Information Security
Far Eastern Federal University
Vladivostok, Russia
BARANOV D.B.
Bachelor student
4th year, Department of Information Security
Far Eastern Federal University
Vladivostok, Russia
Kovalenko M.A.
Bachelor student
4th year, Department of Information Security
Far Eastern Federal University
Vladivostok, Russia
YAVTUKHOVSKY E.Y.
Assistant Lecturer, Department of Information Security
Far Eastern Federal University
Vladivostok, Russia

The article describes methods for protecting web applications from network threats using Web Application Firewall (WAF) technologies. The study examines the main threat vectors reflected in the OWASP Top 10 and explores possible approaches to building protection based on WAF solutions. Using two generations of a domestic product of this class developed by Positive Technologies as an example, the paper outlines implementation principles, integration with enterprise infrastructure, and improvements in resistance to attacks. It also considers various architectural and functional features relevant when selecting a WAF solution.

Keywords: *web applications, web application firewall, application-level threats, information security, injection attacks.*

Развитие веб-технологий влечёт за собой не только повышение удобства и скорости предоставления цифровых услуг, но и рост числа кибератак, ориентированных на прикладной уровень. Современные веб-приложения, работающие в условиях постоянного взаимодействия с пользователями и открытого доступа через интернет, представляют собой критически важные элементы инфраструктуры и требуют специализированной защиты.

Классические средства сетевой безопасности (NGFW, IDS/IPS, МЭ) оказываются недостаточными для выявления и блокировки атак, использующих особенности логики приложения. Именно в этих условиях актуальными становятся межсетевые экраны уровня приложений – Web Application Firewall (WAF).

По данным за первый квартал 2025 года, атаки на веб-приложения для организаций составили 26% от всех успешных [1], уступая только атакам на пользователей (55%) и на инфраструктурное оборудование и сетевые устройства (76%). Это ставит веб-приложения на третье место среди приоритетных целей злоумышленников. Подобный уровень внимания обусловлен тем, что веб-интерфейсы нередко выступают в роли первичной точки входа в

информационные системы, через которую далее возможно развитие атаки вглубь корпоративной инфраструктуры.

Особенно уязвимыми веб-приложения делает их сложная логика, активное взаимодействие с внешним пользователем и необходимость обработки входящих запросов через открытые сети. Многие из этих угроз систематизированы в проекте OWASP Top 10 (используется редакция 2021 года, остающаяся актуальной по настоящее время) [5], который отражает наиболее распространённые и опасные векторы атак на прикладном уровне (рисунок 1).

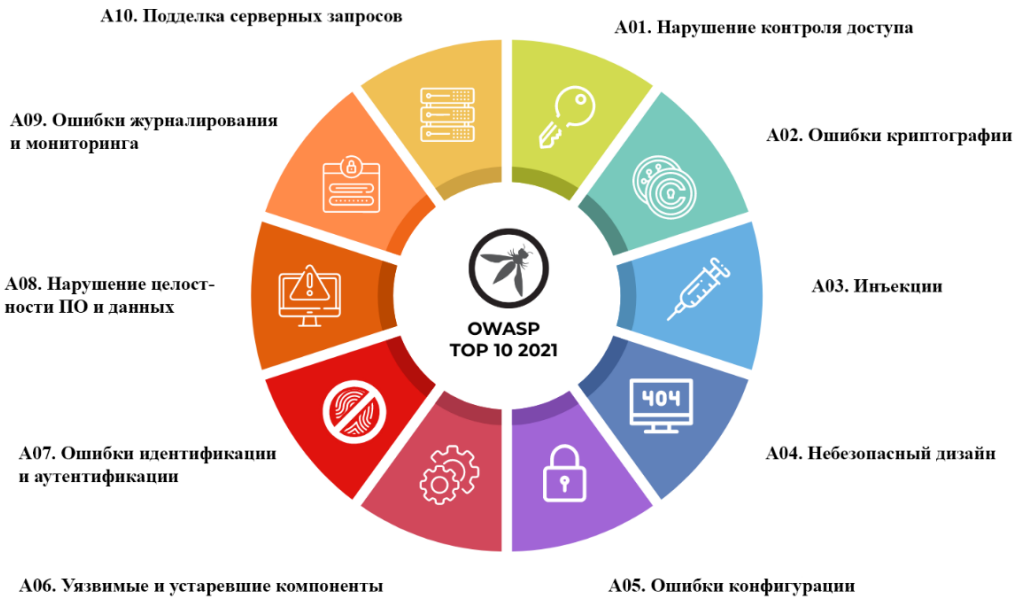

Рисунок 1 - OWASP top 10

В данный момент топ от OWASP выглядит следующим образом:

1. Broken Access Control (или нарушение контроля доступа) – ошибки в разграничении прав пользователей: доступ к чужим данным, функциям, ролям.
2. Cryptographic Failures (или ошибки криптографии) – неправильное использование шифрования, приводящее к утечкам данных и компрометации систем.
3. Injection (или инъекция) – внедрение вредоносных команд (SQL, XSS и др.), позволяющее управлять приложением, в основном используется там, где есть поля для ввода текста.

4. Insecure Design (или небезопасное проектирование) – уязвимости, заложенные на уровне архитектуры – отсутствие безопасных шаблонов и моделей.

5. Security Misconfiguration (или ошибки конфигурации) – неправильные настройки серверов, фреймворков и сервисов, открывающие доступ к системе.

6. Vulnerable and Outdated Components (или уязвимые и устаревшие компоненты) – использование библиотек и модулей с известными уязвимостями без обновлений и дополнительных мероприятий по защите.

7. Identification and Authentication Failures (или ошибки идентификации и аутентификации) – проблемы с логинами, паролями, токенами позволяют атакующему выдать себя за другого пользователя.

8. Software and Data Integrity Failures (или нарушение целостности ПО и данных) – доверие к обновлениям, скриптам, CI/CD – без проверки их подлинности, дает возможность внедрения вредоносного кода.

9. Security Logging and Monitoring Failures (или отсутствие журналирования и мониторинга) – слабое логирование и оповещение мешают выявлению атак и проведению расследований.

10. Server-Side Request Forgery (или подделка серверных запросов) – принуждение сервера отправлять запросы от своего имени, в том числе к внутренним ресурсам.

В рамках анализа угроз, отражённых в актуальной редакции OWASP Top 10, можно заметить, что значительная часть уязвимостей связана не только с первичным проникновением в систему, но и с последующей эксплуатацией – расширением прав, обходом аутентификации, закреплением в системе и проведением скрытных операций. Однако для получения первичного доступа злоумышленники чаще всего используют инъекции, недостатки управления доступом и криптографические ошибки. Ярким примером являются уязвимости, объединённые в категорию A03: Injection, которая занимает третье место в списке и охватывает такие методы атак, как SQL-инъекции, командные инъекции, а также XSS, входящую теперь в этот же класс. Эти типовые уязвимости остаются

крайне актуальными и служат входной точкой во многих целевых атаках на веб-приложения.

Важно подчеркнуть, что значительная часть этих атак не может быть выявлена средствами традиционной сетевой защиты, поскольку они маскируются под корректные с точки зрения протокола HTTP запросы, но нарушают логику приложения. Именно поэтому защита на уровне прикладного трафика требует внедрения специализированных решений – таких как Web Application Firewall.

Существует несколько подходов к реализации WAF, различающихся по архитектуре, способу развертывания и возможностям [6]:

- аппаратные решения – высокопроизводительные, но часто предполагают жёсткую зависимость от конкретного производителя как в плане технической поддержки, так и в части обновлений или совместимости оборудования, их эксплуатация требует использования исключительно фирменного оборудования, что ограничивает гибкость масштабирования и затрудняет интеграцию с другими системами. Обновления, замена компонентов и сопровождение полностью контролируются производителем, что увеличивает стоимость владения и снижает адаптивность решения;
- программные решения – развёртываются на физических или виртуальных серверах, как внутри периметра заказчика (on-premise), так и в облаке, что обеспечивает гибкость и масштабируемость, зависимость от вендора сохраняется на уровне лицензий и платформенных ограничений, но выражена значительно слабее, чем у аппаратных решений;
- облачные WAF – удобны для малого и среднего бизнеса, но могут не соответствовать требованиям регуляторов и зачастую предполагают зависимость от внешнего поставщика, что делает их менее приемлемыми для организаций с повышенными требованиями к контролю и автономности;

- гибридные решения – совмещают преимущества локальной фильтрации и централизованной аналитики.

Основу эффективности любого WAF составляет не только способ его развёртывания, но и применяемые методы анализа прикладного трафика. Современные решения сочетают различные подходы, среди которых можно выделить [6]:

- сигнатурный анализ – выявление атак по известным шаблонам (например, SQLi, XSS);
- эвристический анализ – обнаружение подозрительных шаблонов поведения, не попадающих под известные сигнатуры;
- поведенческий анализ – построение моделей "нормального" трафика и выявление аномалий на основе отклонений;
- контекстная валидация – проверка соответствия параметров типовым структурам и допустимым значениям;
- декомпозиция сложных форматов – анализ вложенных JSON, XML, JWT, включая поля авторизации и токены;
- интеграция с внешними системами – использование данных от SIEM, CMDB, систем реагирования и каталогов угроз для корреляции событий.

Функционально WAF способен не только блокировать или разрешать трафик, но и:

- осуществлять маскирование ошибок сервера и убирать лишнюю информацию в заголовках;
- внедрять капчи и механизмы защиты от автоматизации (ботов);
- выполнять сценарии предобработки и трансформации трафика;
- логировать события и формировать отчёты для последующего анализа.

Такой комплексный подход позволяет не только обнаруживать атаки на ранних этапах, но и минимизировать количество ложных срабатываний при одновременном сохранении гибкости и управляемости системы.

Выбор WAF и его интеграция должны основываться не только на технических характеристиках, но и на понимании угроз, процессов и требований законодательства.

Ключевые аспекты, которые необходимо учитывать:

- анализ угроз – определить наиболее вероятные векторы атак: инъекции, API, обходы авторизации, попытки автоматизированного доступа. При наличии нестандартной логики необходим поведенческий анализ и контекстная валидация;
- архитектура инфраструктуры – важно учитывать, используется ли централизованная или распределённая модель, присутствуют ли микросервисы, облачные среды. От этого зависит выбор модели внедрения: reverse proxy (обратный прокси), bridge (мост с прозрачной фильтрацией на более низких уровнях) или agent-based (агенты на конечных узлах) [2, с. 280];
- форматы и протоколы – современное веб-приложение должно быть защищено от атак на структуры JSON, XML, JWT и др. Поддержка этих форматов является обязательной для актуального WAF;
- интеграция с экосистемой ИБ – решение должно поддерживать взаимодействие с SIEM и CMDB, системами управления уязвимостями и средствами реагирования на инциденты;
- управляемость и логирование – наличие REST API, нормализованных логов, централизованного интерфейса управления и возможности экспорта событий в требуемый формат;
- масштабируемость – система должна работать стабильно при средней нагрузке и масштабироваться при её росте;
- обновляемость – наличие актуальных сигнатур, механизмов самообучения, оперативной поддержки и регулярных обновлений.

Примером развития WAF-архитектуры в рамках одного вендора является переход от PTAF 3 к PTAF PRO (версия 4), разработанных компанией Positive Technologies.

PTAF 3 – монолитное решение на базе NGINX [3]. Оно поддерживает различные режимы: обратный (Рис. 2) и прозрачный прокси, L2-мост, sniffer; использует централизованное ядро, позволяя вручную модифицировать конфигурации и отличаясь высокой стабильностью. Однако масштабируемость и адаптация под распределённые среды были ограничены.

PTAF PRO – микросервисная архитектура, основанная на оркестрации Kubernetes. Все компоненты (Ingress, Border, Policy Engine, UI и др.) функционируют в отдельных подах [4], обеспечивая:

- горизонтальное масштабирование;
- автоматическое обновление и управление через REST API;
- интеграцию с внешними системами через API;
- совместимость с ClickHouse, PostgreSQL, MinIO и RabbitMQ;
- изоляцию политик (Рис. 2) и многоарендность (multi-tenancy).

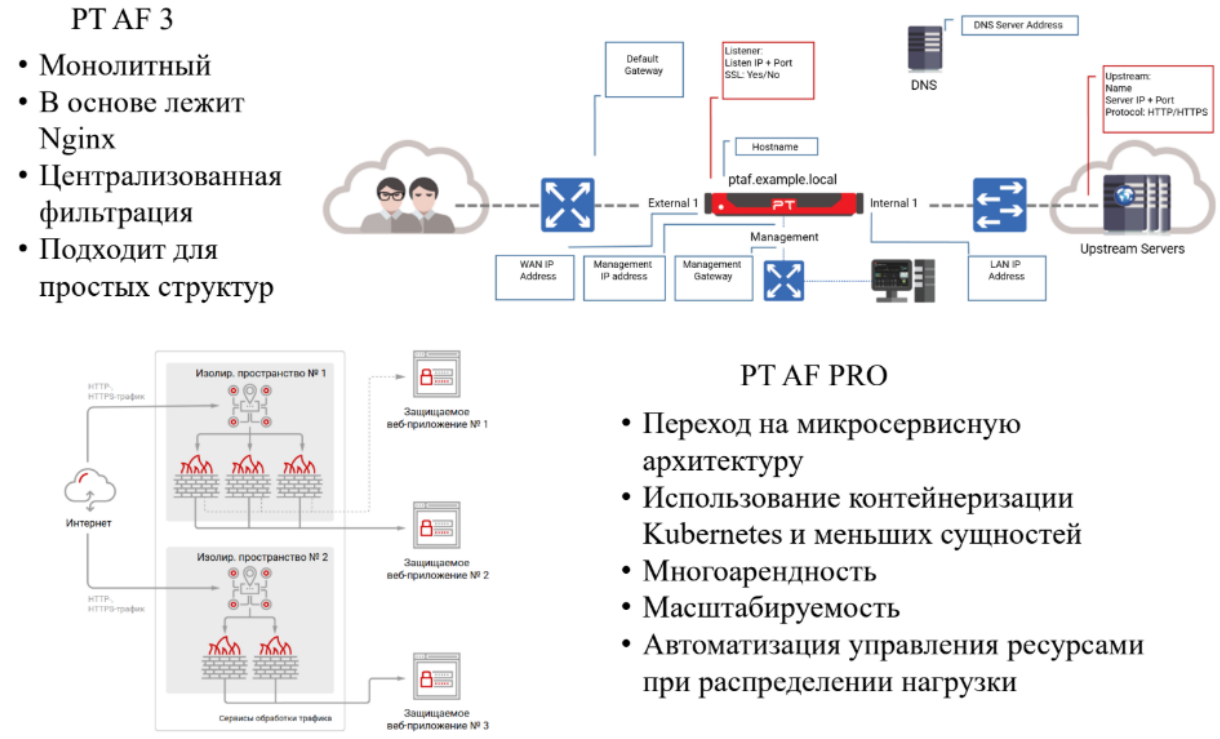

Рисунок 2 - архитектура поколений PTAF

Ключевым архитектурным преобразованием является отказ от централизованного ядра в пользу распределённой модели обработки данных и

управления. Функциональные элементы WAF, такие как анализаторы трафика, сенсоры, службы политики, интерфейсы управления, базы хранения и модули интеграции с внешними системами реализованы как изолированные сервисы, взаимодействующие между собой через внутренние API. В контексте Kubernetes эти изолированные функциональные единицы представлены в виде подов – минимальных функциональных единиц для развёртывания в платформе, которая представляет собой группу из одного или нескольких контейнеров, работающих вместе и использующих общие ресурсы. Каждый под отвечает за выполнение строго определённой задачи в общей системе: например, один под может обеспечивать обработку HTTP-запросов, другой – собирать события безопасности, третий – обеспечивать взаимодействие с внешним SIEM-решением. Такая декомпозиция позволяет динамически масштабировать отдельные компоненты в зависимости от текущего приоритета по нагрузке, устранять единичные точки отказа и достигать высокой отказоустойчивости без необходимости перезапуска всей системы.

Благодаря возможностям Kubernetes, система PT AF PRO поддерживает автоматическое восстановление подов в случае сбоев и горизонтальное масштабирование – создание новых экземпляров при росте нагрузки, а также размещение компонентов на разных физических и виртуальных узлах в рамках единого логического кластера. Кроме того, использование подов и контейнеризированных компонентов обеспечивает изоляцию процессов и упрощает внедрение обновлений. Каждый сервис можно обновлять независимо или изменять, без воздействия на остальные элементы системы. Это существенно снижает риски при внедрении новых функциональных особенностей, исправлении ошибок и переходе между версиями.

Стоит отметить, что при переходе к архитектуре PTAF PRO были утрачены некоторые функциональные компоненты, ранее доступные в PTAF 3:

- ICAP – ранее использовался для интеграции с песочницами и CABЗ, позволяя передавать трафик на внешнюю проверку;

- группировка портов – обеспечивала удобную маршрутизацию и настройку всех входных и выходных точек взаимодействия в рамках одного сервиса;
- модификация запросов – на базе встроенных возможностей NGINX позволял изменять, удалять и переопределять элементы трафика внутри PTAF.

Отсутствие этих возможностей в PTAF PRO напрямую связано с переходом на микросервисную архитектуру и эксплуатацию в Kubernetes, где такие функции изначально не входят в концепцию лёгких контейнерных компонентов.

Реализация соответствующей логики возможна, но теперь требует внедрения внешнего балансировщика, на котором будут реализовываться данные функции. Таким образом, архитектурный сдвиг в сторону микросервисности не столько лишает систему прежних возможностей, сколько требует переосмысления точек их реализации. Это отражает общий вектор развития: от централизованных и монолитных WAF-комбайнов к гибким, распределённым и облачно-ориентированным платформам.

Web Application Firewall является ключевым элементом защиты современного веб-пространства. Разнообразие угроз требует перехода от простых сигнатурных решений к комплексным архитектурам, способным адаптироваться к бизнес-логике приложений.

Развитие отечественных WAF-средств, таких как PTAF PRO, демонстрирует движение в сторону микросервисности, масштабируемости и совместимости с DevOps-инфраструктурой. Грамотный выбор WAF и его внедрение с учётом архитектуры организации позволяет существенно повысить защищённость критически важных веб-сервисов.

СПИСОК ЛИТЕРАТУРЫ:

1. Голушко А. Актуальные киберугрозы: IV квартал 2024 года — I квартал 2025 года // Positive Technologies: электрон. ресурс. URL: https://ptsecurity.com/ru-

ru/research/analytics/aktualnye-kiberugrozy-iv-kvartal-2024-goda-i-kvartal-2025-goda/ (дата обращения: 25.06.2025).

2. Богомолов С.В., Демкин Д.А. ТЕХНОЛОГИЯ WAF И ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ // Вестник науки №9 (78) том 2. С. 279 - 282. 2024 г. ISSN 2712-8849 // URL: https://www.вестник-науки.рф/article/17172 (дата обращения: 04.07.2025 г.)

3. Positive Technologies. Документация PT AF 3 // справочный портал Positive Technologies: электрон. ресурс. URL: https://help.ptsecurity.com/ru-RU/projects/af3/3.7.4/help/71660299 (дата обращения: 28.06.2025).

4. Positive Technologies. Документация PT AF PRO // справочный портал Positive Technologies: электрон. ресурс. URL: https://help.ptsecurity.com/ru-RU/projects/af/latest/help/71660299 (дата обращения: 05.07.2025).

5. OWASP Foundation. Top Ten Top 10 Web Application Security Risks // OWASP: электрон. ресурс. URL: https://owasp.org/www-project-top-ten/ (дата обращения: 25.06.2025).

6. Wikipedia. Файрвол веб-приложений // Свободная энциклопедия Wikipedia : электрон. ресурс. URL: https://ru.wikipedia.org/wiki/Файрвол_веб-приложений (дата обращения: 27.06.2025).

artem-maxik2@mail.ru